WebSphere® Process Server for Multiplatforms

Version 6.1.0

IBM

Administering WebSphere Process Server

WebSphere® Process Server for Multiplatforms

Version 6.1.0

Administering WebSphere Process Server

IBM

Contents

Figures

vii

Tables

Chapter 1. Overview of administering WebSphere Process Server

Administering WebSphere® Process Server involves preparing, monitoring, and modifying the environment into which applications and resources are deployed, as well as working with the applications and resources themselves. Use the following topics to learn more about the interfaces and configuration files used for administration tasks.

The administrative console for WebSphere Process Server

The administrative console is a browser-based interface used to administer WebSphere Process Server applications, services, and other resources at a cell, node, or server scope. You can use it from stand-alone process servers and from deployment managers that manage all servers in a cell in a networked environment.

Note: The WebSphere Process Server administrative console is part of the Integrated Solutions Console framework in general, and the WebSphere Application Server administrative console in particular. As a result, many administrative tasks (for example, setting security, viewing logs, and installing applications) are the same for both WebSphere Process Server and WebSphere Application Server. Those tasks are documented in the WebSphere Application Server Information Center.

Understanding the WebSphere Process Server tasks associated with the console

Common WebSphere Process Server tasks that are performed in the console include:
- Setting up the administrative architecture and environment
- Configuring process servers and their settings
- Deploying new applications to a server
- Managing existing applications and configurations
- Managing resource providers for applications
- Managing server resources, such as relationships, business processes, tasks, adapters, business rules, and selectors
- Administering the Business Process Choreographer
- Managing failed events on the process server
- Administering the Common Event Infrastructure
- Configuring product security
- Collecting data for troubleshooting purposes

Understanding the administrative console interface

The administrative console has three distinct areas:

Task bar

The task bar is located at the very top of the console. It provides options for logging out of the console, accessing product information, and accessing support.

Navigation tree

The navigation tree is on the left side of the console. It provides links to console pages that you use to create and administer servers, applications, and other resources.

Click the plus sign (+) beside an item in the navigation tree to expand it, or click the minus sign (-) to collapse the item. You can also click the item itself to toggle between its expanded and collapsed state.

Workspace

The workspace is located on the right side of the console. It displays pages that you use to create and administer servers, applications, and other resources. You access these pages by clicking the links in the navigation tree, or by clicking links within the workspace pages themselves.

See "Administrative console pages" on page 4 for a discussion of the types of pages that are displayed in the workspace.

On the far right side of the workspace is the help portal. It provides brief information about each field on the current page, as well as a link to more detailed information in the help browser.

Locating WebSphere Process Server-specific areas of the administrative console

WebSphere Process Server resources are grouped into several areas of the administrative console. Use the navigation tree to locate these resources, as follows:

- **Servers > Deployment Environments**: Provides access to manage deployment environments, as well as a wizard to help you create a new deployment environment.

  This option is available only if you have installed WebSphere Application Server Network Deployment.

- **Servers > Application servers >** *server_name*: Provides access to the following:
  - Container settings for business processes and human tasks
  - Business Integration configuration (tabbed page of deployment target functions)
  - Service Component Architecture configuration
  - Common Event Infrastructure server and destination configuration
  - Business Process Choreographer configuration
  - Business rules configuration
  - Selectors
  - WebSphere Business Integration Adapter Service
  - Application Scheduler
  - Extended Messaging Service

- **Servers > clusters >** *cluster_name*: Provides access to the following:
  - Container settings for business processes and human tasks

- Business Integration configuration (tabbed page of deployment target functions)
  - Service Component Architecture configuration
  - Common Event Infrastructure server and destination configuration
  - Business Process Choreographer configuration
  - Business rules configuration
- **Resources** : Provides access to the following:
  - WebSphere Business Integration Adapters
  - People directory provider
  - Extended Messaging Provider
  - Remote Artifacts
- **Integration Applications**: Provides access to the following:
  - Failed event manager
  - Relationship manager
  - Common Base Event Browser

The Welcome page provides a task filtering selector to help refine the administrative console pages. Each filter provides a subset of administrative console functionality pertinent to a particular set of tasks (for example, process server administration or enterprise service bus administration).

In addition, the main Welcome page contains links to product-specific Welcome pages for each product you have installed. Use these pages to see detailed information about each product.

Accessing online help from the administrative console

The administrative console provides online help for each page and field. Access the help in one of the following ways:

- Click **Help** from the console task bar to view online help in a new Web browser.

  From the help browser, you can do the following:
  - Browse for the topic you want to view in the Index tab. Click the link for that topic to open it in the right panel of the browser.
  - Search for a topic by specifying one or more key words in the Search tab. All matching topics are displayed in the navigation tree; click a topic link to view it.
- Place the cursor over a field to view hover help about that field.
- Place the cursor over a field and wait for the question mark (**?**) icon to appear. When the icon appears, click the field name to display brief help about it in the help portal (the right-most panel in the workspace).

  If you want to view extended information about the field, or about the entire page and its associated tasks, click the **More information about this page** link at the bottom of the help portal.

Administrative console guided activities

Guided activities lead you through common administrative tasks that require you to visit multiple administrative console pages.

Guided activities display each administrative console page that you need to perform a task, surrounded by the following information to help you perform the task successfully:

- An introduction to the task and its essential concepts
- A description of when and why to perform the task
- A list of other tasks to do before and after performing the current task
- The main steps to complete during the task
- Hints and tips to help you avoid or recover from problems
- Links to field descriptions and extended task information in the online documentation

Figure 1 shows an example of the administrative console displaying a guided activity.
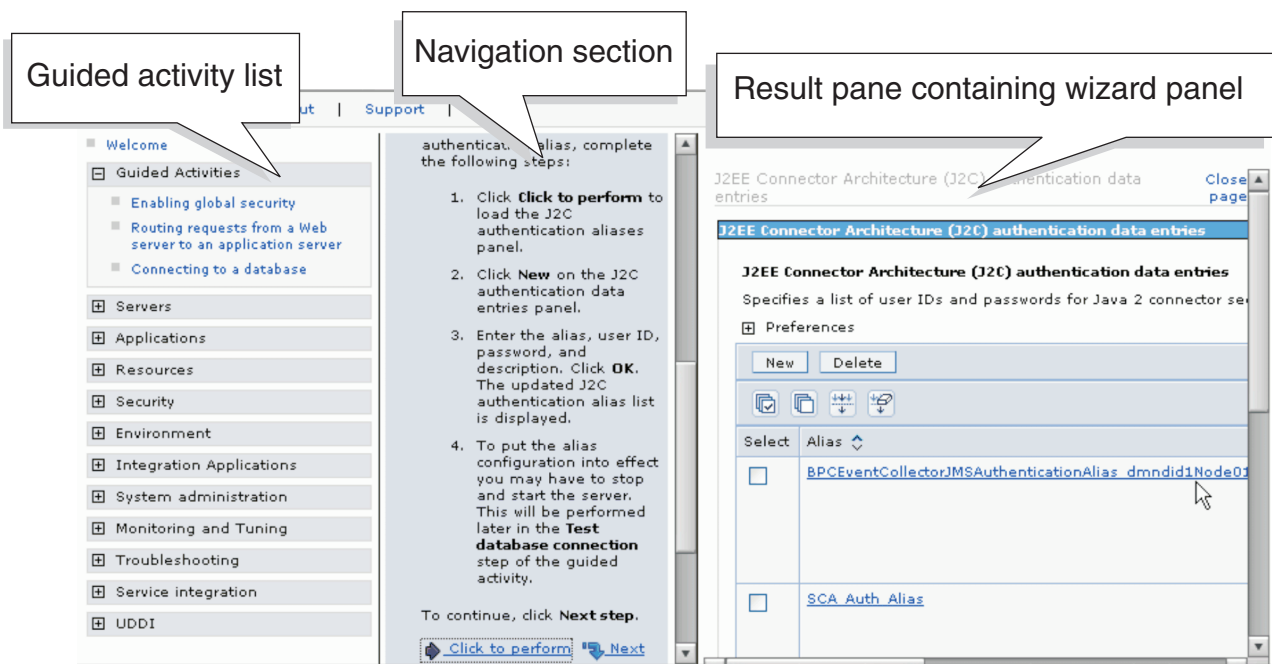
Figure 1. A guided activity

Administrative console pages

Administrative console pages are formatted in one of three ways: Collection, detail, and wizard pages. Understanding the layout and behavior of each type of page can help you use them more effectively.

Collection pages

A collection page manages a collection of existing administrative objects (for example, relationships, failed events, or resource adapters). It contains one or more of the following elements:

Scope and Preferences
> The scope and preferences help determine which administrative objects are displayed in the table, and how they should appear.

Table of existing objects
> The table displays existing administrative objects of the type specified by the collection page. The table columns summarize the values of the key

settings for these objects. If no objects exist yet, the table is empty. Use the available buttons to create a new object.

Buttons for performing actions
The typical buttons are described in "Administrative console buttons." In most cases, you need to select one or more objects in the collection table, then click a button. The action is applied to all selected objects.

Sorting toggle buttons
After each column heading in the table are icons to sort the entries in ascending (^) or descending (v) order. By default, items such as object names are sorted in descending order (alphabetically).

Detail pages

A detail page is used to view details about an object and to configure specific objects (such as an application server or a listener port extension). It typically contains one or more of the following elements:

Configuration tabbed page
This tabbed page is used to modify the configuration of an administrative object. Each configuration page has a set of general properties specific to the object. Additional properties can be displayed on the page, depending on the type of administrative object you are configuring.

Changes to this tabbed page can require a server restart before they take effect.

Runtime tabbed page
This tabbed page displays the configuration that is currently in use for the administrative object. It can be read-only. Note that some detail pages do not have runtime tabs.

Changes to this tabbed page take effect immediately.

Local topology tabbed page
This tabbed page displays the topology that is currently in use for the administrative object. View the topology by expanding and collapsing the different levels of the topology. Note that some detail pages do not have local topology tabs.

Buttons for performing actions
Buttons to perform specific actions display only on configuration tabbed pages and runtime tabbed pages. The typical buttons are described in "Administrative console buttons."

Wizard pages

Wizard pages help you complete a configuration process comprised of several steps. Be aware that wizards can show or hide certain steps, depending on the characteristics of the specific object you are configuring. See "Administrative console guided activities" on page 3.

Administrative console buttons

The administrative console interface contains a number of buttons, depending on which page you are currently viewing. This topic describes the available console buttons.

The following graphical buttons are located at the top of a table that displays WebSphere Process Server resources:

Button	Resulting action
Check all	Selects each resource (for example, a failed event or a relationship instance) that is listed in the table, in preparation for performing an action against those resources.
Uncheck all	Clears all selected resources so that no action is performed against them.
Show the filter view	Opens a dialog box to set a filter. Filters are used to specify a subset of resources to view in the table. See "Setting administrative console filters" on page 19.
Hide the filter view	Hides the dialog box used to set a filter.
Clear filter value	Clears all changes made to the filter and restores the most recently saved values.

The following buttons appear at the bottom of a WebSphere Process Server administrative console page. Not all buttons appear on all pages.

Add   Adds the selected or typed item to a list, or produces a dialog box for adding an item to a list.

Apply Saves your changes to a page without exiting the page.

Back   Displays the previous page or item in a sequence. The administrative console does not support using the Back and Forward options in the web browser, which can cause intermittent problems. Use the **Back** or **Cancel** buttons in the console instead.

Cancel
    Exits the current page or dialog box, discarding all unsaved changes. The administrative console does not support using the Back and Forward options in the web browser, which can cause intermittent problems. Use the **Back** or **Cancel** buttons in the console instead.

Clear   Clears your changes and restores the most recently saved values.

Clear selections
    Clears any selected cells in the tables on this tabbed page.

Close Exits the dialog.

Delete Removes the selected instance.

OK   Saves your changes and exits the page.

Reset Clears your changes on the tab or page and restores the most recently saved values.

Save   Saves the changes in your local configuration to the master configuration.

For a complete list of buttons used in the administrative console (for administering both WebSphere Application Server and WebSphere Process Server resources), refer to the WebSphere Application Server Information Center.

Command-line tools, scripts, and programming interfaces

WebSphere Process Server provides command-line tools, scripting interfaces, and programming interfaces (administrative programs) to administer the runtime environment.

Command-line tools

Command-line tools are simple programs that you run from an operating system command-line prompt to perform specific tasks. Using these tools, you can start and stop application servers, check server status, add or remove nodes, and other tasks.

The WebSphere Process Server command-line tools include the serviceDeploy command, which processes .jar, .ear, .war and .rar files exported from a WebSphere Integration Developer environment and prepares them for installation to the production server.

See **Reference > Commands and scripts** in this information center for details about the command-line tools.

Scripting (wsadmin)

The WebSphere administrative (wsadmin) scripting program is a non-graphical command interpreter environment that enables you to run administrative options in a scripting language and to submit scripting language programs for execution. It supports the same tasks as the administrative console. The wsadmin tool is intended for production environments and unattended operations.

See **Reference > Commands and scripts** in this information center for details about the programming interfaces.

Administrative programs

Administrative programs are a set of Java™ classes and methods under the Java Management Extensions (JMX) specification that provide support for administering Service Component Architecture (SCA) and business objects. Each programming interface includes a description of its purpose, an example that demonstrates how to use the interface or class, and references to the individual method descriptions.

See **Reference > Javadoc** in this information center for details about the programming interfaces.

About Business Process Choreographer Explorer

Business Process Choreographer Explorer is a Web application that implements a generic Web user interface for interacting with business processes and human tasks.

You can configure one or more Business Process Choreographer Explorer instances on a server or cluster. It is sufficient to have a WebSphere Process Server installation with a WebSphere Process Server profile, or a WebSphere Process Server client installation – it is not necessary to have Business Process Choreographer configured on the server or cluster. The WebSphere Process Server client installation is only the infrastructure that you need to connect a client to a WebSphere Process Server, it does not contain the Business Process Choreographer

Explorer. Use the deployment manager to install the Business Process Choreographer Explorer on the servers in the WebSphere Process Server client installation as well.

A single Business Process Choreographer Explorer can only connect to one Business Process Choreographer configuration, though it does not have to connect to a local configuration. However, you can configure multiple instances of the Business Process Choreographer Explorer on the same server or cluster, and each instance can connect to different Business Process Choreographer configurations.
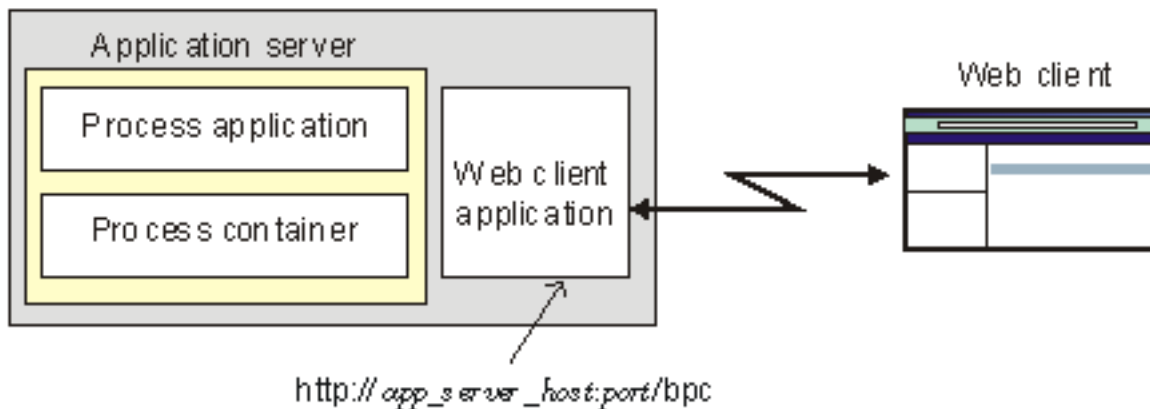
When you start Business Process Choreographer Explorer, the objects that you see in the user interface and the actions that you can take depend on the user group that you belong to and the authorization granted to that group. For example, if you are a business process administrator, you are responsible for the smooth operation of deployed business processes. You can view information about process and task templates, process instances, task instances, and their associated objects. You can also act on these objects; for example, you can start new process instances, create and start tasks, repair and restart failed activities, manage work items, and delete completed process instances and task instances. However, if you are a user, you can view and act on only those tasks that have been assigned to you.

About Business Process Choreographer Observer

About Business Process Choreographer Observer.

You can use Business Process Choreographer Observer to create reports on processes that have been completed. You can also use it to view the status of running processes. This describes the architecture and possible configuration paths.

Business Process Choreographer Observer uses the Common Event Infrastructure (CEI) to collect events that are emitted by WebSphere Process Server. You can either use a number of predefined reports or define your own reports to get an overview of the number of processes, activities, or other aggregate data. You can also get information about specific processes or activities.

Business Process Choreographer Observer is based on two J2EE enterprise applications, which are shown in the following figure:
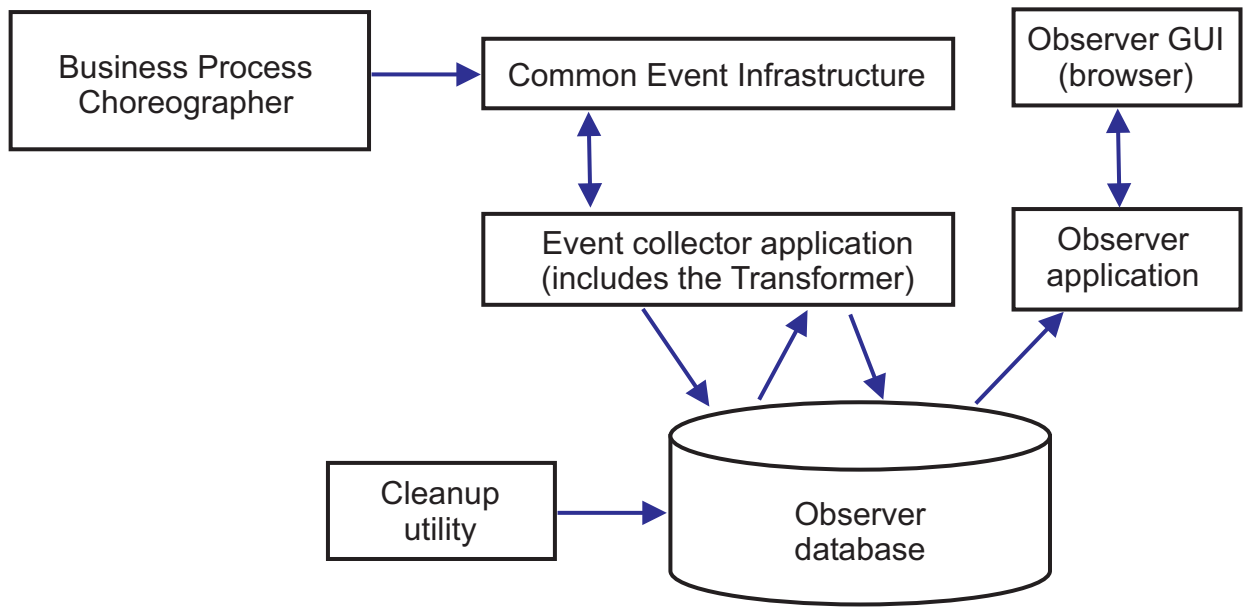
Figure 2. Business Process Choreographer Observer architecture

- The event collector application reads event information from the CEI bus and stores it in the event collector table in the Business Process Choreographer Observer database.
- The observer database is a set of database tables that store the event data.
- Periodically the event transformer is triggered, which transforms the raw event data into a format that is suitable for queries from the Business Process Choreographer Observer.
- The observer application generates the reports and performs other actions that the user can initiate using the observer graphical user interface (GUI).
- You can use the GUI to generate your reports. You can also store and retrieve reports that you have defined.
- A cleanup utility can be used to remove records from the observer database, which can help to improve the performance.

Simple configurations

A simple configuration, where performance is not an important consideration is illustrated in the following figure.
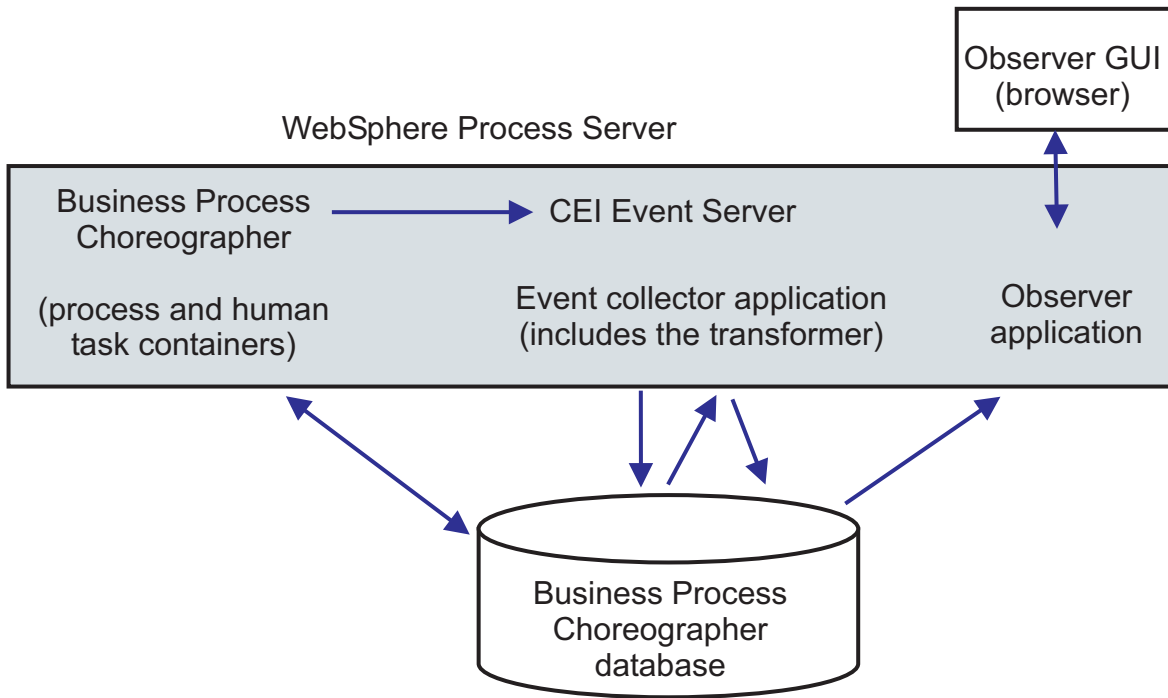
Figure 3. Business Process Choreographer Observer standalone setup

Everything is installed on a single machine, and Business Process Choreographer and Business Process Choreographer Observer use the same database.

This kind of simple configuration is created if you create a sample Business Process Choreographer configuration. Also the bpeconfig.jacl tool defaults to configuring the observer and event collector applications on the same deployment target as the Business Process Choreographer configuration. Common Event Infrastructure (CEI) logging will be enabled and the necessary database schema is created in the Business Process Choreographer Derby database BPEDB. This configuration path can be ideal if performance is not an important consideration.

High-performance configurations

Interactive configuration tools are provided which give you the freedom to exploit the full potential of the Business Process Choreographer Observer architecture. For example, in an ideal configuration for performance in a production system, the three Business Process Choreographer elements run on separate machines, and the Business Process Choreographer Observer has its own database.
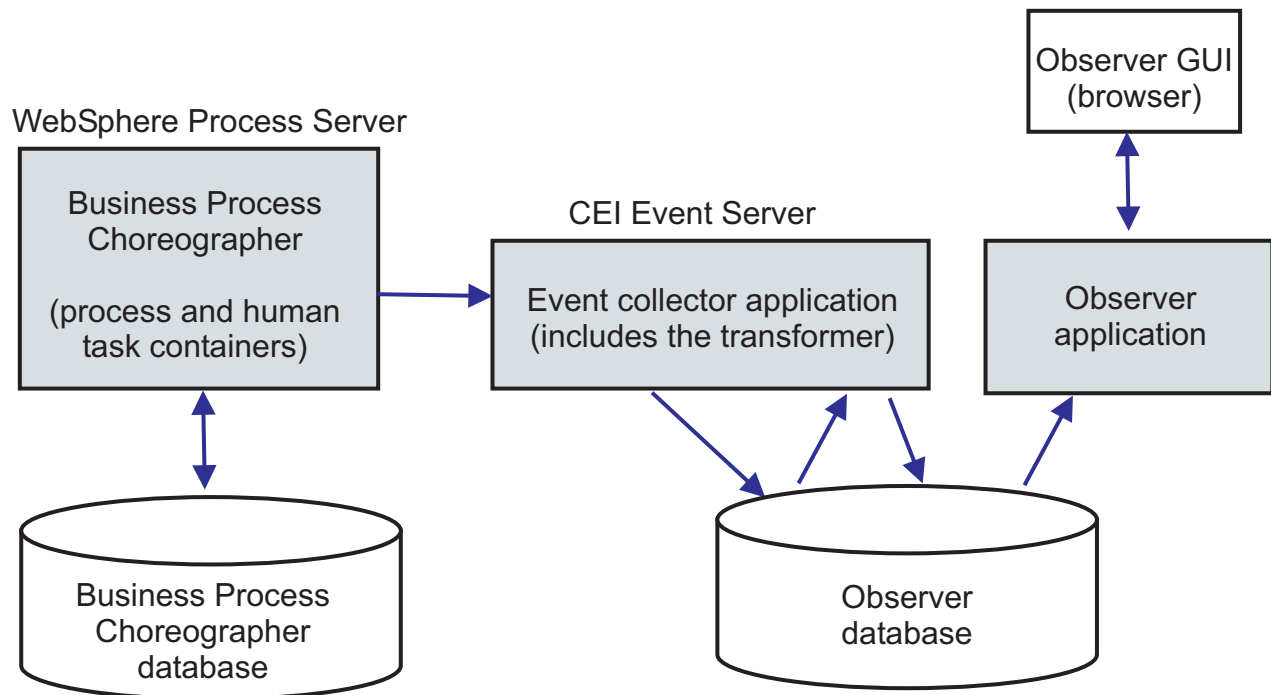
Figure 4. Business Process Choreographer Observer setup for production performance

If you want to use a separate database for the Business Process Choreographer Observer, or to add the Business Process Choreographer Observer to an existing Business Process Choreographer configuration in a clustered setup, or to use more sophisticated database options, perform Configuring the Business Process Choreographer Observer.

In a network deployment environment

The following constraints apply if you want to configure Business Process Choreographer Observer in a network deployment environment.

- CEI must be configured in your cell.
- As illustrated in the previous figure, the Business Process Choreographer event collector must be configured on a deployment target where the CEI Event server is configured. If the CEI Event server is configured on a different cluster than Business Process Choreographer, you must configure the Business Process Choreographer event collector on a deployment target where the CEI Event server is configured. The Business Process Choreographer Observer application does not need to be installed on the same machine as the event collector.

Business rules manager

The business rules manager is a Web-based tool that assists the business analyst in browsing and modifying business rule values. The tool is an option of WebSphere Process Server that you can select to install at profile creation time or after the initially installing the server.

Business rules are designed and developed in WebSphere Integration Developer using if/then rule sets and decision tables to implement their operations. Business rules can also be created in WebSphere Business Modeler; however Modeler only supports the creation of business rule tasks, which become rule sets when exported out of Modeler. The rule sets and decision tables are set into templates. The

templates control which aspects of a business rule you can modify and by exactly how much. They define the structure of if/then rules, condition cases, and actions for decision tables.

The templates provide the mechanism for business rule runtime authoring in the business rules manager. Using the template, you can modify business rule values, create a new rule within a rule set or a new condition or action within a decision table, and publish changes to business rule definitions at run time.

Business rules are organized into business rule groups. Business rule groups are used to interface to and invoke rules. Rule sets and decision tables are never invoked directly.

For more information about building and deploying business rules, see the WebSphere Integration Developer Information Center.

Configuration information

Most configuration data for WebSphere Process Server is stored in XML files, which are kept in directories in the configuration repository tree (the master repository).

The directory in which a configuration file exists determines its scope, or how broadly or narrowly that data applies.
- Files in an individual server directory apply to only that server.
- Files in an application directory apply to only that application.
- Files in a cluster-level directory apply to only that cluster.
- Files in a node-level directory apply to every server on that node.
- Files in a cell directory apply to every server on every node within the entire cell.

Table 1. WebSphere Process Server configuration files

Configuration file	Description
server-wbi.xml	Identifies a server and its components, including Adaptive Entity Service, Extended Messaging Service, Business Rules and Selector Auditing Service, and WebSphere Business Integration Adapter Service configuration.
resources-wbi.xml	Defines operating environment resources for WebSphere Process Server and is present at the cell, node, and server scopes. This includes Extended Messaging Providers and WebSphere Business Integration Adapters.
cell-wbi.xml	Identifies a cell. This file is used to store the Relationship Service configuration, and is only present at the cell scope.
server-bpc.xml	Identifies a Business Process Choreographer container and its components.
resources-bpc.xml	Defines operating environment resources for a Business Process Choreographer container, including configuration information for the people directory provider. This file is present at the cell, node, and server scopes.

Table 1. WebSphere Process Server configuration files (continued)

Configuration file	Description
deployment-bpc.xml	Configures application deployment settings for a business process container.
server-core.xml	Identifies configuration information for core WebSphere Process Server configurations, including the Artifact Loader Service and Business Context Data Service.

WebSphere Process Server configuration files can be edited through the administrative console, wsadmin, and scripting. No manual editing is required.

See the WebSphere Application Server Information Center for complete information about server configuration files.

Chapter 2. Getting started with the administrative interfaces

Use the information in these topics to set up, explore, and manage WebSphere Process Server.

Getting started with the administrative console

Use the tasks in this topic to get started using the administrative console to manage and administer WebSphere Process Server resources.

The following tasks help you start the server and the administrative console, set the console scope and preferences, and save your work to the master repository.

- **Start the server.**

  Before you can use the administrative console, you must start the stand-alone server or deployment manager. To start a stand-alone server, see Starting a stand-alone server. For network deployment, see Starting a deployment manager.

- **Start the administrative console.**

  See "Starting and stopping the administrative console" for details.

- **Specify console preferences.**

  Preferences control how data is displayed in the administrative console, as well as how the workspace behaves. See "Setting administrative console preferences" on page 18.

- **Set the console scope.**

  The scope specifies the level at which a resource is visible on the administrative console. A resource can be visible in a console collection table at the cell, node, cluster, or server scope. See the WebSphere Application Server Information Center for details on setting the scope.

- **Create filters to view information.**

  Filters specify which data is shown in a column on a collection page. See "Setting administrative console filters" on page 19.

- **Optional: Set the session timeout for the console.**

  By default, a console session times out after 30 minutes of inactivity. You can change this value by editing the deployment.xml configuration file, as described in the WebSphere Application Server Information Center.

- **Save your work to the master repository.**

  Until you save your changes to the master repository, the console uses a local workspace to track the changes. To save your changes, click **System Administration > Save Changes to Master Repository** to display the Save page, and then click **Save**.

Starting and stopping the administrative console

To access the administrative console, you must start it and then log in. After you finish working in the console, save your work and log out.

Before you begin

Ensure you have started the application server required by the administrative console.

About this task

To start the console and log in, use the following procedure.

Procedure
1. Start the administrative console:
   a. Enable cookies in the Web browser that you plan to use to access the administrative console.
   b. Optional: Enable JavaScript. JavaScript™ enablement is recommended so that all the features of the administrative console are available to you.
   c. In your cookie-enabled Web browser, type the following:

      `http://`*your_fully_qualified_server_name*`:`*portNumber*`/ibm/console`

      where *your_fully_qualified_server_name* specifies the fully qualified host name for the machine that contains the administrative server and *portNumber* is the administrative console port number. When the administrative console is on the local machine, *your_fully_qualified_server_name* can be `localhost` unless security is enabled.

      On Windows® platforms, use the actual host name if `localhost` is not recognized.

      If security is enabled, your request is redirected to https://*your_fully_qualified_server_name:secure_portNumber*/ibm/console, where *your_fully_qualified_server_name* is the fully qualified host name for the machine that contains the administrative server and *secure_portNumber* is the administrative console secure port number.

      Note: The default port number for an unsecure administrative console is port 9060, and for a secure administrative console the default port number is 9043. Each new administrative console that you deploy during profile creation is assigned a new unsecure port number and, if you enable security during profile creation, a new secure port number.
   d. Check the System.Out.log file of the server that runs the console application to verify that the console application has started successfully. A successful start produces the message `WSVR0221I: Application started: isclite`.

      If you are unable to start the console because the console port conflicts with an application that is already running on the machine, change the port number in the following files:
      - *profile_root*/config/cells/*cell_name*/nodes/*node_name*/serverindex.xml
      - *profile_root*/config/cells/*cell_name*/virtualhosts.xml

      Change all occurrences of the port selected during profile creation (by default, 9060) to the port for the console. Alternatively, shut down the other application that uses the conflicting port before starting the administrative console.

      `i5/OS`   If you cannot start the administrative console because the console port conflicts with an application that is already running on the machine, use the chgwassvr script command to change the port number. See Changing the ports associated with an application server for further information. Alternatively, shut down the other application that uses the conflicting port before starting the application server.

   The administrative console loads in the browser, displaying a login page.
2. Log into the console:

a. In the **User ID** field, enter your user name or user ID. The user ID lasts only for the duration of the session for which it is used to log in.

   Note: If you enter an ID that is already in use (and in session) you are prompted to do one of the following:

   - Log out the other user with the same user ID. You can recover changes made during he other user's session.
   - Return to the login page and enter a different user ID.

   Any changes made to server configurations are saved to the user ID. Server configurations are also saved to the user ID if a session times out.

b. If security is enabled for the console, you must also enter a password in the **Password** field.

c. Click **OK**.

   The administrative console now displays the Welcome page.

3. Log off the console:

   - To save the work you have done during this session, click **System administration > Save changes to master repository > Save**, and then click **Logout** to exit the console.
   - To exit the console without saving your changes to the repository, click **Logout**.

   If you close the browser before saving your work, you can recover any unsaved changes the next time that you log in with the same user ID.

   Related tasks

   "Starting and stopping servers and clusters" on page 42
   Servers must be running before you can start applications on them. There are several methods of starting servers, and the choice of methods is different for stand-alone servers and managed servers. With managed servers, the node agent must be running before you can start the servers. You can start managed servers from the administrative console of the deployment manager. If you have deployment environments or clusters, you can start or stop all of the servers in one action, from the administrative console of the deployment manager.

   "Starting a stand-alone server" on page 43
   The server process must be started before you can use the administrative console or run applications on the server.

   "Stopping a node agent" on page 48
   The node agent of a managed node is a server process that you may need to stop.

   "Restarting a node agent" on page 49
   The node agent of a managed node is a server process that you may need to restart.

   "Starting and stopping deployment environments" on page 50
   You can start or stop deployment environments based on IBM-supplied patterns directly from the administrative console.

   "Starting a cluster" on page 51
   You can start all the servers in a cluster (cluster members) in one action. Ripplestart first stops and then starts each server in turn. When you start a cluster you automatically enable workload management.

   "Stopping a cluster" on page 51
   You can stop all the servers in a cluster (cluster members) in one action.

"Starting a server" on page 52

The server process must be started before you can run applications on a managed server. You can start a single server, even if the server is a member of a cluster.

"Stopping a server" on page 54

The server process must be stopped if want to make configuration changes to the server or any of the modules deployed to it. You can stop a single server, even if the server is a member of a cluster.

Setting administrative console preferences

The display of data on a collection page (a page that lists collections of data or resources in a table) can be customized through administrative console preferences. Preferences are set on a user level, and typically must be set separately for each area of the administrative console.

About this task

You can set the following display preferences for collection pages:

- **Maximum rows**: Specifies the maximum number of rows that are displayed when the collection is large. If there are more rows than the specified maximum, they are displayed on subsequent pages. The default value is 20.
- **Retain filter criteria**: Specifies whether the last search criteria entered in the filter function is retained. If this is enabled, the console collection pages initially use the retained filter criteria to display the data in the table following the preferences. See "Setting administrative console filters" on page 19 for more information.
- **Max result set size**: Specifies the maximum number of resources that a search can return. The default value is 500.
- **Max column width**: Specifies the maximum number of characters viewable in a collection column. The default value is 18.

Perform the following steps to set display preferences for a collection page:

Procedure

1. From any collection page, click **Preferences**.

   The page expands to display the preference fields.
2. Modify the values for the **Maximum rows**, **Retain filter criteria**, **Max result set size**, and **Maximum column width** fields as desired.
3. Click **Apply**.

   The collection table is refreshed to display according to the values you specified.

What to do next

You can also set global administrative console preferences, such as whether the workspace is automatically refreshed and which scope to use by default. To access the Preferences page in the administrative console, click **System administration > Console settings > Preferences**. See the WebSphere Application Server Information Center for documentation on setting these preferences.

Setting administrative console filters

Each table on a collection page in the administrative console displays a list of WebSphere Process Server data or resources. You can use a filter to specify exactly which resources or data to display in a particular column of the table. Filters can be set on a single column only.

Procedure

1. From the buttons at the top of the table, click **Filter the view**.

   The filter dialog box opens above the top row of the table.

2. Use the **Filter** drop-down menu to select the column you want to include in the filter.

3. In the **Search term(s)** field, specify the filter criteria.

   The criteria is a string that must be found in the name of a table entry in order for it to be displayed. The string can contain the percent sign (%), asterisk (*), or question mark (?) symbols as wildcard characters. For example, on the Resource Adapters page, you can enter *JMS* as the filter criteria for the Name column to find any resource adapter whose name contains the string JMS.

   Prefix each of the following characters that appear as part of the string with a backslash (\) so that the regular expression engine performing the search correctly matches the search criteria: ( ) ^ * % { } \ + & .

   For example, if you want to search for all Java DataBase (JDBC) providers containing (XA) in the provider name, specify the following string in the Search term(s) field:

   \(XA\)

4. Click **Go**.

   The table refreshes, and only those items in the selected column that meet the filter criteria are displayed.

Using My tasks

Customize console navigation by creating and editing a task view.

About this task

Use **My tasks** to create and edit a list of tasks to view in the console navigation. A task includes a page that contains one or more Web applications, or console modules, that are used to complete that task. When you first access the console, all tasks to which you have access are displayed in the navigation. My tasks is especially useful to customize the navigation to show only the tasks you use most often. After you customize your tasks, **My tasks** is initially displayed each time you log in to the console.

Procedure

1. Click the **Welcome** link in the navigation tree.
2. Select **My tasks** from the **View** selection list in the navigation. If you have never used **My tasks** before, you must click **Add tasks** to open it.
3. Select the tasks you want to add to **My tasks** list.
4. To save your changes, click **Apply**.
5. To cancel your changes, click **Reset**.

Results

After you click **Apply**, your customized task list is displayed in the navigation. You do not have to shut down and restart the administrative console.

Getting started with Business Process Choreographer Explorer

Depending on your user role, you can use Business Process Choreographer Explorer to manage business processes and human tasks, or work with your assigned tasks.

About this task

You can use Business Process Choreographer Explorer to perform the following tasks:

- If you are a business administrator, you can manage the lifecycle of business processes, and you can repair business processes. For example, you can restart or force the completion of single activities, or compensate the business process as a whole. If compensations failed, you can retry, skip or stop the process instances. In addition, you can add and update custom properties for business processes and activities.
- If you are a human task administrator, you can manage the life cycle of human tasks, and manage work assignments. For example, you can assign responsibility to users, or manage absence handling and substitutes for users. You can also change the priority and business category for human tasks, and add or update custom properties.
- If you are a business user, you can use Business Process Choreographer Explorer to work with your assigned tasks. For example, you can initiate business processes, services, and human tasks, and you can work on, edit, save, complete, or release human tasks. In addition, you can flag your absence and define substitutes.

Furthermore, Business Process Choreographer Explorer offers a search function that you can use to discover business processes and their related activities and human tasks that need attention. For example, you can check the status of these instances, navigate between related instances and templates, and retrieve a graphical view of the process states which includes the associated activities and human tasks.

Related tasks

Administering task templates and task instances

Use the administrative console or the administrative commands to administer task templates. Use Business Process Choreographer Explorer to work with task instances.

Managing work assignments

After a task has started, you might need to manage work assignments for the task, for example, to better distribute the work load over the members of a work group.

Creating and starting a task instance

You can create and start a task instance from any of the task templates that you are authorized to use.

Working on your tasks

To work on a task, you must claim the task and then perform the actions that are needed to complete it.

Business Process Choreographer Explorer user interface

Business Process Choreographer Explorer is a stand-alone Web application that provides a set of administration functions for managing business processes and human tasks. The interface consists of a taskbar, a navigation pane, and the workspace.

The following figure shows the layout of the Business Process Choreographer Explorer user interface.

Taskbar	
Navigation pane	**Page title**
	Action bar
	Start Stop Complete
	Content...
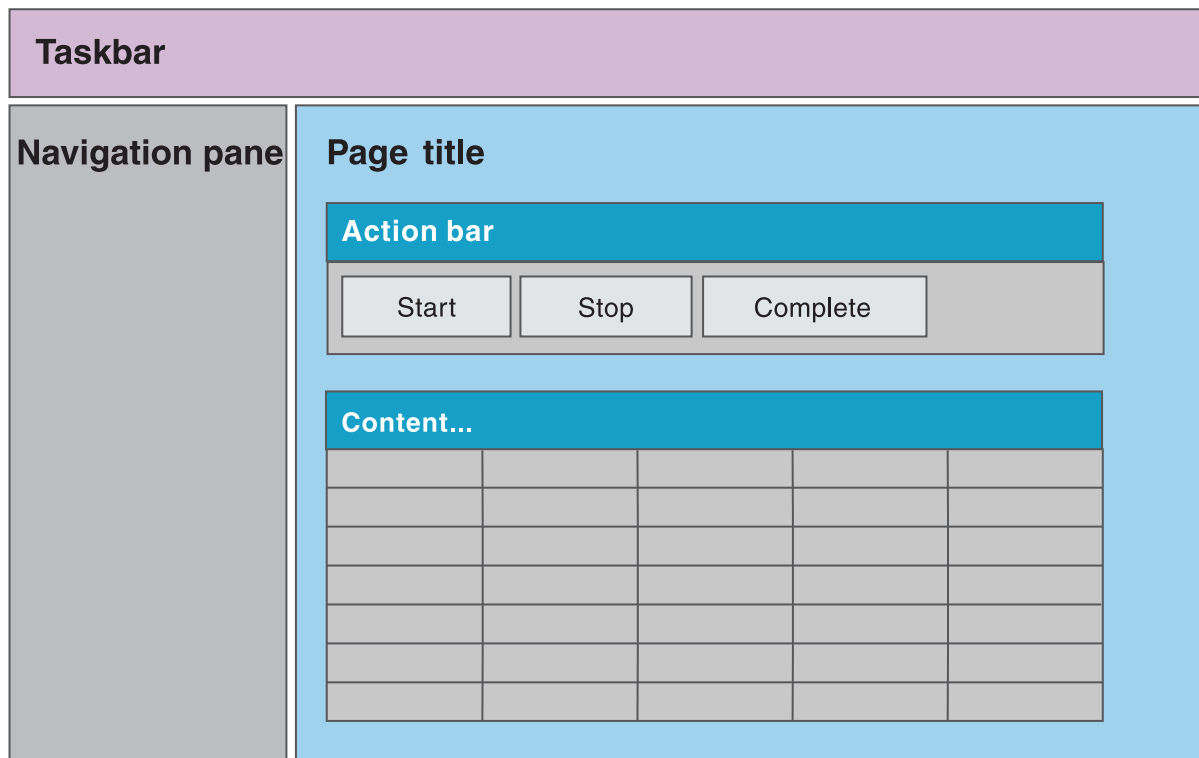
The user interface has the following main areas.

Taskbar

For all users, the taskbar offers options for logging out of Business Process Choreographer Explorer, specifying absence settings, and accessing online help. If you have system administrator rights, the taskbar also includes the following options:

Customize
Select this option to add views to and remove views from the navigation pane for this instance of Business Process Choreographer Explorer. You can also define the view that your users see when they log in.

Define views
Select this option to define customized views for your user group.

Define Substitutes
Select this option to define absence settings for users.

Navigation pane

The navigation pane contains links to views that you use to administer objects, for example, process instances that you started, or human tasks that you are authorized to administer. The default user interface contains links to predefined views for business processes and tasks.

The system administrator can customize the content of the default navigation pane by adding and removing predefined views from the navigation pane and defining custom views to add to the navigation pane. All users can define personalized views from the navigation pane.

Workspace

The workspace contains pages that you use to view and administer business-process and human-task related objects. You access these pages by clicking the links in the navigation pane, by clicking an action in the action bar, or by clicking links within the workspace pages.

Business Process Choreographer Explorer navigation pane

Use the navigation pane to access views that you use to administer business process and human task objects, such as process instances and work assignments. The default user interface contains links to predefined views for business processes and tasks. You can also define your own personalized views, which are added to the navigation pane. In addition, if you are a system administrator, you can define customized views that are available to all users.

Available actions

The following actions are available in the navigation pane:
- Navigate to a view.

  Click the view name to navigate to that view.
- Collapse and expand a group.

  Click the arrow beside an item in the navigation pane to expand or collapse the item.
- Define a new search.

  Click the **New Search** icon ( ), to search for objects, or to define a personalized view.

Additional actions are available from the pop-up menu depending on the view type. An icon indicates whether a pop-up menu is available.

- To delete the view, click the **Delete** icon ( ).

- To modify the view, click the **Edit** icon ( ).

- To create a copy of the view and modify the copy, click the **Copy** icon ( ).

- To move the view up or down in the list, click the **Up** icon ( ) or the **Down** icon ( ).

Predefined views in the navigation pane

The default navigation pane contains the following groups of views. The views that are shown in the navigation pane in your Business Process Choreographer Explorer might differ depending on whether your system administrator has added views to, or removed views from the navigation pane. If no view is defined for a group of views, the group is not displayed.

Process templates

The process templates group contains the following view:

My Process Templates

This view shows a list of process templates. From this view you can display information about the process template and its structure, display a list of process instances that are associated with a template, and start process instances.

Process instances

The process instances group contains the following views:

Started By Me

This view shows the process instances that you started. From this view, you can monitor the progress of the process instance, and list the activities, processes, or tasks that are related to it.

Administered By Me

This view shows the process instances that you are authorized to administer. From this view, you can act on the process instance, for example, to suspend and resume a process, or monitor the progress of the activities in a process instance.

Critical Processes

This view shows process instances in the running state that contain activities in the stopped state. From this view, you can act on the process instances, or list the activities and then act on them.

Terminated Processes

This view shows process instances that are in the terminated state. From this view, you can act on these process instances.

Failed Compensations

This view shows the compensation actions that have failed for microflows.

Activity instances

By default, the activity instances group does not contain any views. Therefore this group is not displayed in the default navigation pane.

Task templates

The task templates group contains the following view:

My Task Templates

This view shows a list of task templates. From this view you can create and start a task instance, and display a list of task instances that are associated with a template.

Task instances

The task instances group contains the following views:

My To-dos

This view shows a list of the task instances that you are authorized

to work with. From this view, you can work on a task instance, release a task instance that you claimed, or transfer a task instance to another user.

All Tasks
This view shows all of the tasks for which you are the owner, potential owner, or editor. From this view, you can work on a task instance, release a task instance that you claimed, or transfer a task instance to another user.

Initiated By Me
This view shows the task instances that you initiated. From this view, you can work on a task instance, release a task instance that you claimed, or transfer a task instance to another user.

Administered By Me
This view shows the task instances that you are authorized to administer. From this view, you can act on the task instance, for example, to suspend and resume a process, to create work items for the task instance, or to display a list of the current work items for the task instance.

My Escalations
This view shows all of the escalations for the logged on user.

View types

The navigation pane can contain the following types of views. Depending on the view, additional actions are available from the pop-up menu.

Predefined views in the default navigation pane.
These groups of views are available only if the navigation pane has not been changed by the system administrator in the Customize Navigation Tree and Login View page. A pop-up menu is not available for these views.

Customized views and predefined views that were added to the navigation pane by the system administrator.
Business users can click the view name and navigate to the view. For system administrators, pop-up menus are available.

- The predefined views are indicated by the **Predefined view** icon: . A system administrator can use the pop-up menu to change the position of these views in the navigation pane.

- The customized views are indicated by the **Custom view** icon: . A system administrator can delete, edit, copy, and move these views.

Personalized views.
These views are indicated by the **Custom view** icon: . These views are only visible to the user who created the views. The user can delete, edit, copy, and move the views.

Starting Business Process Choreographer Explorer

Business Process Choreographer Explorer is a Web application that can be installed as part of the configuration of the business process container. Before you can start using Business Process Choreographer Explorer from a Web browser, you must have installed the business process container, human task container, and the Business Process Choreographer Explorer application, and the application must be running.

About this task

To start Business Process Choreographer Explorer, complete the following steps.

Procedure

1. Direct your Web browser to the Business Process Choreographer Explorer URL.

   The URL takes the following form. The value of the URL depends on how the virtual host and context root were configured for your installation.

   `http://app_server_host:port_no/context_root`

   Where:

   app_server_host
   > The network name for the host of the application server that provides the business process application with which you want to work.

   port_no
   > The port number used by Business Process Choreographer Explorer. The port number depends on your system configuration. The default port number is `9080`

   context_root
   > The root directory for the Business Process Choreographer Explorer application on the application server. The default is `bpc`.

2. If security is enabled, you must enter a user ID and password, then click **Login**.

Results

The initial page of the Business Process Choreographer Explorer is displayed. By default, this is the page that shows the My To-dos view.

Customizing Business Process Choreographer Explorer

Business Process Choreographer Explorer provides a user interface for administrators to manage business processes and human tasks, and for business users to work with their assigned tasks. Because this is a generic interface, you might want to customize the interface for a specific Business Process Choreographer Explorer instance to address the business needs of user groups that are assigned to this instance.

About this task

You can customize the user interface in various ways.

Customizing the Business Process Choreographer Explorer interface for different user groups

The navigation pane in the default Business Process Choreographer Explorer user interface contains a set of links to predefined views. The My To-dos view is the default view that is shown after you log in. If you have one of the Business Process Choreographer system administrator roles, you can customize the links that are shown in the navigation pane, the view that your users see after they log in, and the information that is shown in the views.

Before you begin

To customize the interface, you must have BPCSystemAdministrator authorization.

About this task

For example, the default user interface for Business Process Choreographer Explorer does not include views for working with business state machines. You can add predefined views to work with process templates and process instances for business state machines.

Or, you might want to offer users that deal with customer orders a different interface to the one that you offer the users dealing with customer service enquiries. You can customize an instance of Business Process Choreographer Explorer so that it meets the workflow patterns of those users who are assigned to the instance.

To customize the default user interface of Business Process Choreographer Explorer, complete the following steps.

Procedure
1. Customize the set of views in the navigation pane and the default login view.
   a. Click **Customize** in the taskbar.
   b. In the Customize Navigation Tree and Login View page, select the views to include in and deselect the views to remove from the navigation pane.
   c. Select the view that your users see when they log into Business Process Choreographer Explorer.

      The list contains the views that you selected in the previous step and any customized views that you created from the Search And Define Customized Views page (see step 2).
   d. To save your changes, click **Save**.

      To return the views for this instance to the default views, click **Restore defaults**. This action resets the navigation pane to the list of predefined views. Customized views in the navigation pane are not affected by this action.
2. Customize the views.

   You can specify the information that is shown in the views for this Business Process Choreographer Explorer instance.
   a. Click **Define Views** in the taskbar.
   b. In the Search And Define Customized Views page, select the type of view that you want to customize, for example, process templates.
   c. In the Search And Define Customized Views page for the view, specify the search criteria.
   d. Use the View Properties tab to select the properties to include in the view, and to specify the list properties.

      If this is a task instance view or a process instance view, click **View Settings** to select a set of actions to add to the action bar in the view, and to specify the items that are included in the view for system administrators and system monitors.
      - Select the view type:
        - To add administrative actions to the action bar in the view, select **Manage Instances**.
        - To add a set of actions to the action bar that enables the logged-on user to work with the instances, select **Work with Instances**.
      - For system administrators and system monitors, you can limit the search result to their own instances:

– To show all items that match the search criteria in the view, select **All Instances**. All of the items are shown regardless of whether the system administrator has work items for these items.

– To show only the items that the logged-on user has work items for, select **Personal Instances**.

e. Enter a display name for the view in the **View Name** field and click **Save**.

The new view appears in your navigation pane. Users see the new view when they next log into Business Process Choreographer Explorer.

Defining views for process templates for business state machines:

Although a predefined view is provided for the process templates for business state machines, you might want to define your own views for this type of template.

Before you begin

To create customized views, you must have BPCSystemAdministrator authorization.

Procedure
1. Click **Define Views** in the taskbar.
2. In the Search and Define Customized Views page, select **Search For Process Templates And Define Customized Views**.
3. Click **Property Filters** → **Custom Property Filter**.
   a. Add a custom property with the following settings:
      * In the **Property Name** field, type generatedBy.
      * In the **Property Value** field, type BusinessStateMachine.
   b. Click **Add**.
   c. Add other custom properties as needed.
4. Click **View Properties** → **List Columns**.
   a. In the List Columns for Custom Properties, add a custom property with the following settings:
      * In the **Property Name** field, type generatedBy.
      * In the **Display Name** field, type a display name for the column, and click **Add**.
   b. Add other columns to or remove columns from the list of selected columns.
5. Type a display name for the query in the **View Name** field, and click **Save**.

Results

By default, a link to the new view is added to the Process Templates group in the navigation pane. Your users see this view the next time they log in to Business Process Choreographer Explorer.

Defining views for process instances for business state machines:

Although a predefined view is provided for the process instances for business state machines, you might want to define your own views for this type of process instance.

Before you begin

To create customized views, you must have BPCSystemAdministrator authorization.

Procedure

1. Click **Define Views** in the taskbar.
2. In the Search and Define Customized Views page, select **Search For Process Instances And Define Customized Views**.
3. Click **Property Filters** → **Custom Property Filter**.
   a. Add a custom property with the following settings:
      - In the **Property Name** field, type `generatedBy`.
      - In the **Property Value** field, type `BusinessStateMachine`.
   b. Click **Add**.
   c. Add other custom properties as needed.
4. Click **View Properties** → **List Columns**.
   a. In the List Columns for Query Properties, add the following query properties.
      - To add business state information to the view, type `name` in the **Property Name** field, `DisplayState` in the **Variable Name** field, and `tns` in the **Namespace** field, where `tns` is the target namespace of the business state machine suffixed by *-process*. Also specify a display name for the column in the **Display Name** field, and click **Add**.
      - To add correlation information to the view, provide the appropriate information in the **Property Name** field, the **Variable Name** field, and the **Namespace** field. These values are derived from the definition of the business state machine. Also provide a display name for the column in the **Display Name** field.

      Property Name
      > The name of the correlation property that you defined for the business state machine.

      Variable Name
      > If the correlation set is initiated by incoming parameters, the variable name has the following format:
      >
      > *operation_name*`_Input_`*operation_parameter_name*
      >
      > where *operation_name* is the name of the operation for the transition out of the initial state.
      >
      > If the correlation set is initiated by outgoing parameters, the variable name has the following format:
      >
      > *operation_name*`_Output_`*operation_parameter_name*

      Namespace
      > The namespace of the query property, where `tns` is the target namespace of the business state machine suffixed by *-process*.

   b. Add other custom properties or query properties, or add columns to or remove columns from the list of selected columns.
5. Type a name for the query in the **View Name** field, and click **Save**.

Results

By default, a link to the new view is added to the Process Instances group in the navigation pane. Your users see this view the next time they log in to Business Process Choreographer Explorer.

Personalizing the Business Process Choreographer Explorer interface

The navigation pane in the default Business Process Choreographer Explorer user interface contains a set of links to predefined views and views that are defined by your system administrator. You can add your own views to your navigation pane, for example, to monitor the progress of a specific task or process.

About this task

In Business Process Choreographer Explorer, complete the following steps to personalize your user interface.

Procedure

1. In the section of the navigation tree where you want to define the new view,

   click the **New search** icon ( ).

2. In the Search and Define Personalized Views page for the view, specify the search criteria.

3. Use the View Properties tab to select the properties to include in the view, and to specify the list properties.

   If this is a task instance view or a process instance view, click **View Settings** to select a set of actions to add to the action bar in the view. If you are a system administrator and system monitor, specify the items that are included in the view.

   - Select the view type:
     – To add administrative actions to the action bar in the view, select **Manage Instances**.
     – To add a set of actions to the action bar that enables the logged-on user to work with the instances, select **Work with Instances**.
   - If you are a system administrator and system monitor, you can limit the search result to your own instances:
     – To show all items that match the search criteria in the view, select **All Instances**. All of the items are shown regardless of whether the system administrator has work items for these items.
     – To show only the items that the logged-on user has work items for, select **Personal Instances**.

4. Enter a display name for the view in the **View Name** field and click **Save**.

Results

The new view appears in your navigation pane.

Changing the appearance of the default Web application

Business Process Choreographer Explorer provides a ready-to-use Web user interface based on JavaServer Pages (JSP) files and JavaServer Faces (JSF) components. A cascading style sheet (CSS) controls how the Web interface is rendered. You can modify the style sheet to adapt the user interface to fit a certain look and feel without writing any new code.

Before you begin

Style sheet modification requires profound knowledge about cascading style sheets.

About this task

You can change the CSS, for example, so that the default interface conforms to guidelines for corporate identity.

Procedure

Modify the style sheet. The default style sheet, `style.css`, contains styles for the elements in the header, the navigation pane, and the content pane.

Related concepts

"Business Process Choreographer Explorer user interface" on page 21
Business Process Choreographer Explorer is a stand-alone Web application that provides a set of administration functions for managing business processes and human tasks. The interface consists of a taskbar, a navigation pane, and the workspace.

Styles used in the Business Process Choreographer Explorer interface:

The `style.css` file contains styles that you can change to adapt the look and feel of the default user interface.

The `style.css` file contains styles for the following elements of the default user interface:
* "Banner"
* "Footer" on page 31
* "Menu bar" on page 31
* "Login page" on page 31
* "Navigator" on page 31
* "Content panels" on page 32
* "Command bar" on page 32
* "Lists" on page 32
* "Details panel" on page 32
* "Message data" on page 32
* "Tabbed panes" on page 33
* "Search pages" on page 33
* "Error details" on page 33

This file is in the following directory:

`<profile_root>\installedApps\<node_name>\<explorer_instance>\bpcexplorer.war\theme`

Banner

Style name	Description
.banner	The division for the banner.
.banner_left	A division in the banner. It is used to embed the title image of the application.
.banner_right	A division in the banner. You can use it, for example, to display further logos.

Footer

Style name	Description
.footer	The division for the footer.
.footer_left	A division in the footer, for example, you can use it to display the company logo for the application.
.footer_right	A division in the footer, for example, you can use it to display further logos.

Menu bar

Style name	Description
.menubar	The JSF subview.
.menuContainer	The container panel including the menu items, for example, labels, and links.
.menuItem	An item on the menu bar.

Login page

Style name	Description
.loginPanel	The panel containing the login form.
.loginTitle	The title on the form.
.loginText	The instructional text.
.loginForm	The form that contains the input controls.
.loginValues	The table that determines the layout of the controls.
.loginField	The labels used for the logon fields, for example, Name or Password.
.loginValue	The text input field.

Navigator

Style name	Description
.pageBodyNavigator	The area that contains the navigator.
.navigator	JSF subview for navigator which contains the links to the lists.
.navigatorTitle	The title for each navigator box.
.taskNavigatorTitle	A class of titles for navigation boxes. They are used to distinguish between links to lists of business process objects and human task objects.
.navigatorFrame	The division for each navigator box, for example, to draw a border.
.navigatorLink	A link in the navigator box.
.expanded	Used when the navigator boxes are expanded.
.collapsed	Used when the navigator boxes are collapsed.

Content panels

Style name	Description
.pageBodyContent	The area that contains the content.
.panelContainer	The division panel that contains the list, details or messages.
.panelTitle	The title for the displayed content, for example, My To-dos.
.panelHelp	The division container that contains the help text and the icon.
.panelGroup	The division container that contains the command bar and list, details or message.

Command bar

Style name	Description
.commandbar	The division container around the command-bar area.
.button	The style that is used for buttons in the command bar.

Lists

Style name	Description
.list	The table that contains the rows.
.listHeader	The style used in the header row of the list.
.ascending	Style for the list header class when the list is sorted by this column in ascending order.
.descending	Style for the list header class when the list is sorted by this column in descending order.
.unsorted	Style for the list header class when the list is not sorted by this column.

Details panel

Style name	Description
.details	The division container around a details panel.
.detailsProperty	The label for a property name.
.detailsValue	The text for a property value.

Message data

Style name	Description
.messageData	The division container around a message.
.messageDataButton	Button style for Add and Remove buttons in the message form.
.messageDataOutput	For rendering read-only text.
.messageDataValidInput	For message values that are valid.

Style name	Description
.messageDataInvalidInput	For message values that are not valid.

Tabbed panes

Style name	Description
.tabbedPane	The division container around all of the tabbed panes.
.tabHeader	The tab header of a tabbed pane.
.selectedTab	The active tab header.
.tab	The inactive tab headers.
.tabPane	The division container that encloses a tabbed pane.
.tabbedPaneNested	The division container around nested tabbed panes used on the search pages.
.tabHeaderSimple	The tab header of a nested tabbed pane.
tabHeaderProcess	The tab header of a nested tabbed pane for process filters.
.tabHeaderTask	The tab header of a nested tabbed pane for task filters.
.tabPaneSimple	The division container that encloses a nested tabbed pane.

Search pages

Style name	Description
.searchPane	The tabbed pane for a search panel. See also tabbed panes.
.searchPanelFilter	The table container for a search form.
.searchLabel	The label for a search form control.
.summary	The container that encloses the search summary pane.
.summaryTitle	The common style for all titles on the search summary pane.
.summaryTitleProcess	A style for the title of process related sections on the search summary pane.
.summaryTitleTask	A style for the title of task related sections on the search summary pane.

Error details

Style name	Description
.errorPage	The tabbed pane for an error page.
.errorLink	Styles uses to render the button links on a page.
.errorDetails	Tabbed pane with error details.
.errorDetailsStack	Tabbed pane with an exception stack.
.errorDetailsMessage	Text style for error message.

Getting started with Business Process Choreographer Observer

While business processes and tasks are running, WebSphere® Process Server can emit events that contain information about state changes of process instances and their related activities. Use Business Process Choreographer Observer to retrieve statistical information based on these events and create reports on processes and activities.

About this task

You can define your own reports, or use a drill-down approach to get more detailed information on specific process instances, activity instances, or inline human tasks. In addition, you can export the reported results for further external processing.

Business Process Choreographer Observer bridges a gap between IT-level monitoring and business level monitoring. By providing means for reporting on events in the Business Flow Manager component it helps you to understand what is happening in Business Process Choreographer.

> **Related concepts**
>
> "Business Process Choreographer Observer user interface"
> Business Process Choreographer Observer is a stand-alone Web application that provides a set of functions for reporting on process and activity events. The interface consists of a taskbar, a navigation pane, and the workspace.
> "Business Process Choreographer Observer navigation pane" on page 35
> Use the navigation pane to select the kind of report that you want to create, such as process or activity reports. You can also store your own report definitions and add these to the navigation pane.
>
> **Related tasks**
>
> ⮊ Reporting on business processes and activities
> During the processing of business processes and activities, an event can be generated when the process, activity, or task changes state. These events are stored and made available for creating reports using Business Process Choreographer Observer, for example, to analyze process bottlenecks, or to evaluate the reliability of a service that is called from an activity.

Business Process Choreographer Observer user interface

Business Process Choreographer Observer is a stand-alone Web application that provides a set of functions for reporting on process and activity events. The interface consists of a taskbar, a navigation pane, and the workspace.

The following figure shows the layout of the Business Process Choreographer Observer user interface.

Taskbar	
Navigation pane	**Page title**
	Content......
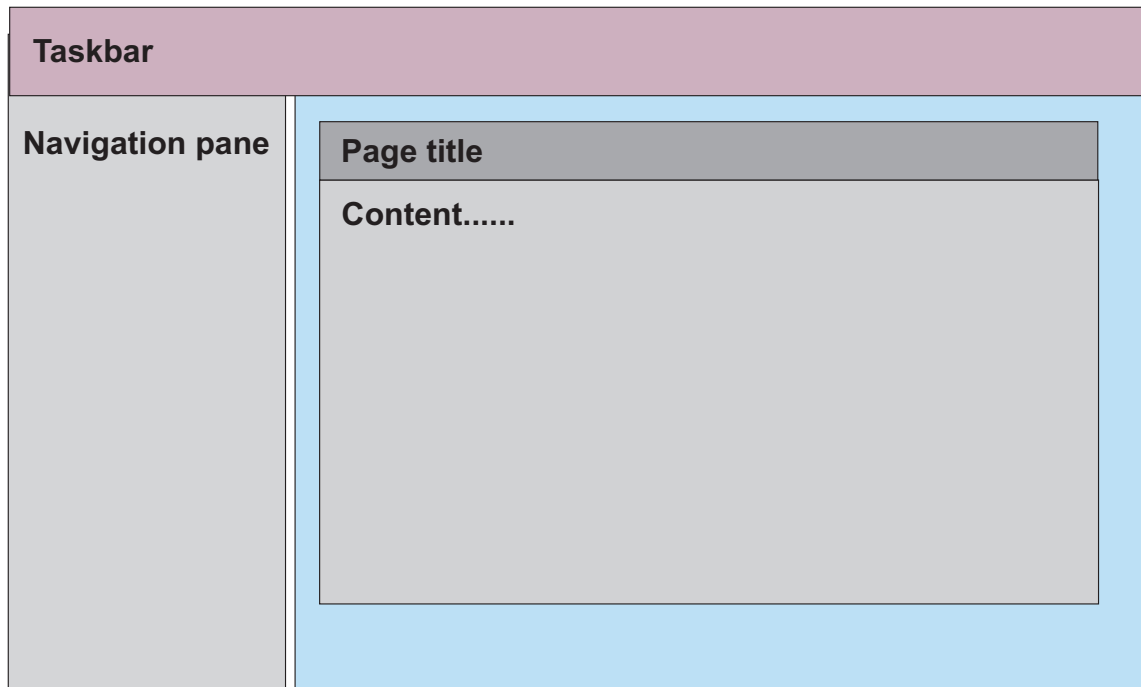
The user interface has the following main areas.

Taskbar

The taskbar offers options for logging out of Business Process Choreographer Observer, and a link to the general Help page.

Navigation pane

The navigation pane on the left side of the user interface contains links that you use to select the kind of report that you want to create, for example, you can view the data for an activity instance in a chart.

Workspace

The workspace on the right side of the user interface contains pages that you use to specify report definitions and view reports. To access these pages click the links in the navigation pane. For information about a page click the **Help** icon [i] on the respective page.

Related tasks

"Getting started with Business Process Choreographer Observer" on page 34 While business processes and tasks are running, WebSphere® Process Server can emit events that contain information about state changes of process instances and their related activities. Use Business Process Choreographer Observer to retrieve statistical information based on these events and create reports on processes and activities.

Business Process Choreographer Observer navigation pane

Use the navigation pane to select the kind of report that you want to create, such as process or activity reports. You can also store your own report definitions and add these to the navigation pane.

Available actions

The following actions are available in the navigation pane:

- Collapse and expand a group.

  Click the plus sign (+) beside an item in the navigation pane to expand it, or click the minus sign (-) to collapse the item.
- Navigate to a predefined list or chart.

  Click the kind of instance that you want to report.
- Navigate to the process or activity report wizard.

  Click the **New report** icon ( ) to specify the type of report, the report content and the filter criteria for a report.
- Run a saved process or activity report.

  Click the report name to run the report.
- Open the pop-up menu of a saved process or activity report definition.

  Click the **Show pop-up menu** icon ( ) to work on a saved report definition.

  - To edit the report definition, click the Edit icon ( ).

  - To copy the report definition, click the Copy icon ( ).

  - To delete the report definition, click the Delete icon ( ).

  - To export the report result, click the Export icon ( ).

  - To run a search asynchronously, click the **Asynchronous Search** icon ( ).
    - After the asynchronous search completes successfully, the **Asynchronous Search Completed** icon ( ) is displayed in the navigation pane. Click the name of the report to view your search results.
    - If the asynchronous search does not complete successfully, the **Asynchronous Search Failed** icon ( ) is displayed.

Predefined lists and charts in the navigation pane

The navigation pane contains the following groups of predefined lists and charts.

Lists   This group contains the following lists:

Processes

Use this list to view processes that emitted a process event during the specified time frame. The processes are listed according to the process state.

Activities

Use this list to view the state that the selected activities reached during the specified time frame. The activities are listed according to the activity state.

Users

Use this list to view the activities that the selected users performed during the specified time frame, and the state the activities reached. The activities are displayed according to their state. The corresponding user for each activity is shown.

> **Attention:** In countries where gathering data on the work performance of employees violates privacy and data protection laws, the process model must be defined to not emit activity events for individual users.

Charts This group contains the following charts:

Process snapshot

Use this chart to check how many process instances are in the different states at the specified time. You can view the data in a bar chart, or in a pie chart.

Processes by period

Use this chart to check the distribution of the number of process instances that reached the specified state during a specified period. Each instance is shown in the time slice in which it reached the specified state. You can view the data in a line, bar, or pie chart

Activity snapshot

Use this chart to check how many activity instances are in the different states at the specified time. You can view the data in a bar chart, or in a pie chart.

Activities by period

Use this chart to check the distribution of the number of activity instances that reached the specified state during a specified period. Each instance is shown in the time slice in which it reached the specified state. You can view the data in a line, bar, or pie chart.

Process and activity reports

The navigation pane links to the following report wizards. The report wizard is indicated by the **New report** icon: .

Process reports

Use process reports to query process instance events. These events describe the state changes of process instances. Use the report wizard to define the data for your reports. You can save and retrieve your report definitions.

Activity reports

With an activity report, you query activity instance events. These events describe state changes of activity instances. Use the report wizard to specify individual reports. You can store and retrieve your report definitions.

Related tasks

"Getting started with Business Process Choreographer Observer" on page 34
While business processes and tasks are running, WebSphere® Process Server can emit events that contain information about state changes of process instances and their related activities. Use Business Process Choreographer Observer to retrieve statistical information based on these events and create reports on processes and activities.

Chapter 3. Administering servers and clusters

Administering servers includes creating them, configuring them to support Service Component Architecture (SCA) modules, and starting and stopping them.

Creating a server

To create a server in a managed node, use the Create New Application Server wizard in the administrative console or the wsadmin createApplicationServer command.

Before you begin

You must have created a custom profile and federated it to a deployment manager to create a managed node.

About this task

These instructions are specifically for creating servers that are managed by a deployment manager. To create a stand-alone server, simply create a stand-alone server profile.

Instead of using the administrative console, you can use other methods to create the servers, such as the wsadmin createApplicationServer command. For more information, see Commands for the AdminTask object.

Procedure

1. In the administrative console, click **Application Servers** ↠ **New** to start the **Create New Application Server** wizard.
2. Follow the instructions in Creating application servers. On the **Select a server template** panel, select the **defaultProcessServer** template, or a suitable user-defined template.
3. **Optional:** If the server will run applications that contain business processes or human tasks, configure Business Process Choreographer. See Configuring Business Process Choreographer. for more information.
4. **Optional:** install the SchedulerCalendars application on one or more servers as described in Configuring Business Process Choreographer.

Results

You can now start the server and deploy modules to it.

Configuring SCA support for a server or cluster

Use the Service Component Architecture console page to enable a server or cluster in a network deployment environment to host service applications, their required messaging engines and destinations, or both.

Before you begin

Before configuring SCA support, determine the following:

- Whether you are using a stand-alone server profile. If so, SCA support is already configured and you cannot use the Service Component Architecture page to remove that support; however, you can use this page to modify some properties for database data sources.
- Where to host the messaging engines and destinations (use either a local or remote bus member).
- Whether you need to configure the SCA system bus only, or whether you also need to configure the SCA application bus. The application bus is configured by default and is required if you plan to deploy SCA applications that use WebSphere Business Integration Adapters.

Security role required for this task: You must be logged in as administrator or configurator to perform the following task.

About this task

To configure SCA support on your server or cluster, perform the following steps.

Procedure

1. From within the administrative console, click one of the following, depending on your scope:
   - **Servers** → **Application Servers** → *serverName* → **Service Component Architecture**
   - **Servers** → **Clusters** → *clusterName* → **Service Component Architecture**
2. Click **Support the Service Component Architecture components**.
3. In the Bus Member Location panel, specify where you want to host the destinations and messaging engines required by the SCA applications. There are two options:
   - **Local**. Specifies that you plan to host SCA applications, destinations, and messaging engines on the current server or cluster.
   - **Remote**. Specifies that you plan to host SCA applications on the current server or cluster while hosting destinations and messaging engines on a remote server or cluster (also referred to as a *deployment target*).
4. **(Remote bus member only)** If you selected **Remote** in the previous step, specify the remote server or cluster you want to use to host application destinations and messaging engines. Use the drop-down menu to select an existing deployment target (one that is already configured as a member of the SCA system bus), or click **New** to select a new server or cluster from the Browse Deployment Target page.

   If you select a new server or cluster from the Browse Deployment Target page, the necessary messaging is automatically configured on that target when you complete the SCA configuration documented in this topic.
5. Use the table in the System Bus Member panel to verify or modify the system bus data source configuration.
   a. Verify any default values in the **Database Instance**, **Schema**, **Create Tables**, **User name Password**, **Server**, and **Provider** fields. See the online help for detailed information about these fields and the values they accept.
   b. If no default values exist in these fields, or if the default values are incorrect, enter the appropriate values for the system bus data source. You can enter values directly in the field or by clicking **Edit** and making edits on the Data Source details page.

c. Optional: Ensure that the data source can contact and authenticate with the database by clicking **Test Connection**.

6. Use the table in the Application Bus Member panel to verify or modify the application bus data source configuration.

   a. Ensure the **Enable the WebSphere Business Integration Adapter components** option is selected.

   Note: If you do not want to use the application bus, clear the **Enable the WebSphere Business Integration Adapter components** option and proceed to Step 7.

   b. Verify any default values in the **Database Instance**, **Schema**, **Create Tables**, **User name Password**, **Server**, and **Provider** fields. See the online help for detailed information about these fields and the values they accept.

   c. If no default values exist in these fields, or if the default values are incorrect, enter the appropriate values for the application bus data source. You can enter values directly in the field or by clicking **Edit** and making edits on the Data Source details page.

   d. Optional: Ensure that the data source can contact and authenticate with the database by clicking **Test Connection**.

7. Click **OK** to complete the SCA configuration.

   Related information

   Planning your deployment environment

   ➡ Managing resources for mediation modules

Considerations for Service Component Architecture support in servers and clusters

Servers and clusters can support Service Component Architecture (SCA) applications, application destinations, or both.

SCA applications (also called service applications) require the use of one or more of the automatically created service integration buses. Each application uses a set of messaging resources, which are called *destinations*. These destinations require configured messaging engines, and they can be hosted on the same server or cluster as the application or on a remote server or cluster. Messaging engines typically use database data sources; note that a file store can be used in place of a database data source in a stand-alone server profile if that option was selected during profile creation.

By default, new servers and clusters in a network deployment or managed node environment are not configured to host SCA applications and their destinations.

Note: A stand-alone server has SCA support automatically configured. You cannot disable this configuration.
To enable this support, use the Service Component Architecture page in the administrative console. For servers, ensure that the application class-loader policy is set to `Multiple`.

Before enabling SCA support for a server or cluster in a network deployment or managed node environment, determine which of the following possible configurations you want to implement:

- **Remote bus member configuration**: The server or cluster hosts SCA applications, but the destinations are hosted on a remote server or cluster. This scenario requires the remote service integration bus members to be configured with the messaging engines needed to host the destination.

  While the use of remote messaging requires initial investment in planning for and configuring the service integration bus and its members, that configuration can be reused by multiple members within the application cluster. Messages are distributed to every member. In addition, the initial configuration can be structured to provide failover support.

- **Local bus member configuration**: The server or cluster hosts both SCA applications and application destinations. The required messaging engines are configured using the local bus members on the server or cluster.

Refer to the planning topics to help you decide which configuration is appropriate for your environment.

Related information

↪ Configuring class loaders of a server

↪ Learning about service integration buses

↪ Messaging engines

Starting and stopping servers and clusters

Servers must be running before you can start applications on them. There are several methods of starting servers, and the choice of methods is different for stand-alone servers and managed servers. With managed servers, the node agent must be running before you can start the servers. You can start managed servers from the administrative console of the deployment manager. If you have deployment environments or clusters, you can start or stop all of the servers in one action, from the administrative console of the deployment manager.

About this task

If you are using clusters, the **Initial State** property of the Application Server subcomponent (**Servers** > **Application servers** > *server_name* > **Administration** > **Server Components** > **Application Server**) is not intended to be used to control the state of individual servers in the cluster at the time the cluster is started. It is intended only as a way to control the state of the Application Server subcomponent of a server. It is best to start and stop the individual members of a cluster using the Server options of the administrative console or command line commands (**startServer** and **stopServer**).

Related tasks

↪ First steps console

After installing WebSphere Process Server, use the First steps console to start product tooling, access product documentation, or direct elements such as servers and administrative consoles related to individual profiles. A generic version of the console, plus a version for each profile in your installation are available.

"Starting and stopping the administrative console" on page 15
To access the administrative console, you must start it and then log in. After you finish working in the console, save your work and log out.

Starting a stand-alone server

The server process must be started before you can use the administrative console or run applications on the server.

Before you begin

The server must not be running. Note that running the installation verification program starts the server.

About this task

There are several methods of starting a stand-alone server. On a host where you have created several stand-alone server profiles, take care that you are using the correct profile.

To start the server do one of the following:

*   Windows From the **Start** menu, click **IBM WebSphere** → **Process Server 6.1** → **Profiles** → *profile_name* → **Start the server**

    where *profile_name* is the name of the profile; for example, **ProcSrv01**.
*   In the First steps console, click **Start the server**.
*   Use the startServer command.

Progress is recorded in the *install_root*/profiles/*profile_name*/logs/ *server_name*/startServer.log. If the server started successfully, the log contains a message stating Server *server_name* open for e-business; process id is *nnnn*.

What to do next

You can now start the administrative console and begin to manage the stand-alone server and deploy applications to it. If you are new to WebSphere Process Server and you installed the samples, you could open the Samples gallery from the First steps console.

> **Related concepts**

> ⏩ Profiles

> A profile defines a unique runtime environment, with separate command files, configuration files, and log files. Profiles define three different types of environments: stand-alone server, deployment manager, and managed node.

> ⏩ Servers

> Servers provide the core functionality of WebSphere Process Server. Process servers extend, or augment, the ability of an application server to handle Service Component Architecture (SCA) modules. Other servers (deployment managers and node agents) are used for managing process servers.

> **Related tasks**

> "Stopping a stand-alone server" on page 44
> The server process must be stopped if want to make configuration changes to the server or any of the modules deployed to it.

> "Starting and stopping the administrative console" on page 15
> To access the administrative console, you must start it and then log in. After you finish working in the console, save your work and log out.

"Enabling a server for Java debugging" on page 53
If, after you have started it, you need to trace and debug a server, you can
enable it for Java debugging.

➦ First steps console

After installing WebSphere Process Server, use the First steps console to start
product tooling, access product documentation, or direct elements such as
servers and administrative consoles related to individual profiles. A generic
version of the console, plus a version for each profile in your installation are
available.

➦ Options on the First steps console

After installing WebSphere Process Server, use the First steps console to start
product tooling, access product documentation, or direct elements such as
servers and administrative consoles related to individual profiles. A generic
version of the console, plus a version for each profile in your installation are
available. Options on each console are displayed dynamically, depending on
features you install and the availability of certain elements on particular
operating systems. Options include verifying your installation, starting or
stopping the server or deployment manager, accessing the administrative
console, starting the Profile Management Tool, accessing the Samples gallery,
accessing the product documentation, or starting the migration wizard.

Related information

➦ startServer command

Stopping a stand-alone server

The server process must be stopped if want to make configuration changes to the
server or any of the modules deployed to it.

Before you begin

1. The server must be running.
2. Allow work to complete. Either wait for a suitable period or use performance
   monitoring infrastructure counters to determine whether all queued work has
   been completed.
3. Prevent new work from starting.

About this task

There are several methods of stopping a server.

To stop the server do one of the following:

* Windows From the **Start** menu, select **IBM WebSphere** ➙ **Process Server 6.1** ➙
  Profiles ➙ *profile_name* ➙ **Stop the server**

  where *profile_name* is the name of the profile; for example, **ProcSrv01**.
* In the First steps console, click **Stop the server**.
* Use the stopServer command.

Progress is recorded in the server log *install_root*/profiles/*profile_name*/logs/
server_name/stopServer.log. If the server stopped successfully, the server log
contains a message stating Server *server_name* stop completed..

What to do next

Wait for the server to stop before restarting it.

Related concepts

⤷ Profiles

A profile defines a unique runtime environment, with separate command files, configuration files, and log files. Profiles define three different types of environments: stand-alone server, deployment manager, and managed node.

⤷ Servers

Servers provide the core functionality of WebSphere Process Server. Process servers extend, or augment, the ability of an application server to handle Service Component Architecture (SCA) modules. Other servers (deployment managers and node agents) are used for managing process servers.

⤷ Profiles

A profile defines a unique runtime environment, with separate command files, configuration files, and log files. Profiles define three different types of environments: stand-alone server, deployment manager, and managed node.

Related tasks

"Starting a stand-alone server" on page 43
The server process must be started before you can use the administrative console or run applications on the server.

⤷ First steps console

After installing WebSphere Process Server, use the First steps console to start product tooling, access product documentation, or direct elements such as servers and administrative consoles related to individual profiles. A generic version of the console, plus a version for each profile in your installation are available.

Related information

⤷ stopServer command

Starting a deployment manager

The deployment manager must be started before you can use its administrative console to manage the cell.

Before you begin

The deployment manager must not be running. Note that running the installation verification program starts the deployment manager.

About this task

There are several methods of starting a deployment manager.

To start the deployment manager:

- Windows From the **Start** menu, select **IBM WebSphere** �˃ **Process Server 6.1** ˃ **Profiles** ˃ *profile_name* ˃ **Start the deployment manager**

  where *profile_name* is the name of the profile; for example, **Dmgr01**.
- In the First steps console, click **Start the deployment manager**.
- Use the startManager command.

Progress is recorded in the *install_root*/profiles/*profile_name*/logs/
server_name/startServer.log. If the server started successfully, the log contains a
message stating Server *server_name* open for e-business; process id is *nnnn*.

What to do next

You can now start the administrative console and begin to manage the cell. You
can federate nodes to the deployment manager, and then start creating servers and
clusters.

Related concepts

Profiles

A profile defines a unique runtime environment, with separate command files,
configuration files, and log files. Profiles define three different types of
environments: stand-alone server, deployment manager, and managed node.

Deployment manager

A deployment manager is a server that manages operations for a logical group,
or cell, of other servers. The deployment manager is the central location for
administering the servers and clusters.

Related tasks

"Stopping a deployment manager"
The deployment manager is a server process, which you may want to stop, for
example, to apply maintenance to the operating system. You can stop the
deployment manager at any time without affecting the operation of the servers
in its domain.

First steps console

After installing WebSphere Process Server, use the First steps console to start
product tooling, access product documentation, or direct elements such as
servers and administrative consoles related to individual profiles. A generic
version of the console, plus a version for each profile in your installation are
available.

Federating custom nodes to a deployment manager

Learn about how to use the **addNode** command to federate a custom node into
a deployment manager cell.

"Starting and stopping the administrative console" on page 15
To access the administrative console, you must start it and then log in. After
you finish working in the console, save your work and log out.

Related information

startManager command

Stopping a deployment manager

The deployment manager is a server process, which you may want to stop, for
example, to apply maintenance to the operating system. You can stop the
deployment manager at any time without affecting the operation of the servers in
its domain.

Before you begin

The deployment manager must be running.

About this task

There are several methods of stopping a deployment manager.

To stop the deployment manager do one of the following:

- **Windows** From the **Start** menu, click **IBM WebSphere** › **Process Server 6.1** › **Profiles** › *profile_name* › **Stop the deployment manager**

  where *profile_name* is the name of the profile; for example, **Dmgr01**.
- In the First steps console, click **Stop the deployment manager**.
- From the administrative console, click **System administration** › **Deployment manager** › **Stop** › **OK**. The administration console closes before the server stops running.
- Use the stopManager command.

Progress is recorded in the server log *install_root*/profiles/*profile_name*/logs/*server_name*/stopServer.log. If the server stopped successfully, the server log contains a message stating Server *server_name* stop completed..

What to do next

Wait for the deployment manager to stop before restarting it.

> **Related concepts**
>
> ⇨ Profiles
>
> A profile defines a unique runtime environment, with separate command files, configuration files, and log files. Profiles define three different types of environments: stand-alone server, deployment manager, and managed node.
>
> ⇨ Deployment manager
>
> A deployment manager is a server that manages operations for a logical group, or cell, of other servers. The deployment manager is the central location for administering the servers and clusters.
>
> **Related tasks**
>
> "Starting a deployment manager" on page 45
> The deployment manager must be started before you can use its administrative console to manage the cell.
>
> ⇨ First steps console
>
> After installing WebSphere Process Server, use the First steps console to start product tooling, access product documentation, or direct elements such as servers and administrative consoles related to individual profiles. A generic version of the console, plus a version for each profile in your installation are available.
>
> **Related information**
>
> ⇨ stopManager command

Starting a node agent

The node agent of a managed node is a server process that must be started before you can start servers on the node. The node agent must be started for the deployment manager to communicate with it.

Before you begin

The node agent must not be running. Instructions are given here for using the administrative console to determine if it is running

About this task

You must start the node agent from the command line of the host on which the node is configured, in the *install_root*/bin directory.

Procedure
1. To determine the correct node agent for a server from the administrative console of the deployment manager, click **Servers** → **Application servers**.
2. To find out if the node agent is running, click **System administration** → **Node agents**.
3. To start the node agent, run the startNode command. To find out the options you can use with this command, use the **-help** option. With no options, the node is started. Wait for the message Server nodeagent open for e-business; process id is nnnn
- To start the node agent in the default profile, type startNode
- To list the options, type startNode -help
- To start the node agent in the Custom03 profile, type startNode -profileName Custom03
- To start the node agent in the Custom03 profile and write trace information to the log file called *install_root*/profiles/Custom03/logs/startServer.log, type startNode -logfile -profileName Custom03

What to do next

You can now manage this node from the deployment manager, including starting the servers on the node.

    Related concepts

    ⬐ Managed node

    A managed node is a node that has been federated into a deployment manager cell. In a managed node, you can configure and run managed servers.

    Related tasks

    "Restarting a node agent" on page 49
    The node agent of a managed node is a server process that you may need to restart.

    "Stopping a node agent"
    The node agent of a managed node is a server process that you may need to stop.

    Related information

    ⬐ startNode command

Stopping a node agent

The node agent of a managed node is a server process that you may need to stop.

Before you begin
1. The node agent must be running.
2. All the servers must be stopped.

Procedure

1. To stop a node agent from the administrative console of the deployment manager, select **System administration** → **Node agents**
2. Select the node agent.
3. Click **Stop**.

Related concepts

⤷ Managed node

A managed node is a node that has been federated into a deployment manager cell. In a managed node, you can configure and run managed servers.

Related tasks

"Starting a node agent" on page 47
The node agent of a managed node is a server process that must be started before you can start servers on the node. The node agent must be started for the deployment manager to communicate with it.

"Starting and stopping the administrative console" on page 15
To access the administrative console, you must start it and then log in. After you finish working in the console, save your work and log out.

"Restarting a node agent"
The node agent of a managed node is a server process that you may need to restart.

Related information

⤷ stopNode command

Restarting a node agent

The node agent of a managed node is a server process that you may need to restart.

Before you begin

The node agent must be running.

Procedure

1. To restart a node agent from the administrative console of the deployment manager, click **System administration** → **Node agents**.
2. Select the node agent.
3. Click **Restart**.

Related concepts

⤷ Managed node

A managed node is a node that has been federated into a deployment manager cell. In a managed node, you can configure and run managed servers.

Related tasks

"Starting a node agent" on page 47
The node agent of a managed node is a server process that must be started before you can start servers on the node. The node agent must be started for the deployment manager to communicate with it.

"Stopping a node agent" on page 48
The node agent of a managed node is a server process that you may need to stop.

"Starting and stopping the administrative console" on page 15
To access the administrative console, you must start it and then log in. After you finish working in the console, save your work and log out.

Related information

⬀ startNode command

⬀ stopNode command

Starting and stopping deployment environments

You can start or stop deployment environments based on IBM-supplied patterns directly from the administrative console.

Before you begin

Required security role for this task: When security and role-base authorization are enabled, you must log in as an administrator or operator to perform this task.

A deployment environment must exist.

About this task

Use these steps whenever you want to start or stop a deployment environment.

Restriction: You cannot use this procedure for custom deployment environment; you must start and stop the clusters individually.

Procedure

1. Log in to the administrative console of the deployment manager that controls the deployment environments you are starting.
2. Navigate to the deployment environment administrative console page. Click **Servers** ⊁ **Deployment Environments**.
3. Click the check boxes next to the names of the deployment environments you are starting or stopping.
4. Start or stop the selected deployment environments.

Action	Result
Click Start.	The deployment manager starts the clusters that make up the deployment environments.
Click Stop.	The deployment manager stops the clusters that make up the deployment environments.

Note: This process can take several minutes depending on the size of your deployment environment.

Results

The display refreshes to indicate the status of the deployment environments. If successful, a deployment environment will have an indication of the deployment environment state, as described in Deployment environment status.

Related concepts

⬀ Deployment environments

A deployment environment is a collection of configured clusters, servers, and middleware that collaborates to provide an environment to host Service Component Architecture (SCA) interactions. For example, a deployment environment might include a host for message destinations, a processor of business events, and administrative programs.

Related tasks

"Starting and stopping the administrative console" on page 15
To access the administrative console, you must start it and then log in. After you finish working in the console, save your work and log out.

Related information

⇨ Using the Administration Thin Client

Starting a cluster

You can start all the servers in a cluster (cluster members) in one action. Ripplestart first stops and then starts each server in turn. When you start a cluster you automatically enable workload management.

Before you begin

1. The node agents must be running.
2. Verify that all resources required by applications deployed to the cluster are available and start all prerequisite subsystems.

Procedure

1. To start the cluster from the administrative console of the deployment manager, click **Servers** → **Clusters**.
2. Select the cluster.
3. If the servers are not running, click **Start**.
4. If the servers are running, click **Ripplestart**. This first stops and then starts each server in turn.

What to do next

You can now start applications on the cluster. If you need to trace and debug processes running on one of the servers, you can enable Java debugging.

Related concepts

⇨ Clusters

Clusters give your applications more capacity and higher availability than a single server.

Related tasks

"Starting and stopping the administrative console" on page 15
To access the administrative console, you must start it and then log in. After you finish working in the console, save your work and log out.
"Stopping a cluster"
You can stop all the servers in a cluster (cluster members) in one action.

Stopping a cluster

You can stop all the servers in a cluster (cluster members) in one action.

Before you begin

1. Allow work to complete. Either wait for a suitable period or use performance monitoring infrastructure counters to determine whether all queued work has been completed.
2. Prevent new work from starting.

Procedure

1. To stop the cluster from the administrative console of the deployment manager, click **Servers** → **Clusters**.
2. Select the cluster.
3. If you want to restart the servers, click **Ripplestart**. This first stops and then starts each server in turn.
4. If the servers are not running, click **Stop**.

What to do next

Unless you used **Ripplestart**, wait for the servers to stop before restarting them.

Related concepts

Clusters

Clusters give your applications more capacity and higher availability than a single server.

Related tasks

"Starting a cluster" on page 51
You can start all the servers in a cluster (cluster members) in one action. Ripplestart first stops and then starts each server in turn. When you start a cluster you automatically enable workload management.

"Starting and stopping the administrative console" on page 15
To access the administrative console, you must start it and then log in. After you finish working in the console, save your work and log out.

Starting a server

The server process must be started before you can run applications on a managed server. You can start a single server, even if the server is a member of a cluster.

Before you begin

1. The node agent must be running.
2. Verify that all resources required by applications deployed to the server are available and start all prerequisite subsystems.
3. The server must not be running. If the server has failed and you want the recovery functions to finish before new work is started, restart the server in recovery mode using the startServer command with the **-recovery** option.

Procedure

1. To start the server from the administrative console of the deployment manager, click **Servers** → **Application servers**.
2. Select the server.
3. Click **Start**. Progress is recorded in the *install_root*/profiles/*profile_name*/logs/*server_name*/startServer.log. If the server started successfully, the log contains a message stating Server *server_name* open for e-business; process id is *nnnn*.

Results

Other processes might not immediately discover the newly-started server.

What to do next

You can now start applications on the server. If you need to trace and debug processes running on the server, you can enable Java debugging.

Related concepts

➡ Managed servers

A managed server is a server that is configured in a managed node. It provides a resource within the deployment environment that runs your applications.

Related tasks

"Starting and stopping the administrative console" on page 15
To access the administrative console, you must start it and then log in. After you finish working in the console, save your work and log out.

"Stopping a server" on page 54
The server process must be stopped if want to make configuration changes to the server or any of the modules deployed to it. You can stop a single server, even if the server is a member of a cluster.

Related information

➡ startServer command

Starting a server from the i5/OS command line
There are two additional methods of starting a server.

- You can use the startServer Qshell command.
- You can use the Submit Job (SBMJOB) CL command. You can run the following CL command from an i5/OS command line.

```
SBMJOB CMD(CALL PGM(QWBI61/QWASSTRSVR) PARM('-profilePath'
    'profile_root' '-server' 'server_name')) JOB(server_name)
    JOBD(QWAS61/QWASJOBD) JOBQ(QWAS61/QWASJOBQ) USER(QEJBSVR)
LANGID(*USRPRF)
    CNTRYID(*USRPRF) CCSID(*USRPRF) OUTQ(QWAS61/QWASOUTQ) ALWMLTTHD(*YES)
```

server_name is the name of the server.

profile_root is the profile directory of the server.

Enabling a server for Java debugging
If, after you have started it, you need to trace and debug a server, you can enable it for Java debugging.

Before you begin

The server must be running.

Procedure

1. To enable Java debugging from the administrative console of the deployment manager, click **Servers** ➤ **Application servers**.
2. Select the server.
3. Click **Server infrastructure** ➤ **Java and Process Management** ➤ **Process Definition** ➤ **Java Virtual Machine**.
4. To enable the standard Java debugger, on the Java virtual machine page, click **Debug mode**
5. If required, set **Debug Mode** arguments.

6. Click **OK**.
7. Save the changes.

What to do next

You must now stop the server and then, if it is a stand-alone server, restart it.

> **Related tasks**
>
> "Starting a stand-alone server" on page 43
> The server process must be started before you can use the administrative
> console or run applications on the server.

Stopping a server

The server process must be stopped if want to make configuration changes to the
server or any of the modules deployed to it. You can stop a single server, even if
the server is a member of a cluster.

Before you begin

1. The server must be running.
2. Allow work to complete. Either wait for a suitable period or use performance
   monitoring infrastructure counters to determine whether all queued work has
   been completed.
3. Prevent new work from starting:
   - If you are using the IBM HTTP Server, change the plugin_cfg.xml file to
     remove the server for HTTP traffic. If you are using a different HTTP server,
     use its instructions to remove the server.
   - For IIOP traffic, set the runtime weight to zero for the server.
   - Quiesce the service integration bus.

Procedure

1. To stop the server from the administrative console of the deployment manager,
   click **Servers** → **Application servers**.
2. Select the server.
3. Click **Stop**. Progress is recorded in the *install_root*/profiles/*profile_name*/
   logs/*server_name*/stopServer.log. If the server started successfully, the log
   contains a message stating Server *server_name* stop completed..

What to do next

Wait for the server to stop before restarting it.

> **Related concepts**
>
> ↪ Managed servers
>
> A managed server is a server that is configured in a managed node. It provides
> a resource within the deployment environment that runs your applications.
>
> **Related tasks**
>
> "Starting a server" on page 52
> The server process must be started before you can run applications on a
> managed server. You can start a single server, even if the server is a member of
> a cluster.

"Starting and stopping the administrative console" on page 15
To access the administrative console, you must start it and then log in. After you finish working in the console, save your work and log out.

Related information

 stopServer command

Chapter 4. Administering deployment environments

The administrative console on the deployment manager provides you the ability to administer the deployment environments defined to it. You can also create, delete, import, and export deployment environments from the administrative console.

Before you begin

A deployment manager must be started and you must log in to the administrative console.

Required security role for this task: When security and role-base authorization are enabled, you must log in as an administrator or operator to perform this task.

About this task

The **Deployment Environments** administrative console page is the starting point for all tasks related to the management and definition of deployment environments defined to a particular deployment manager.

Procedure

1. Navigate to the administrative console page. Click **Servers > Deployment Environments**
2. To display the components of a deployment environment, click the *deployment environment name*.
3. For existing environments, click the check box next to the deployment environments to manage and click the button for the function to perform.

Function	Task
Start or Stop	Starting and stopping deployment environments
Remove	Removing resources from a deployment environment. This option does not delete the resources created.
Export . . .	Exporting deployment environments

4. To add new deployment environments to the deployment manager use either **New . . .** or **Import . . . .**

What to do next

Manage deployment environment entities.

Overview of custom deployment environment layout configuration

This overview describes two major configuration section for custom deployment environment.

"Select Clusters and Single Servers for use with this deployment environment" on page 58 defines the clusters and servers that make up your deployment environment. Unlike the patterned deployment environments where clusters are

created for each function you add the clusters and servers that you need to fulfill the custom deployment environment functions.

"Specify the Deployment Environment Configuration" on page 59 describes the functions you configure the clusters and servers to provide. These functions can be messaging, Common Event Infrastructure. or application support.

Before you complete the deployment environment configuration in the system by generating the environment, you can return to your configuration and make changes. Once you generate the deployment environment configuration in the system, you can look at the current configuration. You can also add more servers and clusters, configure more functions, or you can remove server and clusters from management by this deployment environment. You cannot undo a function configuration you have already generated and you cannot remove a server or cluster from the deployment environment definition still in use by the servers and clusters in your deployment environment.

Administrative console behaviors

A custom deployment environment layout has restrictions such as:
- Once a user completes a configuration by generating the deployment environment, the associated controls become check and disabled. This means you cannot undo the configuration.
- If a control is not checked and disabled for a component, you need to configure the associated messaging engine or Common Event Infrastructure (CEI) server first. Configure functions by group: messaging, then CEI, and then application support as described later in this topic.
- The configurations that exist on a system take precedence over the topology layout configuration. Thus, exporting a custom topology reflects the actual configuration of the servers involved in the topology.

The Topology Layout page has four sections that must be configured for a custom topology:
- Select Cluster and Single Servers
- Messaging
- Common Event Infrastructure
- Components

There are restrictions that you must keep in mind that could prevent you from completing your custom topology layout configuration. Each of the following sections lists specific requirements or restrictions.

Select Clusters and Single Servers for use with this deployment environment

Use this section to manage the clusters and servers within the deployment environment and define which functions they provide.

The drop-down list contains available clusters and servers that you configure as part of this deployment environment. Assign clusters and servers to collaborative units in the function configuration. Each collaborative unit represents a group of clusters and servers that provides, as a whole, a function in the deployment environment. You can remove clusters or servers that this deployment environment should no longer manage. However, you can only remove clusters or servers that are no longer needed by other clusters or servers in the configuration.

Specify the Deployment Environment Configuration

Messaging

> **Restriction:** Partitioned messaging engines are not supported.

> Use the Messaging tab to configure the messaging destination location for selected targets. There can be multiple tables in the Messaging section. Each table represents a single collaborative unit. You must select only one target (Cluster/Server) for the option of local configuration for each unit and all other targets in this unit assumes the remote destination. Messages sent to targets with remote destination configuration will route the messages to the local target for their unit.

> The messaging configuration applies to the SCA system buses, the CEI and Business Process Choreographer system bus.

> To prevent conflicts with the local destinations within your topology configuration, the following rules applies:

> - The SCA system bus messaging engine configuration determines the local and remote destination locations. The SCA application, the CEI and Business Process Choreographer bus configurations follow the SCA system bus configuration.

> - If you locate the messaging engines for other buses on different targets in a unit, then the other targets within that unit assume the remote destination role. No visual feedback is given on the table if CEI and Business Process Choreographer have different configurations. An information message will show that the messaging engine for a given bus is not located on the same target where the SCA messaging engine is located.

> - You cannot add a target that already has a remote or local destination configured that conflicts with the current bus settings of a given unit. If you attempt this, the system generates an error message.

Common Event Infrastructure

> Like Messaging, Common Event Infrastructure (CEI) can have multiple tables where each table is a unit. Within each table is one CEI Cluster/Server that acts as the server by selecting the **Server** radio button. All targets not configured as a server assume the destination role. The corresponding targets have their Event Infrastructure emitter factory Java Naming and Directory Interface (JNDI) name configured so that common base events emitted on this target are sent to the server in their respective collaborative unit.

Application support

> The Application Support tab lists all the components you can configure for a given deployment target. You configure component functions in a related collaborative unit. For example, you configure a Business Process Choreographer Event Collector in a unit to collect the common base events emitted by the Business Process Choreographer Container configured in the same unit. Each component configuration has restrictions and dependencies on other component configurations. Dependencies are represented by unchecked and disabled controls. To enable them, you must configure dependent controls first.

> **Important:** Dependent controls are either on the Messaging or the CEI tab. Table 2 on page 60 describes the relationships between the components.

Table 2. Components, their interrelationships and considerations

Component	Purpose	Related Component	Considerations
Service Component Architecture (SCA)	Configures the deployment target for SCA application support. The SCA system and application bus members are configured locally if the corresponding messaging configuration is local, otherwise they are configured remotely with remote destination location as specified in the corresponding messaging unit.	Messaging	SCA configuration is not available if you have not configured the deployment target for messaging.
Business Process Choreographer Container	Configures the deployment target for both business flow and human task support. The configuration follows the SCA configuration for setting up the Business Process Choreographer system bus.	Messaging Service Component Architecture Business Process Choreographer explorer	Business Process Choreographer configuration is not available if the deployment target has not been configured for messaging or if it has not been configured for Service Component Architecture support. One collaborative unit supports one Business Process Choreographer configuration. Add as many units as you need Business Process Choreographer configurations on the Application Support tab. To manage a container, consider configuring the Business Process Choreographer explorer.

Table 2. Components, their interrelationships and considerations (continued)

Component	Purpose	Related Component	Considerations
Business Process Choreographer Explorer	Configures the Business Process Choreographer explorer on the selected deployment target. The Business Process Choreographer explorer is a web application that manages the Business Process Choreographer container configured in the same collaborative unit.	Business Process Choreographer Container	The Business Process Choreographer explorer configuration is available once you have selected a Business Process Choreographer container configuration in the same collaborative unit. You must configure the deployment target for web application support You can configure as many Business Process Choreographer explorers on a given deployment target as you want. Add the deployment target of choice to the collaborative units with a configured container and check the explorer's configuration control.

Table 2. Components, their interrelationships and considerations (continued)

Component	Purpose	Related Component	Considerations
Business Process Choreographer Event Collector	Configures the Business Process Choreographer Event Collector on the selected deployment target. The Business Process Choreographer Event Collector gathers common base events that are emitted from the Business Process Choreographer container configured in the same collaborative unit. Statistical information about the observed container is recorded in a database.	Business Process Choreographer container Common Event Infrastructure Business Process Choreographer observer	Configure first the Common Event Infrastructure server on the same deployment target that you plan to use for the Business Process Choreographer Event Collector. Additionally the Business Process Choreographer Event Collector is only available once you have configured the Business Process Choreographer container in the same collaborative unit. If you are not sure whether you need to observe a given Business Process Choreographer container, you can always decide to configure this function later. To examine the observed data, consider configuring the Business Process Choreographer observer.

Component	Purpose	Related Component	Considerations
Business Process Choreographer Observer	Configures the Business Process Choreographer observer on the selected deployment target. The Business Process Choreographer observer is a web application that allows you to examine the statistical information gathered by the Business Process Choreographer event collector for the Business Process Choreographer container configured in the same collaborative unit.	Business Process Choreographer Container Business Process Choreographer Event Collector	The Business Process Choreographer observer configuration is available once you select a Business Process Choreographer event collector configuration in the same collaborative unit. You must configure the deployment target for web application support. You can configure as many Business Process Choreographer observers on a given deployment target as you want. Add the deployment target of choice to the collaborative units with a configured event collector and check the observer's configuration control.
Business Rules Manager	Configures the Business Rules Manager on the selected deployment target. The Business Rules Manager allows you to configure business rules that determine business process behavior.	Service Component Architecture	The Business Rules Manager configuration control is available once you have configured SCA support on the same deployment target. You can configure only one Business Rules Manager for a given deployment environment. You may only need to configure one Business Rules Manager in your system as one Business Rules Manager takes care of the business rules configuration of the complete cell.

Creating deployment environments

Once you have decided on a pattern that meets your needs, use the Deployment Environment Configuration Wizard to create the deployment environment based on the pattern.

Before you begin

You must be at the administrative console of a deployment manager. Navigate to the page by clicking **Servers > Deployment Environments**.

Required security role for this task: When security and role-base authorization are enabled, you must log in as an administrator or operator to perform this task.

About this task

This task creates a deployment environment to a specific pattern. The Deployment Environment Configuration Wizard directs the steps in an orderly fashion to ease the process.

Note: If you make an error while working in the wizard, you can go back by clicking **Back**.

Procedure

1. Launch the Deployment Environment Configuration Wizard by clicking **New** on the Deployment Environments page.

   The system displays the first page of the Deployment Environment Configuration Wizard with **Create a new deployment environment** already selected.

2. Give the deployment environment a unique name, select the runtime capability from the list, then click **Next**.

   Runtime Capability can be:

   - **WESB** which provides a deployment environment that supports mediations.
   - **WPS** which provides a deployment environment that supports mediations, business processes, human tasks and business rules.

   The default for **Runtime Capability** will match the capability of the deployment manager. The wizard displays the Deployment Environment Patterns page.

3. Select the deployment environment pattern for this deployment environment then click **Next**.

   This pattern should match the pattern you chose for this environment during the planning phase. If you are unfamiliar with the patterns, see Deployment environment patterns.

   Note: If you select the Custom pattern, you define your custom deployment environment configuration. For more information, see Deployment environment patterns.
   The wizard displays the Select Nodes page.

4. Select the nodes to include in this deployment environment then click **Next**.

   To include a node, select the checkbox next to the node name. Use **Node Mapping** to map the selected node to another node name. The wizard displays the Clusters page.

5. Assign the number of cluster members on each node for each function of the deployment environment.

   The default is to assign one cluster member on each node for each function. You change the number by typing over the number in each column. If you are unfamiliar with functions, see Deployment environment functions. A 0 for a node means that the node does not contribute to that particular function. The wizard displays the Database page.

6. Configure the databases for the deployment environment then click **Next**.

   On this page, define the database information for the components included in this deployment environment. Where possible, the wizard supplies default information for the parameters but change those values to match the values that you defined when you planned the environment.

7. Optional: Define the Business Process Choreographer configuration and then click **Next**.

   On this page you specify the values for:
   - Context roots
   - Security roles
   - Authentication aliases
   - Human task manager mail session, if desired

   Note: This page is only displayed if **WPS** is the **Runtime Capability**.

   The wizard displays the Business Rules Manager page.

8. Optional: Specify the context root for the Business Rules Manager and click **Next**.

   Note: This page is only displayed if **WPS** is the **Runtime Capability**.

   The wizard displays the Summary page.

9. Verify that the information on the Summary page is correct and click **Finish and Generate Environment** to save and complete the configuration of the deployment environment. To exit without completing the configuration click, **Finish**.

Results

When the configuration completes, you can examine the configuration files to view the changes made.

What to do next

Either save the changes to the master configuration or discard the changes.

   Related concepts

    Deployment environment patterns

   A deployment environment pattern specifies the constraints and requirements of the components and resources involved in a deployment environment. The patterns are designed to meet the needs of most business requirements and are intended to help you create a deployment environment in the most straightforward way.

    Deployment environment functions within deployment patterns

   To design a robust deployment environment, you need to understand the functionality each cluster can provide in a particular IBM-supplied deployment

environment pattern or a custom deployment environment. This knowledge can help you make the correct decisions as to which deployment environment pattern best meets your needs.

⮕ Database specifications

WebSphere Process Server uses a number of database tables to hold, store and track information. Some of the components that comprise WebSphere Process Server use their own database tables. You can create these database tables during profile creation or you can choose to create them separately using scripts.

"Business rules manager" on page 11
The business rules manager is a Web-based tool that assists the business analyst in browsing and modifying business rule values. The tool is an option of WebSphere Process Server that you can select to install at profile creation time or after the initially installing the server.

Related information

⮕ Configuring Business Process Choreographer

Configuring deployment environment functions

From a single page, you can add nodes to topologies and assign functions to clusters within a deployment environment based on an IBM-supplied pattern. You can also delete nodes from the deployment environment.

Before you begin

- Deployment environments exist on this deployment manager.
- You must be at the administrative console of the deployment manager.

Navigate to this page by clicking **Servers** › **Deployment Environments**. Select your deployment environment and then click **Additional Properties** › **Deployment Topology**.

Required security role for this task: When security and role-base authorization are enabled, you must log in as an administrator or operator to perform this task.

About this task

Important: You can only configure functions for deployment environments based on IBM-supplied patterns. You cannot access the Deployment Topology page from a custom deployment environment.

As your deployment environment requirements change, you will add and remove nodes to existing deployment environments or change the allocation of resources to the various functions in the deployment environment.

- To add a federated node, select a node from the list and click **Add Node**.

  This action places the node at the end of the node list in the display.

  To add a node into an empty row of the display:

  1. Select the node from the list
  2. click the check box on the empty row
  3. click **Add node**
- To add an empty or nonexistent node to the configuration, type the name in the **Node name:** field and click **Add Node**.

- To delete a node, click the check box next to the node to delete and click **Delete**.
- To assign functions to nodes, type a number in the field in each column. The number represents the number of cluster members the system configures to support the given function.
- Save the configuration. Click **OK** or **Apply**. If there are resources to configure, the system initiates the configuration process. You will see a configuration progress dialog and the system will prompt you to save the changes to the master configuration.

  Note: The system does not complete the configuration until you click **Generate Environment**.

Results

The page updates with your changes and the status of the nodes and roles for this deployment environment.

> **Related reference**
>
> ⮕ Deployment environment status information
>
> Describes the status of specific entities within a deployment environment.
>
> ⮕ Deployment environment function status
>
> Describes the state of the minimum required entities and the redundant entities of a configured deployment environment.
>
> Deployment environment status
>
> Describes the indicators that show the state of a deployment environment. The warning icon in the topology status indicates the presence of warnings for that deployment environment.

Configuring host aliases

Configure the IBM® HTTP server or a server of your choice to allow communication between managed nodes and the deployment manager.

Before you begin

You need to create and configure deployment manager and associated nodes.

About this task

So that the managed nodes and the deployment manager can communicate with each other, you need to make sure that the host name alias for each node in the deployment target cluster is visible to the deployment manager. The host name alias consists of the DNS host name and port number. You will use this alias as part of a URL address to access applications once these applications are executing on the deployment target.

Note: This description uses two servers named AppCluster_member1 and AppCluster_member2. Substitute your server names in the instructions.

Procedure

1. From the administrative console, expand **Servers**, then click **Application servers**.
2. Click the name *AppCluster_member1*.

3. Under the Communications heading, expand **Ports**.

4. Note the port value listed for *WC_defaulthost*. You will need to use it later.

5. Repeat steps 1 on page 67 through 4, except in step 2 on page 67, select *AppCluster_member2*. Repeat this for each additional application cluster member.

   When done, you will have a list of the cluster members and the port numbers for their default host.

6. From the administrative console, click **Environment** → **Virtual hosts**.

7. Click the name **default_host**.

8. Under **Additional Properties**, click **Host aliases**.

9. If an entry for the correct combination of host name and port value for cluster members is not in the displayed list, add the missing entries to the list.

10. If you added any new entries to the list, click **Save** and then **Synchronize**.

What to do next

Verify your installation by installing a test application.

Configuring authentication aliases for a deployment environment

From a single page, you can review or edit all your authentication aliases.

Before you begin
- Deployment environments exist on this deployment manager.
- You must be at the administrative console of the deployment manager.

Navigate to this page by clicking **Servers** → **Deployment Environments**. Select your deployment environment and then click **Additional Properties** → **Authentication Aliases**.

Required security role for this task: When security and role-base authorization are enabled, you must log in as an administrator or operator to perform this task.

About this task

From this consolidated list of authentication aliases, you can:
- Review all the aliases for a given deployment environment
- Access the authentication configuration page through the *Alias* link

Reset resets the selected rows to the currently configured values. Click the *Alias* link to access the authentication configuration page where you can make your changes.

Procedure
1. Select the row you want to change.
2. Do one of the following:

Option	Description
To edit the row	Click *Alias* link.
To reset the row	Click **Reset**.

Editing a row takes you to the authentication configuration page where you make your changes.

3. Click **OK** or **Apply** to save any changes.

Configuring custom deployment environments

Use the Custom Deployment Topology Detail page to configure your custom deployment environment.

Before you begin

- Deployment environments exist on this deployment manager.
- You must be at the administrative console of the deployment manager.

Navigate to this page by clicking **Servers** → **Deployment Environments**. Select your deployment environment and then click **Additional Properties** → **Custom Deployment Topology Detail**.

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an administrator or a configurator to perform this task.

Restrictions:

- The configurations that exist on a system take precedence over the deployment environment configuration. Thus, exporting a custom deployment environment reflects the actual configuration of the servers involved in the deployment environment.
- You need to configure the messaging units before configuring the component units. If the checkbox is unavailable, then you have not yet configured messaging support.

About this task

For a custom deployment environment you can decide how to configure each function according to your needs. Configure each function to either clusters or single servers. There are three major areas in configuring a custom deployment environment topology:

- Messaging, which supports component internal communication.
- Common Event Infrastructure, which unifies event and monitoring functionality.
- Application Support, which supports business integration service components such as business processes and human tasks.

For more information see "Overview of custom deployment environment layout configuration."

Procedure

1. In **Select Clusters and Servers for use with this Deployment Environment**, select a cluster or server from the list.
2. Click **Add**. The cluster or single server will be added to the table below.
3. Repeat steps 1 and 2 until you have all the clusters and servers you need for this deployment environment.
4. Click on the **Messaging** tab.

a. Decide how many independent messaging units you need for your deployment environment and add that number by clicking **Add New Unit**.

The system names each unit Messaging Unit x, where x is the number of the unit.

b. Assign clusters and servers from the table created in step 2 on page 69 to each unit.

Select the cluster or server to add to the unit and then choose the unit from the list named **Add selected to unit**.

c. Decide which deployment target in each unit is to host local messaging support.

d. Configure the local messaging host by clicking **Local Bus Member** on the row that defines that deployment target in the unit.

All other clusters or servers are automatically configured for remote messaging destinations.

5. Click on the **Common Events Infrastructure** tab.

a. Decide how many independent Common Events Infrastructure units you need for your deployment environment and add that number by clicking **Add New Unit**.

The system names each unit Common Event Infrastructure Unit x, where x is the number of the unit.

b. Assign clusters and servers from the table created in step 2 on page 69 to each unit.

Select the cluster or server to add to the unit and then choose the unit from the list named **Add selected to unit**.

c. Decide which deployment target in each unit is to host the Common Event Infrastructure server.

d. Configure the Common Event Infrastructure server host by clicking **Server** on the row that defines that deployment target in the unit.

All other clusters or servers are automatically configured for remote Common Event Infrastructure destinations.

6. Click on the **Application Support** tab. This tab shows all the components that can be configured for a given deployment target.

Restriction: You must complete the messaging units for each component before you can configure the component in this section. For example, if the check box is unavailable for Service Component Architecture, then the associated messaging units have not been configured. See "Overview of deployment environment configuration" for additional restrictions.

a. Decide how many independent Application Support units you need for your deployment environment and add that number by clicking **Add New Unit**.

The number of units you need depends on how many Business Process Choreographer containers you need. If you do not need Business Process Choreographer containers a single unit will be sufficient for Service Component Architecture applications.

The system names each unit Application Support Unit x, where x is the number of the unit.

b. Assign clusters and servers from the table created in step 2 on page 69 to each unit.

Select the cluster or server to add to the unit and then choose the unit from the list named **Add selected to unit**.

c. In a unit, select what cluster or server belongs to each component for your deployment environment.

d. Repeat steps 6b on page 70 and 6c until you configure all the component in each unit you need for your deployment environment.

What to do next

After completing or making edits to an existing deployment environment, the Custom Deployment Environment Configuration Wizard opens. You can review the information and make any necessary changes.

> **Related concepts**
>
> "Overview of custom deployment environment layout configuration" on page 57
> This overview describes two major configuration section for custom deployment environment.

Configuring deferred configurations for a deployment environment

If you must defer the creation of your databases and tables, use the Deferred Configuration page. This page provides instructions on how to locate and run scripts for database and table creation.

Before you begin

- Deployment environments exist on this deployment manager.
- You must be at the administrative console of the deployment manager.

Navigate to this page by clicking **Servers** → **Deployment Environments**. Select your deployment environment and then click **Additional Properties** → **Deferred Configuration**.

Required security role for this task: When security and role-base authorization are enabled, you must log in as an administrator or operator to perform this task.

About this task

The Deferred Configuration page shows the configuration steps needed to correctly configure your topology's databases. In most cases this page shows:

- Script location
- Instructions on how to run the scripts

Procedure

1. Perform the instructions provided in the Deferred Configuration page.
2. When you have finished, click **Configuration Done**.

What to do next

A text box shows by whom and when the deferred configuration was performed last. These instructions will remain for future reference.

Configuring deployment environments using the command line

You can configure deployment environments using the wsadmin interface. This capability allows you to configure multiple deployment environments unattended on a deployment manager using a script.

Before you begin

You must be using the deployment manager on which you are configuring deployment environments.

Required security role for this task: When security and role-base authorization are enabled, you must log in as an administrator or operator to perform this task.

About this task

After you have imported or created deployment environments on a deployment manager, you can configure the deployment environments using the generateDeploymentEnv command.

Procedure

1. Enter the wsadmin environment.
2. Enter the generateDeploymentEnv command for each topology you are configuring.

The following command configures topologies eastEnvironment and westEnvironment on host myDmgr.

```
wsadmin -connType SOAP -host myDmgr -port 8879
> $AdminTask generateDeploymentEnv -topologyName eastTopology
> $AdminTask generateDeploymentEnv -topologyName westTopology
> $AdminConfig save
```

Note: When you are running with global security enabled, you will be prompted for a user ID and password after the system processes the wsadmin command.

What to do next

Save the configured deployment environments. From the command line, you can enter $AdminConfig save.

> **Related reference**

⯈ generateDeploymentEnv command

Use generateDeploymentEnv to configure deployment environments on a deployment manager.

> **Related information**

⯈ Managing node agents

Viewing the deployment topology

Use the Deployment Topology page to review the configuration information for your IBM-supplied patterns.

Before you begin

Navigate to this page by clicking **Servers** → **Deployment Environments**. Select your deployment environment and then click **Additional Properties** → **Deployment Topology**.

About this task

Use this page to add nodes to your deployment environment, if needed. You can also change the number of cluster members participating in a particular function for each node.

- To add an unnamed node, click **Unnamed node** then **Add**. This action adds an empty node.
- To add a new node, click **New node**, enter the node name, and then click **Add**.
- To add an existing node, click **Existing node**, select the node name from the list, and then click **Add**.
- To replace a node in the deployment environment, click **Select** next to the node to replace, click either **Unnamed node**, **New node**, or **Existing node** depending on how you are replacing the selected node, and then click **Replace Selected**.
- To change the number of cluster members involved in each function, type the number in the entry field.

  The functions may include:

  - Application Deployment Target
  - Messaging Infrastructure
  - Supporting Infrastructure

  Remember: You must have at least one cluster member assigned for each function.
- Click **Apply** to keep the updates and remain on the Deployment Topology page. Click **OK** to keep the updates and return to the previous page.

What to do next

Either save the changes or discard them.

> **Related concepts**
>
> "Overview of custom deployment environment layout configuration" on page 57
> This overview describes two major configuration section for custom deployment environment.

Cluster, single server and node status

Describes the status of specific entities within a deployment environment.

Table 3. Deployment environment status

Icon	Status	Description
	Unknown	The system cannot determine the status of the entity. This could indicate an incomplete configuration.
	Unavailable	The entity is configured, but unavailable.
	Stopped	The entity is stopped.
	Partially stopped	The entity is partially stopped. Less than all of the entities involved have stopped.

Table 3. Deployment environment status (continued)

Icon	Status	Description
⟹	Running	The entity is fully functional and running.
◁⟹	Partially started	The entity is partially started. Less than all of the entities involved have started.

Deployment Environment function status

Describes the state of the minimum required entities and the redundant entities of a configured deployment environment.

Explanation of entities state

The function status is typically used for clusters that perform a given function. A cluster is an example of a redundant entity state where its cluster members make up the redundant parts. A deployment environment status is an example of a minimum entity state where all functions need to be available for the deployment environment to be available.

Table 4. Aggregated state of entities

Icon	Status	Minimum entities state	Redundant entities state
⊘	Unknown	At least one of the minimum entities state is unknown, making the entire state unknown.	The configuration did not complete for the deployment environment.
🚫	Unavailable	At least one of the minimum entities is unavailable.	All of the entities in the deployment environment are unavailable.
✖	Stopped	All entities are stopped.	All minimum entities are stopped. If some entities are not stopped, this indicates those entities have problems.
✖	Partially stopped	There is at least one partially stopped entity and any number of stopped entities.	At least one partially stopped or stopped entity and any number of unavailable entities.
⟹	Running	All entities are running.	All minimum entities are running. If some entities are not running, this indicates that those entities have problems.
◁⟹	Partially started	There is at least one running entity and any number of stopped, partially stopped, or partially running entities.	At least one partially running or running and any number of stopped, partially stopped or unavailable entities.

Deployment environment status

Describes the indicators that show the state of a deployment environment. The warning icon in the topology status indicates the presence of warnings for that deployment environment.

Table 5. States of a topology instance in order of least to most available

Status icon	Warning icon	State	Description
	None	Unknown	The system cannot determine the current state of the deployment environment.
		Incomplete	The deployment environment is not missing any elements but is incomplete in some way. The warning message contains additional details.
		Not configured	The configuration is known and complete but has not yet been generated.
		Deferred configuration	The deployment environment has been generated but deferred configuration has not been completed.
		Unavailable	The deployment environment is complete but at least one function is unavailable.
		Partially stopped	The deployment environment is available but at least one function is stopped or partially stopped.
		Stopped	All functions are stopped.
		Partially running	The deployment environment is available but at least one function is partially running.
		Running	The deployment environment is available and all functions are running.

Editing deployment environments

As your deployment environment needs evolve, you can edit your deployment environment to add resources to the deployment environment.

Before you begin
- A deployment environment must exist.
- You must be at the administrative console of a deployment manager. Navigate to the page by clicking **Servers > Deployment Environments**.
- You must completely stop any nodes you are removing from the deployment environment before removing those nodes.

Required security role for this task: When security and role-base authorization are enabled, you must log in as an administrator or operator to perform this task.

About this task

You edit deployment environments to change the resources that comprise a specific deployment environment. You can:
- Add or remove servers and clusters.
- Change which nodes participate in specific functions.
- Change the configuration of data sources.
- Change authentication aliases.
- Obtain information on how to configure databases or tables if you previously deferred that operation.

Procedure

1. Select the deployment environment to edit. Click on the deployment environment name. The system displays the Deployment Environment Configuration page which lists:
   - **Deployment Environment**
   - **Deployment Environment Pattern**
   - **Description**
   - **Deployment Environment Status**
   - **Deployment Environment Functions**
   - Links to the configuration pages

2. Select the part of the deployment environment to change.

   Select each link until you have completed your changes.

   Under **Additional Properties** choose from these links.
   - Deployment Topology - for deployment environment based on IBM-supplied patterns.
   - Custom Deployment Topology Detail - for custom deployment environments.
   - Deferred Configuration

   Under **Related Items** choose from these links.
   - Data Sources
   - Authentication Aliases

3. Complete the configuration. Choose the option that takes the action you require.

   Note: The system does not complete the configuration until you click **Generate Environment**.

Action	Result
Click OK or Apply	Both options save the configuration. **Apply** leaves you on the current page, **OK** returns you to the Deployment Environments page.
Click Generate Environment	Saves the configuration and starts the configuration process. **Note:** If the deployment environment does not meet the minimum constraints or is incomplete, you will be unable to select this option.

What to do next

Manage the deployment environment.

Related tasks

"Configuring deployment environment functions" on page 66
From a single page, you can add nodes to topologies and assign functions to clusters within a deployment environment based on an IBM-supplied pattern. You can also delete nodes from the deployment environment.

"Configuring custom deployment environments" on page 69
Use the Custom Deployment Topology Detail page to configure your custom deployment environment.

"Configuring a data source for your deployment environment" on page 78
Configure your business integration data source for the first time using the Database Configuration Provider page.

"Configuring authentication aliases for a deployment environment" on page 68
From a single page, you can review or edit all your authentication aliases.

"Configuring deferred configurations for a deployment environment" on page 71
If you must defer the creation of your databases and tables, use the Deferred Configuration page. This page provides instructions on how to locate and run scripts for database and table creation.

Editing the data source configuration

Use the Business Integration Data Source page as a central location for all the data sources in your deployment environment for multiple edits.

Before you begin
- Deployment environments exist on this deployment manager.
- You must be at the administrative console of the deployment manager.

Navigate to this page by clicking **Servers** ⟶ **Deployment Environments**. Select your deployment environment and then click **Additional Properties** ⟶ **Data Sources**.

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an administrator or a configurator to perform this task.

About this task

The Data Sources page is a collection of all the data sources in your deployment environment that you can edit. You cannot add a new data source in the Data Sources page. The number text fields may differ depending on the component and data source provider for each data source.

Important: If you make edits that conflict, such as using a schema name used by another data source, the system displays a warning message. You can still save your changes but the message persists until you resolve the conflict.

Procedure
1. In the Data Source page, select the component that contains the data source you want to edit.
2. Make any changes you need.
3. Click **Apply** or **OK** to save any changes you made.

Editing your data base provider

Use the Database Provider Configuration page to make changes to your database provider.

Before you begin
- Deployment environments exist on this deployment manager.
- You must be at the administrative console of the deployment manager.

Navigate to this page by clicking **Servers** ⟶ **Deployment Environments**. Select your deployment environment and then click **Additional Properties** ⟶ **Data Sources**.

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an administrator or a configurator to perform this task.

About this task

The Database Provider Configuration page shows text boxes which you can edit. Some sections such as **Component Specific Properties** have a different number of text boxes depending on the database provider.

Procedure

1. In the Data Source Configuration page, select the component that contains the data source provider you want to edit.
2. Click **Edit Provider** to open the Database Provider Configuration page.
3. Make the changes you need.
4. Click **Apply** or **OK** to save your changes.

   Related concepts

   ➡ Database specifications

   WebSphere Process Server uses a number of database tables to hold, store and track information. Some of the components that comprise WebSphere Process Server use their own database tables. You can create these database tables during profile creation or you can choose to create them separately using scripts.

   ➡ Common database specifications

   The Common database specifications contain information about supported database types; scripts and their locations; profile creation configuration actions; installation parameters; types of created tables and user ID privileges.

Configuring a data source for your deployment environment

Configure your business integration data source for the first time using the Database Configuration Provider page.

Before you begin

- Deployment environments exist on this deployment manager.
- You must be at the administrative console of the deployment manager.

Navigate to this page by clicking **Servers** ⟶ **Deployment Environments**. Select your deployment environment and then click **Additional Properties** ⟶ **Data Sources**.

Required security role for this task: When security and role-base authorization are enabled, you must log in as an administrator or operator to perform this task.

About this task

The Data Sources page is a collection of all the data sources that are needed in your deployment environment. Use this page to configure the data sources based on the available databases.

The component that needs the data source determines all needed fields based on the **Database Provider** you select, and these needed fields need to be completed. The component fills out the rest of the fields with default values. You may either keep the default values or change them according to your needs. In most cases, the component determines the **Scope** value.

A business integration data source can only be configured once. After you configure the data source and save it, some text boxes will be unavailable and you cannot change the values. All other text boxes within the page can be edited.

Procedure

1. In the **Business Integration Data Source** page, select the data source you want to configure.
2. Click **Edit** if you need to edit additional data source fields that are not shown by default.
3. Fill in the information. For a list of supported database types, see "Database specifications."
4. Click **Apply** or **OK** to save your changes.

   Related concepts

   ➦ Database specifications

   WebSphere Process Server uses a number of database tables to hold, store and track information. Some of the components that comprise WebSphere Process Server use their own database tables. You can create these database tables during profile creation or you can choose to create them separately using scripts.

   ➦ Common database specifications

   The Common database specifications contain information about supported database types; scripts and their locations; profile creation configuration actions; installation parameters; types of created tables and user ID privileges.

Editing a data source in your deployment environment

Use the Data Source page to edit your data source properties.

Before you begin

- Deployment environments exist on this deployment manager.
- You must be at the administrative console of the deployment manager.

Navigate to this page by clicking **Servers** ▸ **Deployment Environments**. Select your deployment environment and then click **Additional Properties** ▸ **Data Sources**.

Required security role for this task: When security and role-base authorization are enabled, you must log in as an administrator or operator to perform this task.

About this task

The Data Sources page is a collection of all the data sources in your deployment environment. You cannot add a new data source in the Data Sources page. You can edit data source information by either clicking on the *Data Source* name or selecting the component. Some text boxes are unavailable and you cannot change these values.

Important: If you make edits that conflict, such as using a schema name used by another data source, a warning message will show. You can still save your changes but the message persists until the conflict is resolved.

Procedure

1. In the Business Integration Data Source page, select the component that contains the data source you want to edit.
2. Click **Edit** and **Data Source** page opens.
3. Edit the information.
4. Click **Apply** or **OK** to save your changes.

Stopping and restarting the deployment manager

After any configuration changes to the deployment manager, you must stop and restart the deployment manager before those changes take effect.

Before you begin

A deployment manager must be started and you must log in to the administrative console.

Required security role for this task: When security and role-base authorization are enabled, you must log in as an administrator or operator to perform this task.

Procedure

1. Choose the method to stop the deployment manager.

Method	Actions
Using the administrative console	1. Click **System Administration → Deployment Manager** 2. Click **Stop**
Using the command line	1. Navigate to the deployment manager's *profile_root*/bin directory. 2. Enter the correct form of stopManager for your operating system. **Note:** If global security is enabled, the system prompts you to enter a user ID and password.

2. Wait for verification that the deployment manager has stopped.
3. Navigate to the deployment manager's *profile_root*/bin directory.
4. Enter the correct form of the startManager command for your operating system.

   Note: If global security is enabled, the system prompts you to enter a user ID and password.

What to do next

Verify the application deployment target cluster can start.

Stopping and restarting a cluster member

Stop and restart your cluster member when you make configuration changes.

Before you begin

1. You must prevent new work from entering the member.
   - If you are using the IBM HTTP Server, change the plugin_cfg.xml file to remove the cluster member for HTTP traffic. If you are using another HTTP server, follow the directions for your HTTP server to remove the cluster member.
   - For IIOP traffic, set the runtime weight to zero for the cluster member.
   - Quiesce the service integration bus.
2. Work currently destined for the cluster member must complete. Either wait a period of time or use Performance Monitoring Infrastructure counters to determine when the cluster completes all queued work.

About this task

Some configuration changes require you to stop and restart some of the server processes before the configuration change can take effect. This involves the stopping and restarting of deployment manager, cluster member, and the node agent. The following procedure stops and restarts a cluster member.

Note: All command files (alternatives to using the administrative console) are located in the install_root/bin subdirectory.

Procedure

1. From the administrative console select **Servers**, then **Application servers**.
2. Select the servers or cluster members to be stopped.
3. Click **Stop**.
4. Wait for the servers or cluster members to stop.
5. Select the servers or cluster members to be restarted and click **Start**.
6. Wait for the servers or cluster members to start.

What to do next

Note: The cluster members alternatively can be stopped and subsequently restarted from the command line using the stopServer and startServer commands for your operating sytem, respectively, or from the administrative console's cluster panel by selecting **Servers > Clusters >** *cluster_name* **>** *cluster_member_name*.

Starting and stopping deployment environments

You can start or stop deployment environments based on IBM-supplied patterns directly from the administrative console.

Before you begin

Required security role for this task: When security and role-base authorization are enabled, you must log in as an administrator or operator to perform this task.

A deployment environment must exist.

About this task

Use these steps whenever you want to start or stop a deployment environment.

Restriction: You cannot use this procedure for custom deployment environment; you must start and stop the clusters individually.

Procedure

1. Log in to the administrative console of the deployment manager that controls the deployment environments you are starting.
2. Navigate to the deployment environment administrative console page. Click **Servers › Deployment Environments**.
3. Click the check boxes next to the names of the deployment environments you are starting or stopping.
4. Start or stop the selected deployment environments.

Action	Result
Click Start.	The deployment manager starts the clusters that make up the deployment environments.
Click Stop.	The deployment manager stops the clusters that make up the deployment environments.

> **Note:** This process can take several minutes depending on the size of your deployment environment.

Results

The display refreshes to indicate the status of the deployment environments. If successful, a deployment environment will have an indication of the deployment environment state, as described in Deployment environment status.

Related concepts

 Deployment environments

A deployment environment is a collection of configured clusters, servers, and middleware that collaborates to provide an environment to host Service Component Architecture (SCA) interactions. For example, a deployment environment might include a host for message destinations, a processor of business events, and administrative programs.

Related tasks

"Starting and stopping the administrative console" on page 15
To access the administrative console, you must start it and then log in. After you finish working in the console, save your work and log out.

Related information

 Using the Administration Thin Client

Exporting deployment environment definitions

Exporting deployment environment definitions helps you speed implementing deployment environments by minimizing the configuration you need to perform on each deployment manager. The exported deployment environment can then be used on other deployment managers as a template for the deployment environment. It also allows you to replicate the same deployment environment configuration on a large scale.

Before you begin

- Fully define at least one deployment environment on a deployment manager.
- You must be at the administrative console of a deployment manager. Navigate to the page by clicking **Servers > Deployment Environments**.

Required security role for this task: When security and role-base authorization are enabled, you must log in as an administrator or operator to perform this task.

About this task

If you are implementing a number of deployment environments based on the same designs, export those deployment environment definitions to ease the configuration of the deployment environments on other deployment managers.

Procedure

1. On the Deployment Environments page, click the check boxes next to the deployment environment definitions you are exporting.
2. Click **Export**. The system response depends on whether you select:

   Single deployment environment
   > You are prompted for the name for the exported file. The default name is *deployment_environment_name*.xml. You can specify the full file path at this point.

   Multiple deployment environments
   > You are prompted for the output directory to place the exported compressed file that contains the deployment environment definitions. By default, the system names the compressed file *first_env_name*.zip.
3. Examine the file path to make sure the system created the files.

What to do next

You can now take the exported files to other deployment managers and import them.

Note: You cannot directly import a compressed file, you must extract the contained deployment environment definitions into the target file system.

Exporting deployment environment definitions using the command line

You can export deployment environment definitions through the wsadmin command interface. This capability allows you to use a script to export multiple deployment environment definitions from a deployment manager, which enables you to replicate working configurations to other deployment managers.

Before you begin

- You must be at the deployment manager from which you are exporting the deployment environment definitions.

Required security role for this task: When security and role-base authorization are enabled, you must log in as an administrator or operator to perform this task.

About this task

Use the command line to export deployment environment definitions when:
- You are exporting multiple deployment environment definitions and prefer not to use the administrative console.
- You are using a single deployment environment definition as a template for multiple deployment environments.
- You decide the number of deployment environment definitions to export is large enough to warrant using a script for the task, in which case you would place the appropriate commands in the script and run that file.

Procedure
1. Log into the deployment manager with a userid that has the correct authorization to enter administrative commands.
2. Open a command line. Use the correct method for the operating system running the deployment manager.
3. Enter the wsadmin environment.
4. Enter the exportDeploymentEnvDef command to export the deployment environment definition from the deployment manager to an output file. The file name will be of the form *depEnvName*.xml

This example exports deployment environment myDepEnv on host myDmgr with global security enabled.

```
wsadmin -connType SOAP -host myDmgr -port 8879 -user dmgradmin -password dmgrpass
> $Admintask exportDeploymentEnvDef -exportDirectory c:/dmgr01/DeploymentEnvs
        -topologyName myDepEnv
```

Note: If you disable global security, you do not need to provide a user ID and password.

> **Related reference**
>
> ⏏ exportDeploymentEnvDef command
>
> Use exportDeploymentEnvDef to export topologies from the deployment manager.

Importing deployment environment definitions

You can import an existing deployment environment definition from another deployment manager to use as a base for a new deployment environment.

Before you begin
- You must be at the administrative console of a deployment manager. Navigate to the page by clicking **Servers > Deployment Environments**.
- You must have a copy of an exported deployment environment definition from another deployment manager. You must be able to access the file from the deployment manager to which you are importing the deployment environment definition.

- The deployment manager importing the deployment environment definition must at least support all the functions defined in the deployment environment definition. For example, you can import a deployment environment definition created on a WebSphere ESB deployment manager into a WebSphere Process Server deployment manager but not the reverse.

Required security role for this task: When security and role-base authorization are enabled, you must log in as an administrator or operator to perform this task.

Important: You cannot import multiple deployment environment definitions from a compressed file at the same time. You must extract the definitions from the compressed file and then import the definitions one at a time.

About this task

Importing an existing deployment environment definition that closely matches your needs can minimize the amount of time you spend configuring a deployment environment. If there is an environment that closely matches your needs, export that deployment environment definition and then import it into the deployment manager you are configuring.

Important: Clicking configure on any panel in the wizard will configure the deployment environment with the current values.

Procedure

1. On the Deployment Environments page, click **Import**. The system launches the Deployment Environment Configuration Wizard with **Load an external topology definition** selected.
2. Indicate the location of the file that contains the deployment environment definition to import. Either type the full path in the entry field or click **Browse** to open a file dialog to select the file.

   Note:
3. Click **Next** to load the configuration and launch the Import deployment environment wizard.

   The wizard displays the Select Nodes page, unless all the node names match currently federated notes. If all nodes match, the wizard displays the Database page.
4. Optional: From the list of possible nodes, select the nodes to include in the deployment environment and click **Next**.

   To include a node, select the checkbox next to the node name. Use **Node Mapping** to map the selected node to another node name.

   Important: Next will not be available if the nodes selected do not meet the constraints imposed by the imported deployment environment definition. For example, if there is a requirement for the deployment environment to contain a node named "Mandatory_Node" and 3 other nodes with any name, you will be unable to continue until you configure "Mandatory_Node" and select 3 other nodes.
5. Configure the databases for the deployment environment then click **Next**.

   On this page, define the database information for the components included in this deployment environment. Where possible, the wizard supplies default information for the parameters but change those values to match the values that you defined when you planned the environment.

6. Optional: Define the Business Process Choreographer configuration and then click **Next**.

   On this page you specify the values for:

   - Context roots
   - Security roles
   - Authentication aliases
   - Human task manager mail session, if desired

   Note: This page is only displayed if **WPS** is the **Runtime Capability**.

   The wizard displays the Business Rules Manager page.

7. Optional: Specify the context root for the Business Rules Manager and click **Next**.

   Note: This page is only displayed if **WPS** is the **Runtime Capability**.

   The wizard displays the Summary page.

8. Verify that the information on the Summary page is correct and click **Finish and Generate Environment** to save and complete the configuration of the deployment environment. To exit without completing the configuration click, **Finish**.

Results

When the configuration completes, you can examine the configuration files to view the changes made.

What to do next

Either save the changes to the master configuration or discard the changes.

    Related concepts

    ➙ Deployment environment patterns

A deployment environment pattern specifies the constraints and requirements of the components and resources involved in a deployment environment. The patterns are designed to meet the needs of most business requirements and are intended to help you create a deployment environment in the most straightforward way.

    ➙ Deployment environment functions within deployment patterns

To design a robust deployment environment, you need to understand the functionality each cluster can provide in a particular IBM-supplied deployment environment pattern or a custom deployment environment. This knowledge can help you make the correct decisions as to which deployment environment pattern best meets your needs.

    ➙ Database specifications

WebSphere Process Server uses a number of database tables to hold, store and track information. Some of the components that comprise WebSphere Process Server use their own database tables. You can create these database tables during profile creation or you can choose to create them separately using scripts.

"Business rules manager" on page 11

The business rules manager is a Web-based tool that assists the business analyst in browsing and modifying business rule values. The tool is an option of

WebSphere Process Server that you can select to install at profile creation time or after the initially installing the server.

Related tasks

"Exporting deployment environment definitions" on page 83
Exporting deployment environment definitions helps you speed implementing deployment environments by minimizing the configuration you need to perform on each deployment manager. The exported deployment environment can then be used on other deployment managers as a template for the deployment environment. It also allows you to replicate the same deployment environment configuration on a large scale.

Related information

↪ Configuring Business Process Choreographer

Importing deployment environment definitions using the command line

You can import deployment environment definitions through the wsadmin command interface. This capability allows you to use a script to import multiple deployment environment definitions to a deployment manager, which enables you to replicate working configurations to other deployment managers.

Before you begin
- You must have exported the deployment environment definition.
- You must be at the deployment manager to which you are importing the deployment environment definitions.
- Make sure that a deployment environment with the same name as the deployment environment definition you are importing exists on this deployment manager.
- The deployment manager importing the deployment environment definition must at least support all the functions defined in the deployment environment definition. For example, you can import a deployment environment created on a WebSphere ESB deployment manager into a WebSphere Process Server deployment environment but not the reverse.

Required security role for this task: When security and role-base authorization are enabled, you must log in as an administrator or operator to perform this task.

About this task

Use the command line to import deployment environment definitons when:
- You are importing multiple deployment environment definitions and prefer not to use the administrative console.
- You are using a single deployment environment definition as a template for multiple deployment environments.
- You decide the number of deployment environment definitions to import is large enough to warrant using a script for the task, in which case you would place the appropriate commands in the script and run that file.

Procedure
1. Open a command line. Use the correct method for the operating system running the deployment manager.

Note: If global security is on, you may be prompted for a user ID and password.

2. Copy the deployment environment definition file you are importing to the system.

3. Enter the wsadmin environment

4. Enter the importDeploymentEnvDef command to import the deployment environment definition from the file you just copied to the deployment manager. You can rename the deployment environment when you import it.

This example imports deployment environment myDepEnv and renames it eastDepEnv on deployment manager myDmgr with global security enabled.

```
wsadmin -connType SOAP -host myDmgr -port 8879 -user dmgrAdmin -password -dmgrPass
> $AdminTask importDeploymentEnvDef
  -filePath c:/dmgr01/importedEnvironments/myDepEnv.xml
  -topologyName eastDepEnv
```

Note: If you disable global security, you do not need to specify a user ID and password.

What to do next

Validate the imported topologies.

Related reference

⤶ importDeploymentEnvDef command

Use importDeploymentEnvDef to import topologies into the deployment manager.

Related information

⤶ Managing node agents

Removing deployment environments

When you no longer require the ability to manage the resources represented by a specific deployment environment as a group, remove that deployment environment definition from the deployment manager.

Before you begin

- You must be at the administrative console of a deployment manager. Navigate to the page by clicking **Servers > Deployment Environments**.
- Deployment environments exist on this deployment manager.
- Consider exporting the deployment environment definition for recovery purposes.

Required security role for this task: When security and role-base authorization are enabled, you must log in as an administrator or operator to perform this task.

About this task

Removing a deployment environment removes the management entity that represents the configurations that comprise the deployment environment. Deleting the deployment environment does not remove or change the configuration of the various servers, nodes, and clusters that comprise the deployment environment.

Deleting deployment environments may be the final phase of moving a
deployment environment from one deployment manager to another.

Procedure

1. On the Deployment Environment page, select the deployment environments to
   remove. Click the check box next to the deployment environment name.
2. Click **Remove**.

Results

The system removes the deployment environment from the display.

Click **Save** to save this change to the master configuration or **Discard** to prevent
the update of the master configuration.

> **Related tasks**
>
> "Exporting deployment environment definitions" on page 83
> Exporting deployment environment definitions helps you speed implementing
> deployment environments by minimizing the configuration you need to
> perform on each deployment manager. The exported deployment environment
> can then be used on other deployment managers as a template for the
> deployment environment. It also allows you to replicate the same deployment
> environment configuration on a large scale.

Chapter 5. Administering Business Process Choreographer

For information on how to administer Business Process Choreographer, go to the WebSphere Process Server for Multiplatforms, version 6.1, information center and review the topics under **Administering WebSphere Process Server > Administering Business Process Choreographer**. You can also find this information in the *Business Process Choreographer* PDF.

Chapter 6. Configuring and administering the Common Event Infrastructure

For information on how to configure and administer the Common Event Infrastructure, go to the WebSphere Process Server for Multiplatforms, version 6.1, information center and review the topics under **Administering WebSphere Process Server > Configuring the Common Event Infrastructure** and **Administering WebSphere Process Server > Administering the Common Event Infrastructure**. You can also find this information in the *Common Event Infrastructure* PDF.

Chapter 7. Administering service components

Use the topics in this section to manage service components.

For information on administering business processes and human tasks, see the topics under **Administering WebSphere Process Server > Administering service components** in the WebSphere Process Server for Multiplatforms, version 6.1, information center or refer to the *Business Process Choreographer* PDF.

Overview of troubleshooting business state machines

You can view the correlation set values and display states variables to debug and administer business state machine instances.

A business state machine is used to represent an event-driven business process. Within a business state machine there are many instances. You can administer and debug business state machine instances using:

- correlation set properties
- display states

Correlation set properties

To distinguish one business state machine instance from another, a correlation set is used to uniquely identify a state machine instance. For example, a correlation set properties could be a customer ID and state. If you want to administer a particular instance, you need the values of the correlation set properties. Correlation set properties are defined in WebSphere Integration Developer and viewed in Business Process Choreographer Explore.

You can define only one correlation set in WebSphere Integration Developer. Multiple correlation sets are not allowed.

Display states

A display state variable indicates the current state of a particular business state machine instance. Knowing the last committed state is useful for debugging or administrating business state machines. Display states are defined in WebSphere Integration Developer and viewed in Business Process Choreographer Explorer.

The display state variable may not always show the most current state of a business state machine instance. If an instance is actively processing an event, the in-memory copy of the display state variable may be different from the last committed value. What you see in Business Process Choreographer Explorer is the display state value that was last written to disk. If a business state machine instance is processing an event, the in-memory value of the variable will not be written to disk until the transaction is completed.

Finding business state machine instances

View correlations set properties to find and administer a particular business state machine instance.

Before you begin

Define the correlation set in WebSphere Integration Developer and save the module. Deploy the module to the server.

About this task

The values of correlation set properties distinguish one business state machine instance from another throughout its life cycle. If you need to end a particular business state machine instance, the values of correlation set properties will identify the correct instance. Use this procedure to view the correlation set properties through the Business Process Choreographer Explorer.

Restriction: You can have only one correlation set defined for a business state machine. Multiple correlation sets are not allowed.

Procedure

1. Under **Process Templates**, select the process template that represents your business state machine.
2. Under **Process Template Name** select your process template and click on **Instances** to view all existing instances still active in your system.
3. For each instance, click on the instance and then click on the **Query Properties** tab to view the correlation set properties under **Property Name**.

What to do next

Perform your administrative tasks.

Viewing display states

View display states to administer or debug business state machine instances.

Before you begin

Initialize the display state variable in WebSphere Integration Developer and save the module. Deploy the module to the server.

About this task

The display state variable allows you to view the current state of an active business state machine instance. For example, if a business state machine instance is not responding as expected, you can view the active business state machine instance to determine the current state and debug the problem. You need the values of the correlation set properties of that active business state machine instance. To view the current state of an active business state machine instance, do the following in Business Process Choreographer Explorer.

Procedure

1. Under **Process Templates**, select the process template that represents your business state machine.
2. Under **Process Template Name** select your process template and click on **Instances** to view all existing instances still active in your system.
3. For each instance, click on the instance and then click on the **Query Properties** tab to view the correlation set properties and display states under **Property Name**.

What to do next

Perform your administrative tasks.

Administering business rules and selectors

Business rules and selectors provide flexibility in a business process by changing the results of a process based on a criteria. Before installing applications that contain business rules and selector components, you must install the business rules dynamic repository. You can install the business rules dynamic repository for a stand-alone server or for network deployment.

Whenever you install a module that contains business rules or selectors or change business rules and selectors on the server, the updates are logged in the system log or another log that you specify when you configure business rule and selector audit logging.

Configuring the business rule and selector audit log

You can configure the server to use different values than the default values for the log that keeps track of new, changed, and deleted business rules and selectors. Changing the configuration can help you conserve resources on your server.

Before you begin

You must be at the administrative console to perform this task.

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as a configurator to perform this task.

About this task

After you have run your server in production for a while, you may have determined that the default values the server uses for the business rules and selectors audit log need adjustment.

To configure the business rule and selector audit log, perform the following steps.

Procedure

1. Navigate to the **Business Rules and Selectors Auditing** page by clicking **Servers > Application servers >** *servername* **Business Rules > Business Rules and Selectors Auditing**.
2. Do one of the following depending on the type of change you want.

Type of change	Actions
Immediate	1. Select the **Runtime** tab.
	2. Enter the desired changes.
	3. **Optional:** To make the changes permanent, copy them to the repository by selecting **Save runtime changes to configuration as well**.
	4. Click **OK** to make the changes and return to the previous page or **Apply** to make the changes and remain on this page.

Type of change	Actions
Delayed	1. Select the **Configuration** tab.
	2. Enter the desired changes.
	3. Click **OK** to make the changes and return to the previous page or **Apply** to make the changes and remain on this page.
	4. When you want the changes to take effect, restart the server.

Results

The audit log takes the attributes you specified.

Note: You may need to modify the configuration for business rules and selector auditing due to the way the server user identity is specified when security is enabled with WebSphere Application Server 6.1. If the default value is used for the server user identity, an automatically generated server identity value is recorded in the audit record for the user when any auditable action involving business rules or selectors is performed when the application containing the business rules or selectors is started after business rule or selector installation. An auditable action occurs when a business rule or selector artifact is changed through application startup after install, management clients, or import or export through the administrative console. The generated value may not match the format of other user IDs used in other audit records, and you may want a more consistent value.

You can specify a server identity by selecting the option to use a "Server identity that is stored in the repository," which will associate a user ID that is in the user repository with the server process. The audit records will use this identity when auditable actions involving business rules or selectors are performed when the application containing the business rules or selectors is started after the business rule or selector artifacts are installed in the repository.

The server identity value has no effect on audit actions involving changes through management clients such as the business rules manager or other administrative actions such as exporting or importing business rule groups. For these actions, the audit record will use the authenticated user.

For more information on changing the server user identity, see the topics under Securing applications and their environment and the WebSphere Application Server WebSphere Application Server Network Deployment, version 6 Security documentation.

Configuring business rule and selector auditing using commands

Use commands to configure business rule and selector auditing when you need to change any of the characteristics while a server is running.

Before you begin

You must run these commands from a command line environment for the server.

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as a configurator to perform this task.

About this task

There may be occasions when you need to change how many servers audit business rules and selectors and cannot restart the servers involved. Using the command line, you can automate configuring the servers in a batch mode. The following tasks show how to use commands to configure one server.

Important: These settings are not saved if you restart the server. To save the configuration after entering these commands, you must use the administration console. Select **Servers > Application servers >** *servername* **> Business Rules > Business Rules and Selectors Auditing > Runtime** or **Servers > Application servers >** *servername* **> Selectors > Business Rules and Selector Auditing > Runtime**.

To configure business rule and selector auditing using commands, perform the following steps.

Note: The following task configures server `server1`. If the server is not named server1, replace `server1` below with the name of the server. All of the steps beginning at step 3 could be placed in a jacl script and run that way.

Procedure

1. Enter the administrative environment.

    `wsadmin`

2. Decide whether you are configuring audit logging or changing an existing configuration.

Task	Command
Configuring audit logging	`set mbean [$AdminControl queryNames *:*,name=CustomizationAuditMBean,process=server1]`
Changing audit logging configuration	`set auditconfig [$AdminConfig list AuditLog]`

3. Enter the appropriate commands.

Commands to configure or change audit logging

Important: When entering commands that change an existing configuration, you must save the changes. The changes do not take effect until you restart the server.

The following are the commands you can enter:

$AdminControl invoke $mbean getSeparateAuditLogEnabled
 Use to determine whether logging is occurring to a separate audit log.

$AdminControl invoke $mbean setSystemOutAuditLogEnabled {*boolean***}**
 Use to enable or disable logging to the SystemOut.log file. *Boolean* can either be true or false.

$AdminControl invoke $mbean getSeparateAuditLogFileName
 Use to determine the file name of the separate audit log.

$AdminControl invoke $mbean setSeparateAuditLogFileName {*filename*}
> Use to set the name of the new log file, for example, MyAudit.log.

$AdminControl invoke $mbean getSeparateAuditLogFileRolloverSize
> Use to determine the size of the audit log.

$AdminControl invoke $mbean setSeparateAuditLogFileRolloverSize *integer*
> Use to set the size of the audit log before the system rolls it over into a history file. The size is in megabytes.

$AdminControl invoke $mbean getSeparateAuditLogFileMaxNumberOfBackupFiles
> Use to determine the number of audit log history files.

$AdminControl invoke setSeparateAuditLogFileMaxNumberOfBackupFiles *integer* Use to set the number of audit log history files.

$AdminControl invoke $mbean setSeparateAuditLogEnabled {*boolean*}
> Use to start or stop logging to a separate log file. *Boolean* can either be true or false.

$AdminConfig showall $auditconfig
> Use to show the current audit log configuration.

$AdminConfig modify $auditconfig {{separateAuditLogEnabled true}}
> Use to enable logging to a separate audit log.

$AdminConfig modify $auditconfig {{systemOutAuditLogEnabled false}}
> Use to disable auditing to the system.Out file.

$AdminConfig modify $auditconfig {{customAuditLog {{maxNumberOfBackupFiles 7} {rolloverSize 7}}}}
> Use to change the number of audit log history files and the size of the audit log file.

$AdminConfig modify $auditconfig {{customAuditLog {{fileName MyAudit.log}}}}
> Use to change the name of the audit log file.

$AdminConfig save
> Use to save the configuration.

Save these changes by opening the administrative console and selecting **Servers > Application servers >** *servername* **> Business Rules > Business Rules and Selectors Auditing > Runtime** or **Servers > Application servers >** *servername* **> Selectors > Business Rules and Selector Auditing > Runtime**. Alternatively, enter $AdminConfig save.

Note: You may need to modify the configuration for business rules and selector auditing due to the way the server user identity is specified when security is enabled with WebSphere Application Server 6.1. If the default value is used for the server user identity, an automatically generated server identity value is recorded in the audit record for the user when any auditable action involving business rules or selectors is performed when the application containing the business rules or selectors is started after business rule or selector installation. An auditable action occurs when a business rule or selector artifact is changed through application startup after install, management clients, or import or export through the administrative console. The generated value may not match the format of other user IDs used in other audit records, and you may want a more consistent value.

You can specify a server identity by selecting the option to use a "Server identity that is stored in the repository," which will associate a user ID that is in the user repository with the server process. The audit records will use this identity when auditable actions involving business rules or selectors are performed when the application containing the business rules or selectors is started after the business rule or selector artifacts are installed in the repository.

The server identity value has no effect on audit actions involving changes through management clients such as the business rules manager or other administrative actions such as exporting or importing business rule groups. For these actions, the audit record will use the authenticated user.

For more information on changing the server user identity, see the topics under Securing applications and their environment and the WebSphere Application Server WebSphere Application Server Network Deployment, version 6 Security documentation.

Considerations for modules containing business rules and selectors

Here is some information to consider when you install or delete modules that contain business rules and selectors.

Business rules and selectors add flexibility to your modules. The added flexibility affects how you install or delete a module because the server saves business rules and selectors in a central repository.

Considerations for changing business rules or selectors

You can change business rules and selectors in your production environment without reassembling and reinstalling the affected modules. These changes are made directly to the repository and are not copied into any of the files that contain the business rules or the selectors. After making a change to business rules or selectors, export the business rules or selectors and import them into your development environment. If you are unfamiliar with exporting and importing business rules and selectors, see the topics that describe those tasks.

Considerations for replacing a module containing business rules or selectors

When you replace a module that contains business rules or selectors, the server overwrites the copies of the business rules and selectors in the repository. When you replace a module, any changes that you made dynamically are lost. To prevent that loss, export the business rules and selectors used by the module, re-import them into your development environment, and rebuild the module before replacing the module on your production system.

If you have made changes to the business rules or selectors implemented by one module, other modules running in the server may need the current copies of the business rules or selectors. If this is the case, you will have to configure different repositories so that the updated module has no effect on the other modules when you install that module in the server. The topic "Configuring the environment" describes configuring the databases.

Considerations for deleting a module containing business rules or selectors

When you delete a module that contains business rules or selectors from the server, the server does not remove the business rules and selectors from the repository. It keeps these artifacts because it cannot determine if another application or module requires the rules.

If you determine that there is no requirement for a business rule or selector, remove it from the repository. "Removing business rule and selector data from the repository" describes how to clear out unneeded business rules or selectors.

Removing business rule and selector data from the repository

When you uninstall an application that uses business rules or selectors, the server does not remove these artifacts from the repository. Delete the unused artifacts from the database manually after you uninstall applications that use them. Remove the artifacts using the tools supplied by the database platform of your repository. The reason this is done is that business rules and selectors contain business logic which may have been updated when the application was installed, and we do not want to delete this important business data when the application is removed.

Before you begin

Make sure to uninstall all copies of applications that use the business rules or selectors that will be removed. You can back up business rule or selector artifacts before deleting them by exporting them out of the server using the administrative console or wsadmin command.

About this task

When you install an application containing business rule or selector artifacts, the server stores these artifacts in database tables so that you can dynamically update them without changing the application. This also allows other servers to share these artifacts. When you uninstall an application, the server does not automatically remove these artifacts from the database tables because the application may still be installed and running on another server. Deleting the artifacts from the database causes the other running copies of the application to fail when they try to use business rules or selectors.

To remove unneeded business rule and selector artifacts from the repository, perform the following steps.

Procedure

1. Locate the following database tables from which you will delete rows:

   BYTESTORE
   > The main table that contains the business rule and selector artifacts

   BYTESTOREOVERFLOW
   > The overflow table for the main table

   APPTIMESTAMP
   > The table that holds a timestamp of installed applications that contain business rule and selector artifacts

   CUSTPROPERTIES
   > The table that holds custom user-defined properties and system properties for a business rules group, rule set, or decision table.

2. Using the tools for your database platform, follow these steps to delete all business rule and selector artifacts for a given application:

   a. Find all of the rows in the BYTESTORE table where the **APPNAME** column is the same as the name of the application.

   b. Record the values of the primary key columns for all of the rows found. The primary key columns for the BYTESTORE table are **ARTIFACTTNS**, **ARTIFACTNAME**, and **ARTIFACTTYPE**.

   c. Delete the rows found in step 2a from the BYTESTORE table.

   d. For each set of primary key values recorded in step 2b, find the rows in the BYTESTOREOVERFLOW table that have the same values in the corresponding columns.

      Note: For a given set of primary key values, there may be zero, one, or more than one row in the BYTESTOREOVERFLOW table.

   e. Delete the rows found in step 2d from the BYTESTOREOVERFLOW table.

   f. For each set of primary key values recorded in step 2b, find the rows in the CUSTPROPERTIES table that have the same values in the corresponding columns.

   g. Delete the rows found in step 2f from the CUSTPROPERTIES table.

   h. Delete the row in the APPTIMESTAMP table where the **APPNAME** column equals the name of the application.

Results

You have removed the unneeded business rules and selector artifacts from the database tables.

Overview of business rules

Use business rules to control the behavior of a business practice.

What is a business rule?

A business rule is anything that imposes structure upon or controls the behavior of a business practice. A rule can enforce business policy, establish common guidelines within an organization, or control access in a business environment.

When to use a business rule

Use business rules to officiate over frequently changing business practices that can come from within a business or mandated from outside a business, such as regulatory agencies. Some typical uses for business rules are as follows:
- Determining current interest rates
- Calculating discounts for products
- Calculating sales tax
- Determining special groups such as senior citizens or preferred customers

How to use business rules

Develop and deploy business rules using the Eclipse-based business rules editors in WebSphere Integration Developer. Manage and modify business rule values using the Web-based business rules manager, which is an option of WebSphere Process Server. For more information about these tools, see the appropriate topics

in the WebSphere Integration Developer Information Center and the WebSphere Process Server Information Center, respectively.

Displaying business rule components

Displaying business rule components is the first step in administering a business rule group. From the display you can export or import any or all of the business rule groups or display the tables that comprise the business rule groups.

Before you begin

You must be at the administrative console for WebSphere Process Server to perform this task.

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an administrator or a configurator to perform this task.

About this task

To determine which business rule groups exist in your server, perform the following steps.

Procedure

1. From the administrative console, click **Servers > Application servers**.
2. Click *servername* to select the server from the server list that displays business rules.
3. Click **Business rules** under Business Integration.

Results

The console displays a list of all the business rule components defined with a description of each group.

Exporting business rules using the administrative console:

Export business rule components when you have made changes to the business rule tables. This will create a file that you can import into your development environment, thereby keeping the development artifacts synchronized with the actual production system artifacts.

Before you begin

Before starting this task, you need to display your business rule components as described in "Displaying business rule components." Click **Servers > Application servers > *servername* > Business rules > Business rules**.

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an administrator or a configurator to perform this task. When security is not enabled, you must log in to the administrative console with a user ID.

About this task

To export business rules using the administrative console, perform the following steps.

Tip: You can also export business rules using the command line. See "exportBusinessRuleArtifacts.jacl command."

Procedure

1. Select the check boxes next to one or more business rule groups and click **Export**.

   The browser displays a list of HTML links to the business rule groups you chose. This is the Business rules export page. Each business rule group has a file extension of `.zip`.

2. Download the files to your system by clicking each file name. When the system prompts you to save the file, click **OK**.

   Note: If you choose to, you can rename the files as you download them.

3. Click **Back** to return to the list of business rule groups.

Results

The system saves the files where you specified. You can then copy them to your test system.

You must import the files into your WebSphere Integration Developer environment. For more information, see the WebSphere Integration Developer Information Center.

Importing business rules using the administrative console:

Import business rules in order to update installed business rules without reinstalling an application.

Before you begin

You must be at the administrative console and have the location of a compressed file created by the export facility.

Before importing business rules, make sure the following are true or the import will fail:
- The file has an extension of `,zip`.
- The compressed file was created by exporting the business rules from a server.
- The application that uses the business rules group has already been installed on a server in the cell.

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an administrator or a configurator to perform this task.

About this task

Import business rules when you have made changes to business rules in use by installed applications and you are ready to bring those changes into another cluster or server. You can also use this facility to synchronize your development environment with changes in the production environment.

To import business rules using the administrative console, perform the following steps.

Tip: You can also import business rules using the command line. See "importBusinessRuleArtifacts.jacl command."

Procedure

1. Display the business rules on the server to which you are importing the business rules. Click **Servers > Application Servers >** *servername* **> Business rules > Business rules**.
2. Click **Import**.
3. Specify the path to the file on the Preparing for importing business rules page.

What to do next

Display the business rules to verify the changed rules.

Business rules manager

The business rules manager is a Web-based tool that assists the business analyst in browsing and modifying business rule values. The tool is an option of WebSphere Process Server that you can select to install at profile creation time or after the initially installing the server.

Business rules are designed and developed in WebSphere Integration Developer using if/then rule sets and decision tables to implement their operations. Business rules can also be created in WebSphere Business Modeler; however Modeler only supports the creation of business rule tasks, which become rule sets when exported out of Modeler. The rule sets and decision tables are set into templates. The templates control which aspects of a business rule you can modify and by exactly how much. They define the structure of if/then rules, condition cases, and actions for decision tables.

The templates provide the mechanism for business rule runtime authoring in the business rules manager. Using the template, you can modify business rule values, create a new rule within a rule set or a new condition or action within a decision table, and publish changes to business rule definitions at run time.

Business rules are organized into business rule groups. Business rule groups are used to interface to and invoke rules. Rule sets and decision tables are never invoked directly.

For more information about building and deploying business rules, see the WebSphere Integration Developer Information Center.

How the business rules manager works

The business rules manager is the main WebSphere Process Server tool that a business analyst uses for runtime rule authoring.

Use the business rules manager to perform the following tasks:
- Retrieve a copy of a business rule from the repository
- Browse and edit a business rule
- Publish a business rule to the repository

The following figure shows how the business rules manager calls and publishes rules.
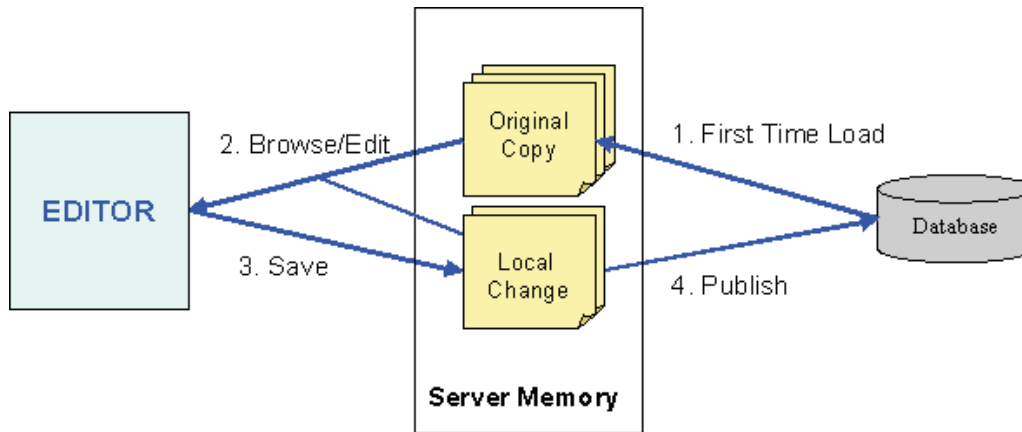
Figure 5. Business rules manager sequence of events

After you log on to the business rules manager, the following events occur when you modify a business rule.

1. When you select a business rule, the business rules manager accesses the business rule group from the repository and stores it in the server memory as an original copy.
2. The business rule group and rule logic are available for editing.
3. You can save changes to a rule set, decision table, and business rule group as a copy in the server memory.
4. You publish the local copy back to the data source. Alternatively, you can cancel the changes with no updates being performed.

Considerations for installing the business rules manager

If you are planning to use the business rules manager in a distributed environment, you must understand the concepts of cells, nodes, and clusters and how to set up the business rules manager for best performance during run time.

The application server is organized on the concept of cells, nodes and servers. In a stand-alone server configuration, a cell contains one node, and each node contains one server. System administration applications and user applications all run in the same server. In a stand-alone server configuration, you can install the business rules manager in the same application server, and it can be accessed by the default URL.

In a distributed server configuration, you can configure a cell to contain multiple nodes, and each node can contain multiple application servers. Each cell constitutes a single administrative domain. With this configuration, you can use central administration, workload management, and failover configuration for the entire domain.

For best performance in a distributed server configuration, install the business rules manager on the administrative deployment target, an application server in the cell where business administration services are centrally hosted. This server is typically the same server that hosts the Common Event Infrastructure service.

Within a cell, all servers use and share a single business rules repository. When you access the business rules repository, you can access all dynamic business rule artifact definitions regardless of the exact location where the business application is installed.

Because of this central storage for all business rules in the cell at run time, you can deploy the business rules manager to any application servers in the cell, and the business rules manager gives a consistent view of all business rules within the cell. However, because of high-availability considerations, it is recommended that system administrators deploy the business rules manager into the administrative deployment target, a dedicated application server in the cell where business administration services are centrally hosted. The administrative deployment target server is the same server where the Common Event Infrastructure service and other business administrative applications are installed. With this configuration, when you require high availability, you can cluster the administrative deployment target server to provide a scalable solution to the application users.

Installing the business rules manager using the administrative console:

You can install the business rules manager as an enterprise application on WebSphere Process Server to manage business rules during run time. For WebSphere Process Server 6.1 and higher, you can install the business rules manager simultaneously when creating a WebSphere Process Server profile by selecting the check box on the Business Rules Manager Configuration page of the Profile Management tool. Alternatively, you can install the business rules manager using three other methods: through the configuration page of the administrative console, by using the JACL command for your operating system, or by using the Admin Tasks command (this method is for WebSphere Process Server 6.1 and higher). For more information, see the individual topics for each installation method.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an administrator or a configurator to perform this task.

About this task

To install the business rules manager using the administrative console, perform the following steps.

Procedure
1. Ensure that the administrative console is running.
2. In the navigation pane click **Servers > Application servers** or **Clusters**.
3. Select the name of your server or cluster target.
4. On the Configuration-tabbed page, under **Business Integration**, expand **Business Rules** and click **Business Rules Manager Configuration**.
5. Under **General Properties** select the **Install business rules manager** check box.

   Note: If the business rules manager has already been installed, the checkbox will be checked but grayed out as it is not possible to uninstall the business rules manager from this page. However, you can uninstall it manually by going to the list of applications and uninstalling it from there.
6. In the **Context root** field either accept the default context root of /br or type a custom context root for the business rules manager URL.
7. Click **OK**.
8. Save the configuration.

In the navigation pane click **Applications > Enterprise Applications** and select **Start Business Rules Manager**.

Installing the business rules manager using the JACL command:

You can use a JACL command for Windows, Linux®, i5/OS®, as an alternative to using the administrative console for installing the business rules manager. Using a JACL command is possible if you did not already install the business rules manager when you installed WebSphere Process Server and created profiles.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, you must be authenticated with a user ID that has been assigned to the administrator or configurator role to perform this task.

About this task

To install the business rules manager using the JACL command, perform the following steps.

Procedure
1. Ensure that WebSphere Process Server is started.
2. Open the shell environment or command prompt for your operating system, and go to the *install_root*/bin directory (*install_root*\bin directory for Windows).
3. Run the specific installation command for your operating system, as follows:
   - For Windows, run: `wsadmin.bat -f installBRManager.jacl [-s` *servername* `-n` *nodename* `| -cl` *clustername*`] -ce` *cellname* `-r` *rootname*
   - For Linux, run: `wsadmin.sh -f ./installBRManager.jacl [-s` *servername* `-n` *nodename* `| -cl` *clustername*`] -ce` *cellname* `-r` *rootname*
   - For i5/OS, run: `wsadmin -f ./installBRManager.jacl [-s` *servername* `-n` *nodename* `| -cl` *clustername*`] -ce` *cellname* `-r` *rootname*

   To install and map the business rules manager to more than one target, run the following command for your operating system:
   - For Windows, run: `wsadmin.bat -f installBRManager.jacl -m "{{target1} {target2} ... {targetn}}" -ce` *cellname* `-r` *rootname*
   - For Linux, run: `wsadmin.sh -f installBRManager.jacl -m "{{target1} {target2} ... {targetn}}" -ce` *cellname* `-r` *rootname*
   - For i5/OS, run: `wsadmin -f installBRManager.jacl -m "{{target1} {target2} ... {targetn}}" -ce` *cellname* `-r` *rootname*
   - For z/OS®, run: `wsadmin.sh -f installBRManager.jacl -m "{{target1} {target2} ... {targetn}}" -ce` *cellname* `-r` *rootname*

   Note: The parameter "-m" (implying ″multiple″) allows you to install and map the business rules manager to many targets at the same time. A pair of double quotation marks encloses the targets.
   where:

   servername
   > The name of the application server.

   > The pair of arguments ″-s servername″ is required in the Network Deployment configuration if a cluster is not specified. If missing, the default value of *servername* is ″server1″.

nodename
> The name of the installation node.
>
> The pair of arguments "-n nodename" is required in the Network Deployment configuration if a cluster is not specified.

clustername
> The name of the cluster where you want to install the application.
>
> The pair of arguments "-cl clustername" is required in the Network Deployment configuration if a server name and a node name are not specified.
>
> **Note:** You must either specify the node and server or specify the cluster. Do not specify both.

cellname
> The name of the installation cell.
>
> The pair of arguments "-ce cellname" is optional.

rootname
> The name of the application root directory.
>
> The pair of arguments "-r rootname" is optional. If missing, the default value of *rootname* is "/br".

target*i* The target (where *i* is 1, 2, ..., n) to which you want to install and map the business rules manager.
> The target can be either (-s *servername* and -n *nodename*) or -cl *clustername*.

Important: If WebSphere Process Server is configured in a single-server environment, all of these pairs of arguments are optional. If WebSphere Process Server is configured for a Network Deployment environment, one of the following argument pairs is required:

- either (-s *servername* and -n *nodename*)
- or -cl *clustername*
- or -m "{{target1} {target2} ... {targetn}}"

The other argument pairs are optional.

Example: Suppose that you want to map the business rules manager application to the following targets:

- cluster "BofACluster"
- Web server "RedirectorServer" and node "AIXNode01"
- application server "LinuxServer" and node "LinuxNode02"

on context root "bofa/brm"

You would run the command, as follows:

```
install_root/bin/wsadmin -f installBRManager.jacl -m "{{-cl BofACluster} {-n
AIXNode01 -s RedirectorServer} {-s LinuxServer -n LinuxNode02}}" -r
bofa/brm
```

Installing the business rules manager using the AdminTask command:

With WebSphere Process Server 6.1 and higher, you can install the business rules manager using the Admin Task command. Similar to using the administrative console or the JACL command, use the Admin Task command if you did not install the business rules manager when you installed WebSphere Process Server and created profiles.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an administrator or a configurator to perform this task.

About this task

To install the business rules manager using the admin task command, perform the following steps.

Procedure
1. Ensure that WebSphere Process Server is started.
2. In a command window, go the WebSphere Process Server home directory and change to the directory /bin.
3. Run the wsadmin command to enter the wsadmin mode.
4. Enter one of the following commands to install the business rules manager:

   wsadmin> $AdminTask configBusinessRulesManager {*-serverName* <serverName> *-nodeName* <nodeName> *-contextRoot* <contextRoot>}

   Or

   wsadmin> $AdminTask configBusinessRulesManager {*-clusterName* <clusterName> *-contextRoot* <contextRoot>}

   where

   serverName
   > The name of the application server.

   nodeName
   > The name of the installation node.

   clusterName
   > The name of the cluster where you want to install the application.

   contextRoot
   > The context root used to launch the application. The default value is /br.
5. Run wsadmin> $AdminConfig save to save the configuration.

   Tip: You can run $AdminTask help configBusinessRulesManager to learn more about its parameters.

Example: To install the business rules manager on server "cvuServer" and node "cvuNode01" with context root "br", you should enter the following command:

wsadmin> $AdminTask configBusinessRulesManager {*-serverName* cvuServer *-nodeName* cvuNode01 *-contextRoot* br}

Then to save the configuration, enter:

wsadmin> $AdminConfig save

Configuring server security for the business rules manager:

If you want to use security with your server, you must configure the server that is using the business rules manager. On a server where security is not enabled, you can use the business rules manager without additional configuration.

About this task

If you have different roles or user IDs, you must set global security when configuring your server. To set security for your server, perform the following steps.

Procedure

1. Set global security on user IDs by assigning a role to each ID when creating the user IDs. Create each user ID and map each user ID to the role `BusinessRuleUser`.

   To set the role, navigate to the business rule manager application (**Applications > Enterprise Applications**), select the business rule manager application, select the Security role to user/group mapping and update the BusinessRuleUser role.

   Note: In an ND environment with global security turned on, if you plan to run the business rules manager on port 908n, where n is a positive integer, you should make sure that port "944(n+3) with the host value of "*" was configured. If there is no such port, manually configure it before you launch the business rules manager.
2. Set the session tracking mechanism to use cookies to track sessions.
3. At a minimum, set an appropriate session timeout value.

Configuring a Web browser for the business rules manager:

The server configures a client automatically while installing the business rules manager, but you must ensure that the Web browser is configured correctly for the business rules manager to work properly.

About this task

To ensure that the Web browser is configured correctly for the business rules manager, perform the following steps.

Procedure

1. Make sure that scripting is enabled in the Web browser.

   The business rules manager requires scripting to function.
2. Make sure that cookies are enabled.

   When necessary, cookies are used to track the session when you are using the business rules manager. Therefore, enable cookies on your browser when tracking sessions. Contact your system administrator if you enable cookies.

Accessing the business rules manager

You access the business rules manager using a Web browser.

Before you begin

Make sure that both the server and client are configured correctly.

About this task

The default URL for accessing the business rules manager is as follows. The URL may vary according to the environment.

http://hostname:port/br

where "hostname" is the name (or IP address) of the current host system, and "port" is the port of the application server where the application was installed.

For example, in the stand-alone environment with only one server, the link is the following:

http://hostname:9080/br

Note: If global security is enabled, the preceding link will automatically be switched to a secure link. For example, in the stand-alone environment with only one server, it is https://hostname:9443/br.

If global security is not enabled, the Business Rule Groups page opens. If global security is enabled on the server, the Login page opens.

If global security is enabled, perform the following steps to log in.

Procedure
1. At the Login page type your **User ID**.
2. Type your **Password**.
3. Click **Login**.

Results

The initial page of the business rules manager opens with the existing business rule groups listed in the navigation pane.

What to do next

You can now browse and edit business rule operations and templatize business rules.

Business Rule Groups page and the business rules manager page layout

When the business rules manager opens, the Business Rule Groups page displays, which allows you to browse all of the business rule groups and their defined operations.

The Business Rule Groups page is the first level of navigation. Its page layout includes many elements generic to the other business rules manager pages.

Toolbar

The toolbar contains the following components:

Welcome
      Displays the name of the user that is currently logged on.

User identification
> Provides the name of the current user preceded with **Welcome User Name**.

Logout
> Opens the Login page if global security is enabled.
>
> **Important:** If you log out without publishing, a dialog box appears asking for confirmation.

Search
> Opens the Search for Business Rule Groups page, which allows you to quickly locate or narrow a specified set of business rule groups that you want to work with.

Help Provides access to business rules topics in the WebSphere Process Server Information Center.

Navigation pane

The navigation pane is the left pane. It provides access to the Publish and Revert page and the available business rule groups. The navigation tree enables you to drill down to the rule level you need.

Note: The navigation pane is not displayed on any page that is in the edit mode.

Important: If you retrieve business rule artifacts with a version number greater than the version number of the current model, the business rule artifacts, known as shells, will become flat text items in the navigation pane. As a result, you will not be able to expose the shells further. You should update your current WebSphere Process Server to the latest one, which has a version equal to or higher than the version of the shells.

Publish and Revert
> Opens the Publish and Revert page where you can publish changes of business rule groups and rule schedules to the database or revert business rule groups or rule schedules to the original copy that was on the database.

Business Rule Groups
> Opens the Business Rule Groups page, which is the top level of browsing. The business rule groups are listed in a navigation tree. You can expand or collapse a business rule group by clicking either the plus (+) or minus (-) next to its display name to show all of its associated rules. When you select a business rule group in the left pane navigation tree, all the child Rule Schedule pages (business rule operations) are listed in the right pane, including all the associated rule sets and decision tables. Clicking any of these opens a corresponding page for editing.

Content area

The content area is the right pane and is the main viewing and editing area. The content area contains a title section, general information section, and page-specific section.

Note: The information displayed in the content area depends on whether you are viewing a Business Rule Group page, Rule Schedule page, Rule Set page, Decision Table page, Publish and Revert page, or Search for Business Rule Groups page.

Title section

The title section includes the following information:

Path information
> Provides the path to the page, such as the name of the business rule group and the Rule Schedule page in the following format:
>
> `BusinessRuleGroup01 > Table1_operation1`
>
> **Example:** `CalculateDiscountBRG > CalculateDiscount`

Rule title
> Provides the resource display name and type of business rule in the following format:
>
> `Ruleset112 - Ruleset`
>
> **Examples:** `calculateDiscount-Rule Schedule, CalculateDiscountRS - Rule Set`

Function buttons
> Enable various actions depending on the purpose of the particular page. Not all function buttons are available for a page, and some buttons appear in other sections of the content area. The following table lists the possible function buttons for a page.

Table 6. Function buttons

Button Name	Function
Add Property	Adds properties to a business rule group in the Business Rule Group page or to create a search query in the Search for Business Rule Groups page.
Back	Returns to the previous page.
Cancel	Discards any changes to the resource and returns to the previous page.
Copy	Copies either a decision table or rule set in order to create a new decision table or rule set. You must copy an existing decision table or rule set and then modify its values in order to make a new decision table or rule set.
Edit	Enables editing of the business rule group, rule schedule, rule set, or decision table.
Publish	Publishes the business rule group or rule schedule to the repository.
Revert	Cancels all changes to the rule that have been saved locally and reverts the rule to the original copy that resides in the server memory. Rules cannot be reverted after publishing.
Save	Validates and saves the changes to the local copy and goes back to the previous page. Note that the running state of the server has not been changed. See "Publish" for how to change the server's state.

Table 6. Function buttons (continued)

Button Name	Function
Search	Initiates the search query on the Search for Business Rule Groups page and returns the business rule groups that match the query as search results on the same Search for Business Rule Groups page.
Sort	Sorts the properties on the business rule groups by the property names in alphabetical ascending order.

Messages field
> Shows the status of an action that has been taken to the rule or that an error has occurred. The following are examples of status messages:
>
> `"calculateDiscount" has been temporarily saved.`
>
> `You may publish the changes from the "Publish and Revert" page.`

General Information section

The General Information section contains the following information.

Note: The Business Rule Group page includes the General Information section for WebSphere Process Server 6.1 and later. The Search for Business Rule Groups page and the Publish and Revert page do not have this section.

Display Name
> Gives the display name of the business rule group, rule set, or decision table for Websphere Process Server 6.1 and later. The display name is read-only in the browse mode but you can modify it in the edit mode on Business Rule Group, Rule Set, and Decision Table pages. Display names can be any string value and can include special characters. Display names of business rule artifacts of the same type do not need to be unique; however, the names of the business rule artifacts still need to be unique in use cases.
>
> If the display name is set, it is used instead of the name value everywhere name values are used, including the navigation pane and when artifacts are displayed in detail. If the display name of a business rule artifact is not set, its name value is used instead. Selecting the **Synchronize with the name** check box synchronizes the display name with the corresponding name value of the target business rule group, rule set, or decision table. The new name takes effect on all pages of the business rules manager when you save the changes made in the edit page.

Last Published
> Shows the last published date of the business rule group, rule schedule, rule set, or decision table.

Status Shows whether the rule schedule, rule set, or decision table is in the edit mode or has been published.

Description
> Provides a brief description of the business rule group, rule schedule, rule set, or decision table. You can edit the description in the edit mode of these pages.

Restriction: Do not use `CDATA` tags when editing the description fields for business rule group components and business rules in the business rules manager as they make business rule groups and business rules uneditable. If `CDATA` tags exist, open the business rule group or business rule with an XML editor and manually remove the `CDATA` tags from the description fields.

Page-specific information section

The content of the page-specific information section depends on whether you are viewing a Business Rule Group page, Rule Schedule page, Rule Set page, or Decision Table page. For specific information for each of these pages, see the individual topics.

For the Business Rule Groups page, the section includes the following information:

Business Rules Resources
Lists the display names of the rule schedules, rule sets, and decision tables.

Description
Provides either a brief description or the name of the resource.

Action
Shows the available actions for the corresponding business rule resource. It is initially empty; but when you expand the business rule group, an **Edit** button appears beside each rule.

Publish and Revert page:

The Publish and Revert page is for publishing locally saved changes for business rule groups and rule schedules to the repository. It is also for reverting business rule groups and rule schedules back to the original copy that was in the server memory before the business rule resource was saved locally.

The page-specific information section of the content area includes the following elements.

Changed Business Rules Resources section

This section provides a list of business rule groups and rule schedules available for publishing or reverting, with the following information:

Business Rule Resources
Lists the names of the changed business rule groups and rule schedules. Resources that are ready for publishing have a check box beside them to select or unselect for publishing.

Status Indicates if the resource is the original or has been changed locally.

Description
Provides a brief description of the resource.

Action
Indicates which resource can be reverted. The resource has a **Revert** button in the corresponding **Action** field.

Business Rule Group page:

The Business Rule Group page lists all the business rules resources associated with the business rule group.

You can browse this page or open the editing page for modifying the information for the business rule group or for the associated business rules resources, including adding, deleting, and modifying the custom properties of the business rule group.

The page-specific information section of the content area includes the following elements.

Properties section

This section provides the custom-defined properties for the business rule group.

Restriction: If the business rule group has no custom properties or its list of custom properties is empty, the Properties section will not display in the browse mode. Also, if the business rule group belongs to a version before WebSphere Process Server 6.1, the Properties section and **Edit** button for the business rule group will not display on the Business Rule Group page.

Name Specifies the name of the property. The name must be unique and cannot be empty. Each property can only be defined once in a business rule group.

Value Specifies the value of the property. Each property must have a defined value. It can be an empty string or zero in length, but not null. Setting a property to null is the same as deleting the property.

Business Rules Resources section

This section provides a list of rule schedules, rule sets, and decision tables associated with the business rule group.

Business Rules Resources
    Lists the display names of the rule schedules, rule sets, and decision tables associated with the business rule group.

Description
    Provides either a brief description or name of the business rule group, rule schedule, rule set, and decision table.

Action
    Shows the available actions for the corresponding business rule listing. It is initially empty; but when you expand the group, an **Edit** button appears beside each rule.

Rule Schedule page:

The Rule Schedule page provides an interface for modifying the values of a business rule group in the scheduled rule logic entries. The information is displayed in table format.

From the Rule Schedule page, you can perform such tasks as browsing, modifying, adding, splitting, or deleting effective dates for a business rule. You can also create a new business rule by copying an existing one.

The page-specific information section of the content area includes the following elements.

Scheduled Rule Logic section

This section provides a list of effective business rules that are the building blocks of that rule and enables working with scheduled rule logic entries, such as adding and sorting them.

Note: You can specify the rule logic selection **Date/Time** value in the business rules manager with either local time (uses the time zone of the client running the Web browser) or Universal Time Coordinated (UTC) time.

Start Date/Time
> Provides the options of either a specific date or "no start date."
>
> **Note:** The "no start date" signifies that the target rule logic is effective for any date before the end date.

End Date/Time
> Provides the option of either a specific date or "no end date."
>
> **Note:** The "no end date" signifies that the rule logic is effective for the start date and any date after it.

Effective Rule Logic
> Specifies the rule set or decision table that is effective in the corresponding time frame.

Action
> Provides options for splitting and deleting scheduled rule logic entries.

Default Rule Logic
> Provides a default rule logic if no other rule logic is applicable. It is selected when the date does not match any of the other scheduled rule logic entries.

Available Rule Logic section

This section provides a list of rule sets or decision tables that can apply to a particular business rule, with their associated descriptions and actions.

Rule Logic
> Specifies the name of the rule set or decision table.

Description
> Provides a brief description of the rule set or decision table.

Action
> Provides options to facilitate editing or copying rules.

Rule Set page:

The Rule Set page lists the rule "instances" for a business rule, their execution order, and associated templates for that rule set.

From the Rule Set page you can browse or edit an existing rule instance using the templates, create a new rule instance from a selected template, specify the execution order of the rules, rename a rule or rule set, browse or edit a rule set display name or rule in a rule set, browse or edit a rule set or rule description or description of a template parameter, save the rule set as a working copy, or delete a rule.

The page-specific information sections of the content area include the following elements.

Rules section

This section provides a list of associated rules with the following information:

Name Provides the name of the rule. This field is visible in edit mode only.

Display Name
Provides the display name of the rule. It is set to the **Name** value if a display name was not specified. It is read-only in the browse mode and editable in the edit mode. The display name can be any string value and can include special characters. It does not need to be unique. Selecting the **Synchronize Name** check box in the **Action** field synchronizes the display name with the corresponding name.

Rule Lists the variables, constraints, range, and enumeration that defines the rule.

Description
Provides more information about each rule in the rule set. It is read only in the browse mode and editable in the edit mode.

Action
Enables reordering rules, deleting rules, and synchronizing the display name with the name by clicking the associated buttons. The actions are available in the edit mode only.

Templates section

This section facilitates creating a new rule in the edit mode using an existing template and includes fields for specifying the following information for the rule:

Template Name
Provides the name of the existing template.

Name Provides a text area for entering and modifying the name of the rule.

Display Name
Provides a text area for entering the display name of the rule. It is set to the **Name** value if a display name is not specified. The display name can be any string value and can include special characters. It does not need to be unique. Selecting the **Synchronize Name** check box synchronizes the display name with the name value of the rule. The new name goes into effect on all pages of the business rules manager when you save the changes made in the edit page.

Note: If the **Synchronize Name** check box is selected, the display name of the rule is disabled and cannot be modified.

Rule Provides a text area for specifying the variables, constraints, range, and enumeration that defines the rule.

Description
Provides more information about each template parameter. It is visible only when a rule set is in the edit mode and you move the mouse over the target template parameter. It is read-only.

Action

Enables adding the rule to the template, deleting the rule from the template or synchronizing the display name with the name value of the rule.

Decision Table page:

The Decision Table page contains the condition cases and actions, their orientation (rows and columns), and the templates associated with that decision table. You open the Decision Table page from the Rule Schedule page.

From the Decision Table page, you can browse or edit an existing condition or action using a template, add a new condition using the templates defined for that decision table, delete a condition, change the order of conditions, change the orientation, change the initialization action rule using the associated template, browse and edit decision table and initialization rule display names and descriptions, and save a decision table as a working copy.

The page-specific information sections of the content area include the following elements.

Initialization Rule section

This section shows the initialization rule of the decision table. The initialization rule displays only if the business rule definition was designed in WebSphere Integration Developer with an initialization action. The initialization rule is invoked directly before the decision table logic is issued and can be used to initialize variables and actions used in the decision table. In the edit mode there are fields for modifying the following information.

Name Provides the name of the initialization rule.

Display Name

Provides the display name of the rule. It is set to the **Name** value if a display name was not specified. The display name can be any string value, can include special characters, and does not need to be unique. Selecting the **Synchronize Name** check box in the **Action** field synchronizes the display name with the corresponding name. The new name goes into effect when you save the changes made in the edit page.

Note: If the **Synchronize Name** check box is selected, the display name of the rule is disabled and cannot be modified.

Rule Lists the variables, constraints, range, and enumeration that defines the initialization rule.

Description

Provides more information about each initialization rule. It is read-only in the browse mode and editable in the edit mode of the decision table.

Action

Enables synchronizing the display name with the name by selecting the **Synchronize Name** check box.

Decision Table section

This section provides the conditional cases, represented in the row and column headings, and the actions, represented as the intersection points of the conditional

cases in the table. You can switch the orientation of condition rows from horizontal to vertical, or vice versa, using the **orientation** icon.

Otherwise

Shows the *otherwise* condition of this decision table. The *otherwise* condition is a special condition that will be entered by default if no other condition in the decision table is applicable. The *otherwise* condition displays only if it was specified in the decision table definition that was designed in WebSphere Integration Developer. You cannot add or remove the *otherwise* condition column from a decision table dynamically from the business rules manager.

Templates section

This section facilitates adding a new rule using an existing template.

Search for Business Rule Groups page:

The Search for Business Rule Groups page is for creating a search query to locate or narrow a specified set of business rules groups that you want to work with. You open the Search for Business Rule Groups page by clicking **Search** in the toolbar at the top of the business rules manager.

On the Search for Business Rule Groups page, you can search by the target namespace, business rule group name, custom properties or any combination of these; you can add one or many custom properties, sort custom properties by their names in alphabetical ascending order, move properties up or down inside the property table, or delete custom properties.

The content area of the Search for Business Rule Groups page includes a **Messages** field and page-specific information sections with the following elements.

Search Data section

This section contains the following elements:

Name Provides a text area for entering the name of the business rule group to search for. If you leave this value empty, it will not be included in the search context. The value you enter is used as both a name and a display name. Consequently, the search will look for business rule groups with either the names or the display names that match the entered name value. If you want to specifically search by either name or display name, but not both, you need to indicate such a search through property names.

Example: If you enter IBMSystemName for the name of a property and VIPGroup for the value of the property, the business rules manager will search for business rule groups with the names, but not display names, matching VIPGroup.

Target Namespace

Provides a text area for entering the URL of the business rule group. If you leave this value empty, it will not be included in the search context.

Properties section

This section opens when you click **Add Property** and contains the following elements:

Logical Operator

Provides a drop-down list for selecting ″And″, ″Or″, or ″Not″ to create a search query containing multiple properties.

Name Provides a text area for entering the name of the property. The name must be unique inside the Properties table of the search context and must not be empty.

Query Operator

Provides a drop-down list for selecting from four query operators for each search data field. The query operators are as follows.

Query Operator	Description
is equal to	Indicates that the value of a business rule group name, target name space, or property must match the specified string exactly.
is like	Indicates that the query should look for business rule groups where the value of a business rule group name, target name space, or property is like the specified string. The string can contain wildcard characters. Use the percent character ('%') to specify a wildcard for any number of characters and use the underscore character ('_') to specify a single character wildcard. These wildcard characters must follow SQL syntax.
is not equal to	Indicates that the value of the business rule group name, target name space, or property must not match the specified string.
is not like	Indicates that the query should look for business rule groups where the value of a business rule group name, target name space, or property is not like the specified string. The string can contain wildcard characters as defined in the ″like″ operator.

Value Provides a text area for entering the property value. The value can be empty and is taken into the Search context.

Example: If the value of property PayMethod is left empty and its query operator is set to ″is not equal to,″ the Search will find all the business rule groups whose PayMethod property has the value set to a non-empty string.

Action

Enables moving a property up or down inside the property table and deleting custom properties.

Search Results section

This section contains the following elements:

Rule Groups

Lists the names of the business rule groups that the search query returned.

Status Shows the status of the business rule group returned from the runtime as a search result. The status can be one of the following four kinds of status.

Tip: Clicking on a result business rule group opens its business rule group page.

Status	Description
Same as Local	Indicates that a copy of the result business rule group already exists in the business rule manager and that its content and the content of the result business rule group are exactly the same. Thus, no further action is taken after the search.
Modified from Runtime	Indicates that a copy of the result business rule group already exists in the business rules manager. However, another user session modified the master copy, and so the contents of the local and result business rule groups are different. The business rules manager will automatically update the local copy to get new modifications from the runtime.
Modified in Local	Indicates that a copy of the result business rule group already exists in the business rules manager. However, it has been modified by the current user. The business rules manager will use the local copy for any further actions by the user.
New to Local	Indicates that a copy of the result business rule group does not exist in the business rules manager. In this case, the business rules manager will create a local copy of the result business rule group and will also display it in the navigation pane.

Description

Provides additional information for the business rule group.

Adding, deleting, and modifying business rule group properties

You can use custom properties on business rule groups for searches in order to retrieve subsets of business rule groups that you want to view and modify. You add new custom properties, delete or modify existing properties through the editing pages of business rule groups. The number of custom properties on a business rule group is unlimited.

Before you begin

You need to be in the edit mode for the business rule group.

Restriction: Properties support on business rule groups is available on 6.1 business rule groups and later.

About this task

To add, delete, or modify business rule group properties, perform the following steps.

Procedure

1. Select from the following options.

Option	Description
Option	**Steps**
Add a property to the rule	1. Click **Add Property**. 2. Specify a unique **Name**. The name cannot be empty. 3. Specify a unique **Value**. Each property can only be defined once in a business rule group and must have a defined value. The value can be an empty string or zero in length, but not null. Setting a property to null is the same as deleting the property.
Delete a property	In the **Action** field of the selected property, click **Delete**.
Modify a property	Enter the new name and value in the corresponding field.
Sort properties	Click **Sort** to sort the properties on the business rule groups by the property names in alphabetical ascending order.

2. Click **Save**.

Results

The business rules manager will validate the rules before sending the properties to the server.

Searching business rule groups

You can perform a search query on a business rule group to locate or narrow a specified set of business rule groups that you want to work with. You create a search query based on the name, target name space, custom properties, or any combination of these.

Before you begin

You need to be on the Search for Business Rule Groups page, which you can open by clicking **Search** in the business rules manager toolbar.

About this task

To create a search query, perform the following steps.

Procedure

1. In the **Name** field enter the name of the business rule group to search for. You can leave this value empty; however, it will not be included in the search context. The value you enter is used as both a name and a display name. Consequently, the search will look for business rule groups with either the names or the display names that match the entered name value. If you want to specifically search by either name or display name, but not both, you need to indicate such a search through property names.

Example: If you enter IBMSystemName for the name of a property and VIPGroup for the value of the property, the business rules manager will search for business rule groups with the names, but not display names, matching VIPGroup.

2. In the **Target Namespace** field enter the URL of the business rule group. You can leave this value empty; however, it will not be included in the search context.

3. For each **Search Data** field select one of the following four query operators.

Option	Description
Query Operator	Description
is equal to	Indicates that the value of a business rule group name, target name space, or property must match the specified string exactly.
is like	Indicates that the query should look for business rule groups where the value of a business rule group name, target name space, or property is like the specified string. The string can contain wildcard characters. Use the percent character ('%') to specify a wildcard for any number of characters and use the underscore character ('_') to specify a single character wildcard. These wildcard characters must follow SQL syntax. **Examples:** 1. If you enter "is like" "Discount" for the business rule group name and "http://calculateDiscounts" as the target name space, the search will return all the business rule groups containing that string and with that URL. 2. If you enter "is like" "%Discount%" for the business rule group name, the search will return all the business rule groups with names such as AirlineTicketDiscount and MovieTicketDiscountRules.
is not equal to	Indicates that the value of the business rule group name, target name space, or property must not match the specified string.
is not like	Indicates that the query should look for business rule groups where the value of a business rule group name, target name space, or property is not like the specified string. The string can contain wildcard characters as defined in the "like" operator.

4. **Optional:** Click **Add Property** to add as many properties as needed for the search context.

   a. Specify the **Name**. It must be unique inside the Properties table of the search context and must not be empty.

   b. Specify the **Query Operator**.

   c. Specify the **Value**. The value can be empty and is taken into the search context.

Example: If the value of property PayMethod is left empty and its query operator is set to "is not equal to," the search will find all the business rule groups whose PayMethod property has the value set to a non-empty string.

   d. Click the up and down arrows in the **Action** field to order the properties.

Tip: You can combine the properties in the **Logical Operator** field using "And", "Or", or "Not" to create a search query containing multiple properties.

Example: To search for all the business rule groups in target namespace "http://calculateDiscounts" and with the DiscountedItem property containing string "men T-Shirts" and with the Ship Handling property set to value "Free", you would use the logical property "And".

Note: Adding, deleting, or modifying the properties on the Search for Business Rule Groups page only applies within the search context. It does not affect the properties of any rule object inside the business rules manager.

5. Click **Search**.

Results

The business rule groups that match the search query display in the **Search Results** section on the Search for Business Rule Groups page. The status of the business rule group returned from the runtime as a search result may be one of the following four kinds of status.

Status	Description
Same as Local	Indicates that a copy of the result business rule group already exists in the business rule manager and that its content and the content of the result business rule group are exactly similar. Consequently, no further action is taken after the search.
Modified from Runtime	Indicates that a copy of the result business rule group already exists in the business rules manager. However, another user session modified the master copy, and so the contents of the local and result business rule groups are different. The business rules manager will automatically update the local copy to get new modifications from the runtime.
Modified in Local	Indicates that a copy of the result business rule group already exists in the business rules manager. However, it has been modified by the current user. The business rules manager will use the local copy for any further actions by the user.
New to Local	Indicates that a copy of the result business rule group does not exist in the business rules manager. In this case, the business rules manager will create a local copy of the result business rule group and will also display it in the navigation pane, too.

Note: The synchronization of changes of the business rule groups occurs at the same time as the search results returned and is applied in the business rules manager context. This means that the next operation on an affected business rule group will work with the latest updates of the business rule group.

Examples: Four business rule groups are installed with the following properties:

Business rule group 1
- **Name:** BRDCR002BRG2.brg
- **Target namespace:** http://BRDCR002BRG2/com/ibm/br/rulegroup
- **Properties:**
  - organization, 7GAA
  - department, accounting
  - ID, 0000047
  - ID_cert45, ABC
  - region, NorthRegion

Business rule group 2
- **Name:** BRDCR002BRG3.brg
- **Target namespace:** http://BRDCR002BRG3/com/ibm/br/rulegroup
- **Properties:**
  - organization, 7FAB
  - department, finance
  - ID, 0000053
  - ID_app45, DEF
  - region, NorthCentralRegion

Business rule group 3
- **Name:** BRDCR002BRG4.brg
- **Target namespace:** http://BRDCR002BRG4/com/ibm/br/rulegroup
- **Properties:**
  - organization, 7HAA
  - department, shipping
  - ID, 0000023
  - ID_app45, GHI
  - region, SouthRegion

Business rule group 4
- **Name:** BRDCR002BRG5.brg
- **Target namespace:** http://BRDCR002BRG5/com/ibm/br/rulegroup
- **Properties:**
  - organization, 8JAA
  - department, claims
  - ID, 00000567
  - region, SouthCentralRegion
  - manager, Joe Bean

Retrieve a business rule group by a single property.

Logical Operator	Name	Query Operator	Value
	department	is equal to	accounting

This returns business rule group 1.

Retrieve business rule groups by two properties using the ′%′ multi-character wildcard.

Logical Operator	Name	Query Operator	Value
	region	is like	%Region
AND	ID	is like	00000%

This returns business rule groups 1-4.

Retrieve business rule groups by using the ′_′ single-character wildcard.

Logical Operator	Name	Query Operator	Value
	ID	is like	00000_3

This returns business rule groups 2 and 3.

Retrieve business rule groups by using multiple ′_′ single-character wildcard.

Logical Operator	Name	Query Operator	Value
	region	is like	__uth%Region

This returns business rule groups 3 and 4.

Retrieve a business rule group by using a ′_′ single-character wildcard and not operator.

Logical Operator	Name	Query Operator	Value
	organization	is not like	7__A

This returns business rule group 4.

Retrieve a business rule group by using a ′%′ multi-character wildcard and not operator.

Logical Operator	Name	Query Operator	Value
	organization	is not like	7%

This returns business rule group 4.

What to do next

Click a result business rule group to open its business rule group page.

Working with scheduled rule logic entries

A scheduled rule logic entry identifies information for a rule, such as its effective dates and the if/then rule set or decision table associated with the rule.

About this task

Use the business rules manager to create, modify, or delete scheduled rule logic entries.

Creating scheduled rule logic entries:

You create scheduled rule logic entries from existing entries.

Before you begin

You need to be in the edit mode for the rule you want to create.

About this task

To create a new scheduled rule logic entry, perform the following steps.

Procedure
1. On the Rule Schedule page click **Add Selection Record**.

   A new scheduled rule logic entry is added at the bottom of the list with the **Start Date/Time** and **End Date/Time** fields set to **Jan 1**. A message displays in the **Messages** field indicating that the date/time field values are invalid.
2. Set the **Start Date/Time** field:
   a. Select the month from the drop-down list.
   b. Select the day from the drop-down list.
   c. Enter the year.
   d. Enter the time (in 24-hour format).
3. Set the **End Date/Time** field.
   a. Select the month from the drop-down list.
   b. Select the day from the drop-down list.
   c. Enter the year.
   d. Enter the time (in 24-hour format).

   Restriction: Only one rule logic can be in effect at any one point in time. Rule dates cannot have date/time ranges that overlap.

   Note: Gaps in date/time ranges are allowed. If you have specified a default rule logic, it is used during the gap. As a best practice, you should always specify a default rule logic.
4. Select the **Effective Rule Logic** from the drop-down list.
5. Click **Save** .

Results

A message displays in the **Messages** field indicating that the scheduled rule logic entry has been temporarily saved and that you can publish the changes from the Publish and Revert page.

Related tasks

"Deleting scheduled rule logic entries" on page 140
You can delete existing scheduled rule logic entries from the scheduled rule logic table. When a scheduled rule logic entry is deleted, the associated rule set or decision table definition remains with the rule group and is listed in the Available Rule Logic section of the page. The scheduled rule logic entry can be added back either as the default rule logic or with a specific date and time.

Modifying scheduled rule logic entries:

You can modify the date and time values of existing scheduled rule logic entries.

Before you begin

You need to be in the edit mode for the rule you want to modify.

About this task

To modify a scheduled rule logic entry, perform the following steps.

Procedure
1. On the Rule Schedule page edit the **Start Date/Time** of the scheduled rule logic entry:
   a. Select the month from the drop-down list.
   b. Select the day from the drop-down list.
   c. Enter the year.
   d. Enter the time (in 24-hour format).
2. Edit the **End Date/Time** of the scheduled rule logic entry:
   a. Select the month from the drop-down list.
   b. Select the day from the drop-down list.
   c. Enter the year.
   d. Enter the time (in 24-hour format).

   Restriction: Only one rule logic can be in effect at any one point in time. Rule dates cannot have date/time ranges that overlap.

   Note: Gaps in date/time ranges are allowed. If you have specified a default rule logic, it is used during the gap. As a best practice, you should always specify a default rule logic.
3. Click **Save**.

   Note: If the **Date/Time** fields are invalid, the fields will turn **red** and a message will display in the **Messages** field indicating that the dates/time field values are invalid.

Results

The scheduled rule logic entry is saved locally and is ready to be published to the repository. For more information, see "Publishing and reverting business rules" on page 141.

What to do next

For more information on setting business rule dates, see "Splitting dates in business rules" on page 132.

   Related tasks

   "Deleting scheduled rule logic entries" on page 140
   You can delete existing scheduled rule logic entries from the scheduled rule logic table. When a scheduled rule logic entry is deleted, the associated rule set or decision table definition remains with the rule group and is listed in the

Available Rule Logic section of the page. The scheduled rule logic entry can be added back either as the default rule logic or with a specific date and time.

Date/Time selections:

Business rules are selected by a date/time specification.

The date is defined either as part of the business rule group's operation parameter or it is derived at run time. The dates are always in terms of UTC and are specific points in time. Only one rule logic can be effective for an operation at any point in time. When no rule logic is found to be in effect for any point in time, the default rule logic is used.

The business rule group supports the following date/time options, which you access by clicking the icon in the **Start Date/Time** and **End Date/Time** fields:

Specify Date/Time
Specifies a date manually.

Continuous
Uses an automatic date calculation that sets the end date to the earliest start date that is later than the scheduled rule logic entry. The continuous date selection is only available on the **End Date/Time** field.

Note: The continuous selection is used when date ranges of two scheduled rule logic entries are contiguous. A continuous attribute is set to the end date of the first scheduled rule logic entry. When this attribute is set, the start date of the second scheduled rule logic entry is set to the end date of the first scheduled rule logic entry so that you do not have to specify both dates.

No Start Date or No End Date
Does not set a starting or ending boundary, depending on which is selected.

Restriction: The business rule group only supports effective dates. If you need to perform another type of selection, use a selector component.

Splitting dates in business rules:

Splitting a date in a business rule provides a shortcut for modifying a business rule for another purpose.

Before you begin

You need to be in the edit mode for the rule you want to modify.

About this task

To split a scheduled rule logic entry, perform the following steps.

Procedure

1. Click **Split** next to the scheduled rule logic entry.

   A new scheduled rule logic entry is created with a start date of Jan 1; and its fields are in red. A message displays in the **Messages** field indicating that the date/time field values are invalid.

2. Select the start date/time for the new scheduled rule logic entry.

The end date/time for the original scheduled rule logic entry changes from *continuous* to the start date/time of the new scheduled rule logic entry, and the end date/time of the new scheduled rule logic entry changes to the end date/time of the previous scheduled rule logic entry.

3. Modify the date/times of the new scheduled rule logic entry.
4. Modify the **Effective Rule Logic** to fit the needs of the new rule.

Rule sets

A rule set is a group of if/then statements or rules where the *if* is the condition and the *then* is the action of the rule. Rule sets are best suited for those business rules that have very few condition clauses.

If the condition is met, the action is performed. This may result in more than one action being performed by the rule set. The order of rule processing is determined by the order of the rule statements in the if/then rule set. Therefore, when you modify or add a rule, you need to be sure that it is in the correct sequence.

A rule set may have two kinds of rules–if/then rules and action rules:
- An if/then rule determines what action to take according to the condition of the incoming message.
- An action rule determines what action to take no matter what the incoming message is.

A condition in a rule contains a condition expression, which could be a simple string or an *and*, *or*, or *not*.

You create new rule sets or modify existing rule sets in the business rules manager using templates defined for that rule set. The templates provide the structure that determines how the rule set functions. Rule templates are not shared between rule sets.

Creating rule set entries:

You create a new rule set entry by copying an existing rule set entry and modifying its values.

About this task

To create a new rule set entry, perform the following steps.

Procedure

1. Click **Copy** next to the scheduled rule logic entry for the selected rule set.
   The edit page opens for the new entry, with a title Edit Mode:Copy_of_TableName-Ruleset.
2. In the **Name** field enter a unique name for the new rule set entry.
3. In the **Display Name** field enter a display name for the new rule set entry. The display name does not need to be unique for the rule set. It can be any string value and can contain special characters. If you do not specify a display name, the **Name** value will be used for the display name.

   Note: To synchronize the display name with the corresponding name of the rule set, select the **Synchronize with the name** check box.
4. In the **Description** field enter a short description for the new rule set entry.
5. Modify the values in each condition.

Tip: To display the parameter settings for each value, place your cursor over a field. A rollover message shows the type of variable and its range.

6. Click the up or down arrow to place the rule in the correct sequence.
7. Click **Save**.

Results

A message displays in the **Messages** field indicating that the rule set entry has been temporarily saved and that you can publish the changes from the Publish and Revert page.

Related tasks

"Deleting scheduled rule logic entries" on page 140
You can delete existing scheduled rule logic entries from the scheduled rule logic table. When a scheduled rule logic entry is deleted, the associated rule set or decision table definition remains with the rule group and is listed in the Available Rule Logic section of the page. The scheduled rule logic entry can be added back either as the default rule logic or with a specific date and time.

Creating rules within rule sets from templates:

You create a new rule within a rule set using the rule templates associated with that rule set.

Before you begin

You need to be in the edit mode for the rule set.

About this task

To create a new rule from a template, perform the following steps.

Procedure

1. Click **New Rule from Template** to display the list of available templates for the rule.
2. Select a template and do the following:
   a. In the **Name** field enter the name of the new rule.
   b. In the **Display Name** field enter a display name for the new rule. The display name does not need to be unique for the rule. It can be any string value and can contain special characters. If you do not specify a display name, the **Name** value will be used for the display name.

      Note: To synchronize the display name with the name value, select the corresponding **Synchronize Name** check box in the **Action** field. If the check box is selected, the display name of the rule is disabled and cannot be modified.
   c. Specify the values for the rule in the input fields or select the variables from the drop-down lists.
   d. Enter a description for the rule.
   e. Click **Add**.
3. Click the up or down arrows in the **Action** field to place the rule in the proper order.

Note: The order of rule processing is determined by the order of the rule statements in the if/then rule set. Therefore, when you modify or add a rule, you need to be sure that it is in the correct sequence.

4. Click **Save**.

What to do next

The rule set is ready for publishing. For more information, see "Publishing and reverting business rules" on page 141.

> **Related tasks**
>
> "Deleting scheduled rule logic entries" on page 140
> You can delete existing scheduled rule logic entries from the scheduled rule logic table. When a scheduled rule logic entry is deleted, the associated rule set or decision table definition remains with the rule group and is listed in the Available Rule Logic section of the page. The scheduled rule logic entry can be added back either as the default rule logic or with a specific date and time.

Modifying rules within rule sets using templates:

You modify a rule in a rule set using templates associated with that rule set.

Before you begin

You need to be in the edit mode for the rule set.

About this task

To modify a rule using an existing template, perform the following steps.

Procedure

1. Edit the value by typing over the existing value in the input field or by clicking the down arrow that appears in the field and selecting a value from the drop-down list.
2. If necessary, click the up or down arrows to place the rule in the proper order.

   Note: The order of rule processing is determined by the order of the rule statements in the if/then rule set. Therefore, when you modify or add a rule, you need to be sure that it is in the correct sequence.
3. Click **Save**.

What to do next

The modified rule set is ready for publishing. For more information, see "Publishing and reverting business rules" on page 141

> **Related tasks**
>
> "Deleting scheduled rule logic entries" on page 140
> You can delete existing scheduled rule logic entries from the scheduled rule logic table. When a scheduled rule logic entry is deleted, the associated rule set or decision table definition remains with the rule group and is listed in the Available Rule Logic section of the page. The scheduled rule logic entry can be added back either as the default rule logic or with a specific date and time.

Decision tables

A decision table is a scheduled rule logic entry, in table format, that consists of conditions, represented in the row and column headings, and actions, represented as the intersection points of the conditional cases in the table. Decision tables are best suited for business rules that have multiple conditions. Adding another condition is done by simply adding another row or column.

Like the if/then rule set, the decision table is driven by the interaction of conditions and actions. The main difference is that in a decision table, the action is decided by more than one condition, and more than one action can be associated with each set of conditions. If the conditions are met, then the corresponding action or actions are performed.

Templates

You use templates to modify decision table values in the business rules manager. The templates are designed in WebSphere Integration Developer and contained in the business rule definition. The templates determine which aspects of a decision table you can modify and provide a list of valid values to choose from. You create new rows or columns in the table or new actions based on the templates defined for that decision table, and you modify existing conditions or actions that were created with the template. Decision table templates are not shared between decision tables.

Initialization action rules

Decision tables support the use of an initialization action rule, which runs before the decision table is executed and allows for pre-processing, such as for creating business objects or setting initial values. You can modify an initialization action rule in the business rules manager, provided that the business rule definition was designed in WebSphere Integration Developer with an initialization action.

Although only one initialization action rule can be created from a single template, the action rule can have multiple action expressions in it, so it can perform multiple actions. If an initialization rule template is defined for a particular decision table, it can only be used in that table.

Otherwise conditions

The *otherwise* condition is a special condition that will be entered by default if no other condition in the decision table is applicable.

The *otherwise* condition will only display in the business rules manager if it is included in the decision table definition that was designed in WebSphere Integration Developer. You cannot add or remove it dynamically in the business rules manager.

However, you can incorporate actions defined with templates for the *otherwise* condition. The *otherwise* condition can be used zero or one time for any condition being checked.

The following figure shows a decision table with an *initialization action rule* that sets the default member type to Silver) and *otherwise conditions* that apply to gold and silver customers spending less than $500. The *conditions* PurchaseAmount and MemberType are along the first and second rows, and the *action* Discount is along

the third row. The orientation of conditions and actions is shown by arrows.

Initialization Rule

Display Name	Rule	Description
Rule1	Default Member Type = Silver	

Decision Table

PurchaseAmount		>= 500 && < 2000		>= 2000		Otherwise	
MemberType		Gold	Silver	Gold	Silver	Gold	Silver
Discount		8 %	3 %	10 %	5 %	2 %	0 %

Figure 6. Decision table

The example shows that gold customers spending $500 - $1999 get an 8% discount while silver customers spending $500 - $2000 get a 3% discount. Gold customers spending $2000 or more get a 10% discount while silver customers spending $2000 or more get a 5% discount. Gold customers spending less than $500 get a 2% discount while silver customers spending less than $500 get a 0% discount.

Creating decision table entries:

You create a new decision table entry by copying an existing decision table entry and modifying its values.

About this task

To create a decision table entry, perform the following steps.

Procedure

1. Click **Copy** next to the scheduled rule logic entry for the selected decision table.

    The edit page opens for the new entry, with a title Edit Mode:Copy_of_TableName-Decision Table.

2. In the **Name** field enter a name for the new decision table entry.

3. In the **Display Name** field enter a display name for the new decision table entry. The display name does not need to be unique for the decision table. It can be any string value and can contain special characters. If you do not specify a display name, the **Name** value will be used for the display name.

    Note: To synchronize the display name with the name value, select the corresponding **Synchronize with the name** check box.

4. In the **Description** field enter a short description of the new decision table entry.

5. Modify the **values** in each condition.

    Tip: To display the parameter settings for each value, place your cursor over a field. A rollover message displays showing the type of variable and its range.

6. Click **Save**.

Results

A message appears in the message field indicating that the decision table entry has been temporarily saved and that you can publish the changes from the Publish and Revert page. For more information, see "Publishing and reverting business rules" on page 141.

Related tasks

"Deleting scheduled rule logic entries" on page 140
You can delete existing scheduled rule logic entries from the scheduled rule logic table. When a scheduled rule logic entry is deleted, the associated rule set or decision table definition remains with the rule group and is listed in the Available Rule Logic section of the page. The scheduled rule logic entry can be added back either as the default rule logic or with a specific date and time.

Special actions menu:

The Decision Table page has a **Special actions** menu to edit the values in a decision table or modify the structure and variables of a template.

The **Special actions** menu is available for any field that has the **Special actions** icon beside it when a decision table is in the edit mode. Clicking the **Special actions** icon for the field opens a list of available options for the field. The following table lists the possible options.

Note: Reordering the columns or rows only affects the visual presentation of the table and has no effect on the order in which the conditions and actions are processed.

Menu option	Description	Modifies condition	Modifies action
Add below	Adds a new condition value (row) below the present cell (orientation is vertical)	Yes	
Add to the right	Adds a new condition value to the right of the cell (orientation is horizontal)	Yes	
Change template	Allows modifications to the cell value	Yes	Yes
Move up	Moves the condition value or variable to the row above (orientation is vertical)	Yes	
Move down	Moves the condition value or variable to the row below (orientation is horizontal)	Yes	
Move left	Moves the condition value or variable to the left (orientation is horizontal)	Yes	

Menu option	Description	Modifies condition	Modifies action
Move right	Moves the condition value or variable to the right (orientation is vertical)	Yes	
Delete	Deletes the condition value or variable	Yes	
Close menu	Closes the menu	Yes	Yes

Modifying decision table entries:

You edit a decision table by directly entering the new value into the appropriate input field or by selecting a value from the field's list box options.

Before you begin

You need to be in the edit mode for the decision table you want to modify.

About this task

To modify the values of a decision table, perform the following steps.

Procedure
1. Edit the value by typing over the existing value in the input field or by clicking the down arrow that appears in the field and selecting a value from the drop-down list.

   Restriction:
   - The initialization rule will only be displayed in the decision table if it is included in the business rule definition that was designed in WebSphere Integration Developer. Only one initialization action rule can be associated with a single template, but the action rule can have multiple action expressions in it.
   - The *otherwise* condition will only be displayed in the decision table if it is included in the business rule definition that was designed in WebSphere Integration Developer. You cannot add or remove the *otherwise* condition in the business rules manager; however, you can incorporate actions defined with templates for the *otherwise* condition.
2. Click the **Special actions** icon beside the field to open a list of available options for the field, and select an action, as desired.

   Note: Selecting an option for reordering the columns or rows only affects the visual presentation of the table and has no effect on the order in which the conditions and actions are processed.
3. Click **Save**.

Results

The rule is modified locally and is ready to be published to the repository. For more information, see "Publishing and reverting business rules" on page 141.
   Related tasks

"Deleting scheduled rule logic entries"
You can delete existing scheduled rule logic entries from the scheduled rule
logic table. When a scheduled rule logic entry is deleted, the associated rule set
or decision table definition remains with the rule group and is listed in the
Available Rule Logic section of the page. The scheduled rule logic entry can be
added back either as the default rule logic or with a specific date and time.

Modifying template values of decision tables:

You modify the structure and values of a decision table template by using the
Special actions menu and by directly entering values into the appropriate input
fields.

Before you begin

You need to be in the edit mode for the decision table you want to modify.

About this task

To modify a decision table template, perform the following steps.

Procedure
1. Click the **Special action** icon located beside the decision table field you want to
   modify to open the list box of options for the field, and select **Change
   Template**.
2. Type the new value for the template over the existing value in the input field.
3. Click **Change** in the **Action** column.
4. Click **Save**.

Results

The decision table template has been modified and is now ready for publishing.
For more information, see "Publishing and reverting business rules" on page 141.

Deleting scheduled rule logic entries
You can delete existing scheduled rule logic entries from the scheduled rule logic
table. When a scheduled rule logic entry is deleted, the associated rule set or
decision table definition remains with the rule group and is listed in the Available
Rule Logic section of the page. The scheduled rule logic entry can be added back
either as the default rule logic or with a specific date and time.

Before you begin

You need to be in the edit mode for the rule you want to delete.

About this task

To delete a scheduled rule logic, perform the following steps.

Procedure
1. On the Rule Schedule page select the scheduled rule logic, and click **Delete**.
   The scheduled rule logic is deleted. The associated rule set or decision table
   definition remains with the rule group and is listed in the Available Rule Logic
   section of the page.

Note: Each operation on a business rule group must have at least one active business rule associated with it, either as a scheduled rule logic entry or as a default rule logic. Attempting to delete all scheduled rule logic entries will result in an error.

2. Click **Save**.

Results

The scheduled rule logic entry is temporarily saved and is ready for publishing to the repository.

Publishing and reverting business rules

When you save any part of a business rule group, the changes are saved locally. In order to store the changes to the data source that resides on the application server, you need to *publish* the changes. Alternatively, you can cancel the changes that have been saved locally to a business rule by *reverting* the rule to its original state.

Before you begin

You need to be on any business rules manager page that has a navigation pane.

About this task

The server publishes changes at the business rule group and rule schedule levels. At the publishing stage, the business rules manager does not need to do any validations because the business rules manager validates all changes you enter on each edit page when you save the information.

To publish the changes to a business rule group or rule schedule, perform the following steps.

Procedure

1. Click **Publish and Revert**.
2. On the Publish and Revert page select the business rule groups and rule schedules to send to the repository by clicking their check boxes in the left-hand column of the content area. You can publish all the business rule groups and rule schedules together as a single transaction, or just a subset of them.

   Note: To cancel all changes that have been saved locally to a business rule group or rule schedule and replace the changed resource with the original copy in the server memory, select the check box for the business rule group or rule schedule and click **Revert**. Business rule groups and rule schedules cannot be reverted after publishing since publishing changes the original copy that resides in the server memory.

3. Click **Publish**.

   The selected business rule groups and rule schedules are written to the server memory.

What to do next

The business rule is ready to be exported to the data source.

Troubleshooting the business rules manager

Some of the problems you may encounter using the business rules manager are login errors, login conflicts, and access conflicts.

You can take various steps to troubleshoot these problems.

Resolving login errors:

A log in error occurs upon logging in.

About this task

The login error message is as follows:

```
Unable to process login. Please check User ID and password and try again.
```

Note: Login errors occur only when global security is enabled and either the user ID, password, or both, are incorrect.

To resolve login errors, perform the following steps.

Procedure
1. Click **OK** on the error message to return to the Login page.
2. Enter the valid **User ID** and **Password**.
   - If passwords are case sensitive, make sure that Caps Lock key is not on.
   - Make sure the user ID and password are spelled correctly.
   - Check with the system administrator to be sure that the user ID and password are correct.
3. Click **Login**.

What to do next

If you resolve the login error, you will now be able to log in to the business rules manager. If the error is not resolved, contact your system administrator.

Resolving login conflict errors:

A login conflict error occurs when another user with the same user ID is already logged in to the application.

About this task

The login conflict message is as follows:

```
Another user is currently logged in with the same User ID. Select from the following options:
```

Usually this error occurs when a user closed the browser without logging out. When this condition occurs, the next attempted login before the session timeout expires results in a login conflict.

Note: Login conflict errors occur only when global security is enabled.

To resolve login conflict errors, select from the following three options:

- Return to the Login page.

  Use this option if you want to open the application with a different user ID.
- Log out the other user with the same user ID.

  Use this option to log out the other user and start a new session.

  Note: Any unpublished local changes made in the other session will be lost.
- Inherit the context of the other user with the same user ID and log out that user.

  Use this option to continue work already in progress. All unpublished local changes in the previous session that have been saved will not be lost. The business rules manager will open to the last page displayed in the previous session.

Resolving access conflict errors:

An access conflict error occurs when a business rule is updated in the data source by one user at the same time another user is updating the same rule.

Before you begin

This error is reported when you publish your local changes to the repository.

About this task

To correct access conflict errors, perform the following actions:
- Find the source of the business rule that is causing the error and check if your changes on the local machine are still valid. Your change may no longer be required after the changes done by another user.
- If you choose to continue working in the business rules manager, you must reload those business rule groups and rule schedules in error from the data source as your local changes of business rule groups and rule schedules in error are no longer usable. Reload a business rule group or rule schedule page, by clicking **Reload** in the Publish and Revert page of the rule for which the error was reported. You can still use local changes in other business rule groups and rule schedules that are not in error.

Overview of selector components

As businesses change, the business processes that drive them must change, too. Some of those changes may require that certain processes return different results than as originally designed without changing the design of the process. The selector component provides the framework for that flexibility.

Selector components provide a single interface to a service that may change results based on certain criteria. The selector component includes an interface and a selector table. Based on the criteria, the selector table determines which component (named the target component) processes the request. The server returns the processing result provided by a target component to the client.

When building a business process, the solution architect identifies the need for a selector component and defines the interface and selector table that the selector component uses to complete processing. The tasks involved in developing a selector component are described in the WebSphere Integration Developer Information Center.

Administering a selector component consists of tasks related to the selector component or tasks related to the selector table.

Restriction: To access any of the selector component pages, you must supply a user ID when you log in to the administrative console. If you are logged in without a user ID, you will receive a warning to log out and log back in with a valid user ID.

Displaying selector components

Displaying selector components is the first step in administering selector components. From the display you can export any or all of the selector components or display the selector tables which comprise the selector components.

Before you begin

You must be at the administrative console for the WebSphere Process Server to perform this task.

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an administrator or a configurator to perform this task.

About this task

To determine which selector components exist in your server, perform the following steps.

Restriction: To access any of the selector component pages, you must supply a user ID when you log in to the administrative console. If you are logged in without a user ID, you will receive a warning to log out and log back in with a valid user ID.

Procedure
1. In the navigation pane, click **Servers** to display the different server types.
2. Click **Application servers** to expand the Application server list.
3. Click the name of your server in the server list.
4. Under **Business Integration** click **Selectors > Selectors** .

   The console displays a list of all the selector components with each component's description.

Displaying selector tables

Displaying selector tables is the first step in administering the tables. The resulting display is a list of target components from which you can alter the processing criteria, change the target component that runs for a specific criterion, add a new target component or delete a target component from the table, thereby removing a criterion.

Before you begin

You must be at the administrative console for the WebSphere Process Server to perform this task.

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an administrator or an operator to perform this task.

About this task

Display a selector table to determine the entries that comprise the table and to perform other selector table-related tasks. To display a selector table, perform the following steps.

Procedure
1. Display the selector components by clicking **Servers > Application servers >** *servername* **> Business Integration > Selectors > Selectors**.
2. Click the selector name from the selector components display to view the selector tables in the selected component.
3. Click one of the selector tables in the display to view the target components that comprise the selector table.

Changing target components

Changing target components allows you to alter selector component processing by either changing the selection criteria for a specific target component, changing the target component for a selection criteria, or changing both the selection criteria and the target component.

Before you begin

To perform this task, a selector table must exist.

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an administrator to perform this task.

About this task

Change a target component to alter the selection criteria or use a different target component for an entry in the selector table. To change target components, perform the following steps.

Important: Before changing target components for long-running applications, stop the application. Do not change target components while a long-running application is processing.

Procedure
1. Display the selector table as described in "Displaying selector tables." Click **Servers > Application Servers >** *servername* **> Business Integration > Selectors > Selectors >** *selectorname*.
2. Click one of the selector tables in the display to view the target components that comprise the selector table.
3. Click the target ID of the target component that you want to change.
4. Change the entry.

Portion of entry to change	Steps to change
Target destination	1. Click the arrow next to the target component list to display the eligible target components.
	2. Select the target component from the list.

Portion of entry to change	Steps to change
Selection criteria	1. Type over either the **Start Date**, **End Date** or both. The date you enter depends on the locale of your system and will display according to the locale format. For the US English locale the format displayed is the following: • Month • Day of month • Year in YYYY format. • Time in HH:MM:SS format • Time zone **Important:** The **Start Date** you specify must be prior to the **End Date** or you will be unable to commit this change.
Target destination and selection criteria	1. Click the arrow next to the target component list to display the eligible target components. 2. Select the target component from the list. 3. Type over either the **Start Date**, **End Date** or both. The date you enter depends on the locale of your system and will display according to the locale format. For the US English locale the format displayed is the following: • Month • Day of month • Year in YYYY format. • Time in HH:MM:SS format • Time zone **Important:** The **Start Date** you specify must be prior to the **End Date** or you will be unable to commit this change.

5. Optional: Click the **Default** check box to make this the default target component.

   If the selection criteria does not fall within the range of any other target components, the selector component uses this target component.

6. Click **Apply** to continue working in this display, or click **OK** to return to the target component display.

7. Click **Save** on the target component display to save the changes to the selector table.

Results

The selector table file now contains the updated selection criteria and target components. The selector component uses the updated selector table to process the next request received.

Adding target components
Add a target component when you need additional processing for a different selection criterion than currently exists in the selector table.

Before you begin

To perform this task, a selector table must exist.

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an administrator to perform this task.

About this task

Add a target component when you need additional flexibility in your business process. The new components can be added while the selector component is active.

To add a target component, perform the following steps.

Procedure

1. Display the selector table as described in "Displaying selector tables". Click **Servers > Application Servers >** *servername* **> Business Integration > Selectors > Selectors >** *selectorname*.
2. Click one of the selector tables in the display to view the target components that comprise the selector table.
3. Click **New** to display a pre-filled target component page.
4. Edit the target destination information to fit the application requirements as described in "Changing target components."
5. Click **OK** to save the target component and return to the target components display.

Results

The selector table now contains the new target components. The selector component uses the updated selector table to process the next request received.

Deleting target components

Deleting target components alters selector component processing by removing the entry in the selector table for a specific selection criterion.

Before you begin

To perform this task, a selector table must exist.

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an administrator to perform this task.

About this task

Delete a target component when the processing is no longer required for the business process. After deleting a target component, if a request comes in and it does not match any other specific selection criteria, the default criteria processes the request.

To delete target components, perform the following steps.

Procedure

1. Display the target components as described in "Displaying selector tables."
2. Click the check box next to the target components that you want to delete, and click **Delete**.
   The system updates the page by displaying the remaining target components.

3. Click **Save**.

The system saves the updated selector table with the entries representing the remaining target components.

Results

The selector table file now contains only the remaining target components. The selector component uses the updated selector table to process the next request received.

Exporting selector components using the administrative console

Export selector components when you have made changes to the selector tables. This will create a file that you can import into your development environment, thereby keeping the development artifacts synchronized with the actual production system artifacts.

Before you begin

Before starting this task, you need to display your selector components as described in "Displaying selector components." Click **Servers > Application servers >** *servername* **> Business Integration > Selectors > Selectors**.

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an administrator or a configurator to perform this task. When security is not enabled, you must log in to the administrative console with a user ID.

About this task

To export selectors using the administrative console, perform the following steps.

Procedure

1. Select the check boxes next to one or more selectors and click **Export**.

The browser displays a list of HTML links to the selector components you chose. This is the Selectors to Export page. Each selector has a file extension of `.zip`.

2. Download the files to your file system by clicking each file name. When the system prompts you to save the file, click **OK**.

Note: If you choose to, you can rename the files as you download them.

3. Click **Back** to return to the list of selectors.

Results

The system saves the files where you specified.

Importing selector components using the administrative console

Import selectors in order to update installed selector components without reinstalling an application.

Before you begin

You must be at the administrative console and have the location of a compressed file created by the export facility.

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an administrator or a configurator to perform this task.

About this task

Import selectors when you have made changes to selectors in use by installed applications, and you are ready to bring those changes into another cluster or server. You can also use this facility to synchronize your development environment with changes in the production environment.

To import selectors using the administrative console, perform the following steps.

Tip: You can also import selector components using the command line.

Procedure
1. Display the selectors on the server to which you are importing the selector components as described in "Displaying selector components." Click **Servers > Application servers >** *servername* **> Business Integration > Selectors > Selectors**.
2. Click **Import**.
3. Specify the path to the file on the Preparing for importing selectors page.

What to do next

Display the selector tables for the updated selectors to verify the changes.

Chapter 8. Bindings

Bindings configure SCA imports and exports to allow interaction via a variety of protocols and transports. SCA applications connect to external clients and services via the transports and protocols of the chosen bindings, which specify the message format and protocol details for a particular interface.

Use JMS bindings, which are asynchronous, if reliability is an important issue for you. Asynchronous communication using messaging queues is often preferred in business processes where reliability is critical. There is a pairing of import and export components with JMS bindings as you would expect from a message consumer and producer.

Web Services, HTTP and EIS bindings can also be used with import and export components.

In the case of an import component only, a stateless session enterprise Java bean (EJB) binding can be used.

Types of bindings

EIS bindings can be used to communicate with external JMS applications, and can also be used from an EIS system to an application or vice-versa. In the first case, you would develop an export component with an EIS binding. In the opposite case you would develop an import component with an EIS binding. EIS bindings are used to access external EIS systems or for external EIS systems to access applications deployed in the server.

JMS bindings are used from a module to a module; that is, you can create an export component in a module with a JMS binding or you can create an import component in a module with a JMS binding. These bindings may share the same JMS resources and thus communicate directly using JMS. You cannot do this with an EIS binding. There are specific JMS bindings for use with the WebSphere embedded JMS provider, the WebSphere MQ JMS provider, and a Generic JMS binding for use with third-party JMS providers.

WebSphere MQ (non-JMS) bindings allow for mediation between, and communication with, native MQ applications, bringing them into the service oriented architecture framework, and providing access to MQ-specific header information.

The following is a complete list of the types of bindings that are supported:
- JMS (WebSphere embedded) bindings
- Generic JMS bindings
- WebSphere MQ JMS bindings
- WebSphere MQ native bindings (WebSphere MQ bindings)
- HTTP bindings
- Web service bindings
- Stateless Session Bean bindings (import only)

It is possible to configure bindings from the command line. Further information can be found in the Reference > Administrative commands section of this infocenter.

SCA EIS bindings

Service Component Architecture (SCA) enterprise information system (EIS) bindings provide connectivity between SCA components and external systems. This communication is mediated by EIS exports and EIS imports.

Your SCA components might require that data be transferred to or from an external EIS. When you create SCA modules requiring such connectivity, you will include (in addition to your SCA component) EIS imports and/or EIS exports for communication with specific external EISs. Details of how to develop EIS imports and exports are provided in the WebSphere Integration Developer Information Center.

The SCA domain is limited to one network deployment cell, so if you use SCA bindings to call a service in one cell from another cell, unique module names cannot be guaranteed. For production use, you should use a different type of binding, for example, MQ or JMS. If you are testing a module with SCA bindings, which is designed to be used to call a service in the same cell, avoid calling from the test cell to the production cell by creating a stub for the service in the test cell itself.

EIS bindings are administered from WebSphere Integration Developer.

Example

A typical SCA module named ContactSyncModule synchronizes contact information between a Siebel system and an SAP system. The SCA component named ContactSync listens via an EIS application export named Siebel Contact for changes to Siebel contacts. The ContactSync SCA component itself makes use of an SAP application via an EIS application import in order to update the SAP contact information accordingly. Since the data structures used for storing contacts is different in Siebel and SAP systems, the ContactSync SCA must provide mapping.

Related concepts

"JMS bindings" on page 158
A Java Message Service (JMS) provider enables messaging based on the Java Messaging Service. It provides J2EE connection factories to create connections for JMS destinations.

"Generic JMS bindings" on page 166
The generic JMS import and export bindings provide connectivity to third-party JMS 1.1 compliant providers. Its operation is similar to that of JMS bindings.

Related tasks

Related information

 Developing services with adapters

EIS bindings: a general perspective

A data binding describes mapping between data passed to the SCA application
and the EIS.

A client uses the appropriate import or export file in the SCA module to define the
functionality of the EIS import or export. At runtime, this model is used to drive
the interactions with the EIS system through the import or export runtime
components.

Key features of EIS bindings

An EIS import is a Service Component Architecture (SCA) import that allows
components in the SCA module to use EIS applications defined outside the SCA
module. An EIS import is used to transfer data from the SCA component to an
external EIS; an EIS export is used to transfer data from an external EIS into the
SCA module.

Imports

The role of the EIS import is to bridge the gap between SCA components and
external EIS systems. External applications can be treated as an EIS export. In this
case, the external application sends its data in the form of a periodic notification.

The EIS import provides SCA components with a uniform view of the applications
external to the module. This allows components to communicate with a variety of
external EISs such as SAP, Siebel or PeopleSoft, using a consistent SCA model.

On the client side of the import there is an interface, exposed by the EIS import
application, with one or more methods, each taking data objects as arguments and
return values. On the implementation side, there is a Common Client Interface
(CCI) implemented by the resource adapter.

The runtime implementation of an EIS import connects the client-side interface and
the CCI. The import maps the invocation of the method on the interface to the
invocation on the CCI. Bindings are created at three levels: the interface binding,
which then uses the contained method bindings, which in turn use data bindings.

The interface binding relates the interface of the import to the connection to the
EIS system providing the application. This reflects the fact that the set of
applications, represented by the interface, is provided by the specific instance of
the EIS and the connection provides access to this instance. The binding element

contains properties with enough information to create the connection (these properties are part of the javax.resource.spi.ManagedConnectionFactory instance).

The method binding associates the method with the specific interaction with the EIS system. For J2C, the interaction is characterized by the set of properties of the javax.resource.cci.InteractionSpec interface implementation. The interaction element of the method binding contains these properties, along with the name of the class thus providing enough information to perform the interaction. The method binding uses data bindings describing the mapping of the argument and result of the interface method to EIS representation.

The runtime scenario for an EIS import is as follows: The method on the import interface is invoked using the SCA programming model. The request, reaching the EIS import handler, contains the name of the method and its arguments. The handler first creates an interface binding implementation; then, using data from the import file, it creates a ConnectionFactory and associates the two. That is, the handler calls setConnectionFactory on the interface binding. The next step is to create the method binding implementation matching the invoked method. The javax.resource.cci.InteractionSpec instance is created and populated; then, data bindings are used to bind the method arguments to a format understood by the resource adapter. At this point, the CCI interface is used to perform the interaction. When the call returns, the data binding is used to create the result of the invocation, and return the result to the caller.

Exports

The role of the EIS export is to bridge the gap between SCA components and external EISs. External applications can be treated as an EIS export. In this case, the external application sends its data in the form of periodic notifications. An EIS export can be thought of as a subscription application listening to an external request from an EIS. The SCA component that utilizes the EIS export views it as a local application.

The EIS export provides SCA components with a uniform view of the applications external to the module. This allows components to communicate with the variety of external EISs like SAP, Siebel or PeopleSoft, using a consistent SCA model.

The export features a listener implementation receiving requests from the EIS. The listener implements either javax.jms.MessageLister interface for the JMS export or a resource adapter-specific listener interface. The export also contains a component implementing interface, exposed to the EIS through the export.

The runtime implementation of an EIS export connects the listener with the component implementing interface. The export maps the EIS request to the invocation of the appropriate operation on the component. Bindings are created at three levels: a listener binding, which then uses contained native method binding, which in turn uses data binding.

The listener binding relates the listener receiving requests with the component exposed through the export. The export definition contains the name of the component; the runtime locates it and forwards requests to it.

The native method binding associates the native method or the event type received by the listener to the operation implemented by the component exposed via the export. There is no relation between the method invoked on the listener and the event type, all the events arrive through one or more methods of the listener. The

native method binding uses the function selector defined in the export to extract the native method name from the inbound data and data bindings to bind the data format of the EIS to a format understood by the component.

The runtime scenario for an EIS export is as follows: The EIS request triggers invocation of the method on the listener implementation. The listener locates and invokes the export handler passing to it all the invocation arguments (for example a JMS message for a JMS export). The handler creates the listener binding implementation. Then, the handler instantiates the function selector and sets it on the listener binding. In the next step, the export handler initializes native method bindings and adds them to the listener binding. For each native method binding, the data bindings are also initialized. Finally, the handler invokes the listener binding. The listener binding locates exported components, and uses the function selector to retrieve the native method name. This name is used to locate native method binding which then invokes the target component.

The adapter interaction style allows for the EIS export binding to invoke the target component either asynchronously (the default) or synchronously.

J2EE resources

The EIS module, an SCA module that follows the EIS module pattern, can be deployed to the J2EE platform.

The deployment of the EIS module to the J2EE platform results in an application that is ready to execute, packaged as an EAR file and deployed to the server. All J2EE artifacts and resources are created; the application is configured and ready to be run.

JCA Interaction Spec and Connection Spec dynamic properties

The SCA EIS binding can accept input for the InteractionSpec and ConnectionSpec specified by using a well-defined child data object that accompanies the payload. This allows for dynamic request-response interactions with a resource adapter through the InteractionSpec and component authentication through the ConnectionSpec.

The javax.cci.InteractionSpec carries information on how the interaction request with the resource adapter should be handled. It can also carry information on how the interaction was achieved after the request. These two-way communications through the interactions are sometimes referred to as conversations.

The SCA EIS binding expects the payload that will be the argument to the resource adapter to contain a child data object called "**properties**". This property data object will contain name/value pairs, with the name of the Interaction Spec properties in a specific format. The formatting rules are:

1. Names must begin with the prefix "IS", followed by the property name. For example, an interaction spec with a Java Bean property called InteractionId, would specify the property name as ISInteractionId.
2. The name/value pair represents the name and the value of the simple type of the Interaction Spec property.

There is an interface with the input of the operation being an "Account" data object. This interface invokes an EIS import binding application with the intention to send and receive dynamic InteractionSpec property called "workingSet" with

value "xyz". One caveat that you can take advantage of is that the Business Graph
or Business Objects in the server contain an underlying "properties" Business
Object that permits the sending of protocol-specific data with the payload. This
properties Business Object is built-in, and does not need to be specified in the XML
schema when constructing a business object. It only needs to be created and used.
If you have your own data types defined based on an XML schema, then you will
need to specify a "**properties**" element that contains your expected name/value
pairs.

```
 BOFactory dataFactory = (BOFactory) \
 serviceManager.locateService("com/ibm/websphere/bo/BOFactory");
  //Wrapper for doc-lit wrapped stle interfaces,
  //skip to payload for non doc-lit
  DataObject docLitWrapper = dataFactory.createByElement /
  ("http://mytest/eis/Account", "AccountWrapper");
```

Create the payload.

```
DataObject account = docLitWrapper.createDataObject(0);
  DataObject accountInfo = account.createDataObject("AccountInfo");
  //Perform your setting up of payload


  //Construct properties data for dynamic interaction

  DataObject properties = account.createDataObject("properties");
```

For name "workingSet", set the value expected "xyz".

```
properties.setString("ISworkingSet", "xyz");


  //Invoke the service with argument

  Service accountImport = (Service) \
  serviceManager.locateService("AccountOutbound");
  DataObject result = accountImport.invoke("createAccount", docLitWrapper);

  //Get returned property
  DataObject retProperties = result.getDataObject("properties");

  String workingset = retProperties.getString("ISworkingSet");
```

You can use ConnectionSpec properties for dynamic component authentication.
The same rules apply as above, except that the property name prefix needs to be
"CS" (instead of "IS"). ConnectionSpec properties are not two-way. The same
"**properties**" Data Object can contain both IS and CS properties. To utilize
ConnectionSpec properties the resAuth specified on the import binding should be
"Application", and the resource adapter must support component authorization.
See chapter 8 of the J2EE Connector Architecture Specification for more details.

Configuring EIS bindings

EIS import bindings are installed in the server as part of your SCA applications.
Administer your imports from the administrative console.

Before you begin

You must have permission to make changes to the master configuration in order to
perform this task.

About this task

You have an installed application that includes an EIS import module.

Procedure

1. View the import that you want to administer. Expand Applications; click SCA Modules; click on the name of the SCA module that you want to administer; under the heading Module Components, expand the module component for Import or Export; and then expand the import or export that you want to administer. You can choose to administer the interfaces or the bindings.

2. Optional: View the WSDL of the import interface. Expand Interfaces and select the interface you want to view. The WSDL of the interface is displayed. The WSDL cannot be edited through the administrative console, but can be altered with text editors.

3. Optional: Administer the binding. Expand Bindings and click on the import or export binding that you want to administer. You can change the port or the name of the imported or exported service.

4. Save changes to the configuration in order for your changes to take effect.

External clients with EIS bindings

The server can send messages to, or receive messages from, external clients using EIS bindings.

An external client, for example a web portal or an EIS, needs to send a message to an SCA module in the server or needs to be invoked by a component from within the server.

The client invokes the EIS import as with any other application, using either the Dynamic Invocation Interface (DII) or Java interface.

First, the external client creates instance of the ServiceManager and looks up the EIS import using its reference name. The result of the lookup is a service interface implementation. Next, the client creates an input argument, a generic DataObject, created dynamically using the data object schema. This step is done using Service Data Object's (SDO's) DataFactory interface implementation. In the last step, the external client invokes EIS and obtains the required results.

Alternatively, the client can invoke EIS import using the Java interface. First, the client creates instance of the ServiceManager and looks up the EIS import using its reference name. The result of the lookup is a Java interface of the EIS import. Next, the client creates an input argument and a typed DataObject, and finally invokes EIS and obtains the required results.

The EIS export interface defines the interface of the exported SCA component that is available to the external EIS applications. This interface can be thought of as the interface that external application like SAP or PeopleSoft will invoke (through the implementation of the EIS export application runtime). The export uses EISExportBinding to bind exported services to the external EIS application, it allows you to subscribe an application contained in your SCA module to listen for EIS service requests. The EIS export binding specifies the mapping between the definition of inbound events as it is understood by the resource adapter (using J2EE Connector Architecture interfaces) and the invocation of SCA operations. The EISExportBinding requires external EIS services to be based on J2EE Connector Architecture 1.5 inbound contracts. The EISExportBinding requires that a DataBinding be specified either at the binding level or the method level.

JMS bindings

A Java Message Service (JMS) provider enables messaging based on the Java Messaging Service. It provides J2EE connection factories to create connections for JMS destinations.

JMS bindings include:

- Service integration bus (SIB) provider binding compliant with JMS JCA 1.5 (*JMS binding*)
- Generic, non-JCA JMS binding, compliant with JMS 1.1 (*generic binding*)
- WebSphere MQ JMS binding, providing JMS provider support for WebSphere MQ and allowing J2EE application interoperability (*WebSphere MQ JMS binding*)

The applications provided through a JMS binding allow a Service Component Architecture (SCA) module to make calls to, and receive messages from, external JMS systems.

Also supported are WebSphere MQ bindings (*WebSphere MQ binding*) which allow native MQ users to handle arbitrary incoming and outgoing message formats (WebSphere MQ required).

The JMS application provides integration through an available JMS JCA 1.5-based resource adapter. The SIB JMS resource adapter is an example of a JCA 1.5-based resource adapter, and complete support is provided for JMS integration of this adapter.

Alternative JMS provider JCA 1.5-based JMS resource adapters are not supported.

Related tasks

"Enabling event sequencing: JMS exports" on page 212
JMS export bindings support event sequencing. In order for events to be handled in the order in which they are received, you must configure properties of the binding.

JMS bindings: a general perspective

JMS bindings provide connectivity between the Service Component Architecture (SCA) environment and external JMS systems.

JMS bindings

The major components of both JMS import and JMS export bindings are:

- Resource adapter: enables managed, bidirectional connectivity between an SCA module and external JMS systems.
- Connections: encapsulate a virtual connection between a client and a provider application.
- Destinations: used by clients to specify the target of messages it produces or the source of messages it consumes.
- Authentication data: used to secure access to the binding.

JMS import bindings

JMS import bindings allow you to import an external JMS application to be used inside your SCA module.

Connections to the associated JMS provider of JMS destinations are created by using a JMS connection factory. Use connection factory administrative objects to manage JMS connection factories for the default messaging provider.

Importing services from external JMS systems utilizes a destination that the data will be sent to and a destination where the reply can be received.

Two types of usage scenario for the JMS import binding are supported:

- One-way: a message is sent to the send destination in the import file. Nothing is set in the replyTo field of the JMS header.
- Two-way (request-response): a message is sent to the send destination specified in the import file. The receive destination is set in the replyTo header property of the outbound message. A message driven bean (MDB) is deployed to listen on the receive destination, and when a reply is received the MDB invokes the callback object. For request-response scenarios, the response can copy the request messageId to the correlationId field of the response message (default), or the response can copy the request correlationId to the correlationId field of the response message.

For both one-way and two-way usage scenarios, dynamic and static header properties can be specified. Some of these properties have special meanings to the SCA JMS runtime.

Figure 7 illustrates how the import is linked to the external service.

Figure 7. JMS import binding resources

JMS export bindings

JMS export bindings provide inbound connectivity from JMS to the SCA system.

The connection that is part of a JMS export is a configurable activation specification.

A JMS export has send and receive destinations. The receive destination is where the incoming message for the target component should be placed. The send destination is where the reply will be sent, unless the incoming message has superseded this using the replyTo header property.

An MDB is deployed to listen to requests incoming to the receive destination specified in the export binding. The destination specified in the send field is used to send the reply to the inbound request if the invoked application provides a reply. The destination specified in the replyTo field of the incoming message overwrites the destination specified in the send.

Figure 8 illustrates how the external requestor is linked to the export.
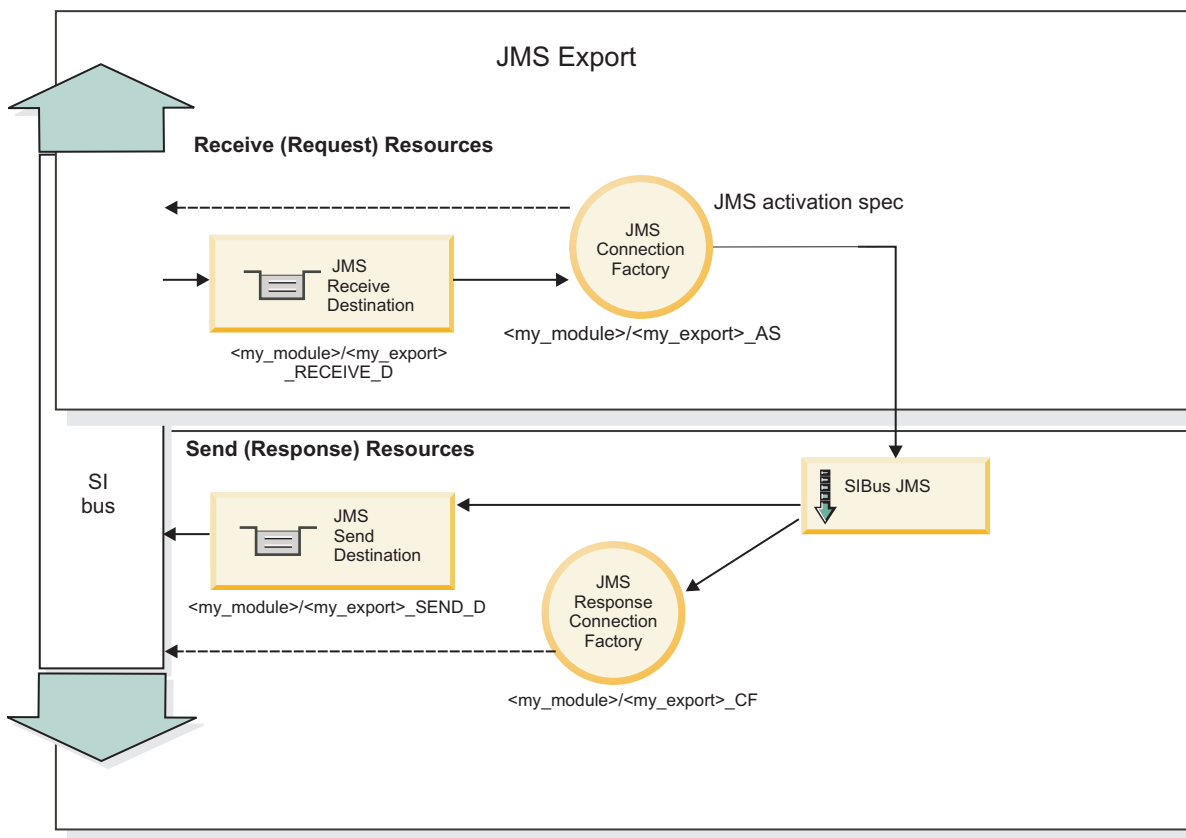
Figure 8. JMS export binding resources

JMS bindings support

The Java Message Service (JMS) provides integration through an available JMS JCA 1.5-based resource adapter. Complete support for JMS integration is provided for the Service Integration Bus (SIB) JMS resource adapter.

Use a JMS provider for JCA 1.5 resource adapter when you want to integrate with an external JCA 1.5-compliant JMS system. External services compliant with JCA 1.5 can receive messages and send messages to integrate with your service component architecture (SCA) components using the SIB JMS resource adapter.

The use of other provider-specific JCA 1.5 resource adapters is not supported.

JMS modules cannot be deployed to a J2SE environment. Such modules are only deployable to a J2EE environment.

Key features of JMS bindings

Key features of JMS import and export bindings include headers and created J2EE resources.

Differences from EIS

Imports

The JMS import application differs from an EIS import in its runtime invocation with respect to handling of responses. The JMS implementation is asynchronous, and the SCA programming model requires it to support three invocations: one-way, two-way (request-response), and callback.

When the JMS import is deployed, a message driven bean (MDB), provided by the runtime environment, is deployed to listen for replies to the request message. The MDB is associated with (listens on) the destination sent with the request in the replyTo header field of the JMS message.

Exports

JMS export bindings differ from EIS export bindings in their handling of the return of the result. A JMS export explicitly sends the response to the JMSReplyTo destination specified on the incoming message. If none is specified, the send destination is used.

When the JMS export is deployed, a message driven bean (a different MDB than the one used for JMS imports) is deployed. It listens for the incoming requests on the receive destination and then dispatches the requests to be processed by the SCA runtime.

Special headers

Special header properties are used in JMS imports and exports to tell the target how to handle the message.

For example, TargetFunctionName maps from the native method to the operation method. It is used by the default function selector (JMSFunctionSelectorImpl) to extract the native method name from the JMS message.

J2EE resources

A number of J2EE resources are created when JMS imports and exports are deployed to a J2EE environment.

ConnectionFactory
    Used by clients to create a connection to the JMS provider.

ActivationSpec
    Imports use this for receiving the response to a request; exports use it when configuring the message endpoints that represent MDBs in their interactions with the messaging system.

Destinations

- Send destination: on an import, this is where the request or outgoing message is sent; on an export, this is the destination where the response message will be sent, if not superseded by the JMSReplyTo header field in the incoming message.
- Receive destination: where the incoming message should be placed; with imports, this is a response; with exports, this is a request.
- Callback destination: SCA JMS system destination used to store correlation information. Do not read or write to this destination.

The installation task creates the ConnectionFactory and three destinations. It also creates the ActivationSpec to enable the runtime MDB to listen for replies on the receive destination. The properties of these resources are specified in the import or export file.

Working with JMS bindings

JMS imports and exports can be managed in the server.

You can configure and administer JMS imports and exports using the administrative console.

Detailed instructions on creating JMS imports and exports can be found in the WebSphere Integration Developer information center at **WebSphere Integration Developer** > **Developing integration applications** > **Accessing external services with messaging systems > Java Message Service (JMS)** > **Working with JMS bindings**.

Parameters of JMS bindings

The JMS import and export bindings can be installed with all the necessary connection factories created, or they can be designed to point to a pre-configured set of artifacts on the server.

Typically, JMS import and export bindings are created in WebSphere Integration Developer. At development time you have an option to create the connections and destinations required for the JMS binding at install time, i.e., when the component is installed on your server. Your other option is to specify the JNDI name of the resources on the server that you intend your JMS imports and exports to utilize.

Configuring the JMS binding depends upon which option was selected.

In the case where new message provider resources are created (that is, the resources are created on the server at install time), the resources will exist and can be located and administered using the administrative console. The JNDI names of the generated artifacts for imports and exports are described in the following tables.

Table 7. JMS import bindings: Names and JNDI names of resources created at installation on the server

Resource	Module name	Import name	Resource global JNDI name
Connection Factory	jms.module	my/import	jms.module/my/import_CF
Activation Spec	jms.module	my/import	jms.module/my/import_AS

Table 7. JMS import bindings: Names and JNDI names of resources created at installation on the server (continued)

Resource	Module name	Import name	Resource global JNDI name
Destination	jms.module	my/import	jms.module/my/import_SEND_D, jms.module/my/import_RECEIVE_D, jms.module/my/import_CALLBACK_D
SIB Destinations	jms.module	my/import	jms.module.my.import_SEND_D_SIB, jms.module.my.import_RECEIVE_D_SIB, jms.module.my.import_CALLBACK_D_SIB

Table 8. JMS export bindings: Names and JNDI names of resources created at installation on the server

Resource	Module name	Export name	Resource global JNDI name
Connection Factory	jms.module	my/export	jms.module/my/export_CF
Activation Spec	jms.module	my/export	jms.module/my/export_AS
Destination	jms.module	my/export	jms.module/my/export_SEND_D, jms.module/my/export_RECEIVE_D, jms.module/my/export_CALLBACK_D
SIB Destinations	jms.module	my/export	jms.module.my.export_SEND_D_SIB, jms.module.my.export_RECEIVE_D_SIB, jms.module.my.export_CALLBACK_D_SIB

Note: The resources are created at the server scope. The scope in the administrative console is initially set at "All scopes". You must set the scope to "cell" or "node" to create a new resource. You can select an existing resource from the default list.

If you select the other option and the binding is expecting to find on the server resources for its use, you must have these resources installed and the import and binding files must contain the appropriate JNDI names. The association between the JMS binding and the resources will then be made.

Configuring JMS bindings

You can configure JMS import and export bindings to apply special features of the resource. The administrative tasks are performed using the WebSphere administrative console.

Before you begin

You must have permission to make and save changes to the profile on the administrative console.

About this task

The JMS binding must be installed as part of a Service Component Architecture (SCA) application in your server profile.

Procedure

1. Select the SCA module. From the administrative console, click **Applications** → **SCA Modules** and then click the *modulename*. The configuration page opens.

2. Under **Module components**, expand **Imports** or **Exports**. This reveals a list of the installed imports or exports.

3. Expand the import or export and then expand **Binding**. This reveals a list of available bindings to administer.

4. Click on the binding to be administered. The general properties of the binding are displayed:

   - The **Send Resources** category contains the Connection Factory and the Send Destination.

   - The **Receive Resources** category contains the Response Connection Factory, the Listener Port, and the Activation Specification.

   - The **Advanced Resources** category contains Callback resources and other available resources.

   Note: You can also access a resource by typing the JNDI name in the text box. Doing so, however, allows you to enter the name of a resource that is not yet configured.

5. Administer the desired resource:

   a. Click the **Browse...** button to open a window with a list of JNDI names; then, choose the desired JNDI name and click Select.

   b. Click the **Configure...** button to open the corresponding panel referred to by the JNDI name. While most resources can be configured at cluster scope, selecting the **Configure...** option at Listener Port displays a panel showing all listener ports with the cluster Listener Port names for all the members of the given cluster; you can then select one listener port.

      When **Configure...** has been selected, the corresponding server panel will open.

6. When you are ready to administer your settings, click **Apply** or **OK**.

What to do next

The application will be automatically restarted for the JNDI bindings to take effect.

JMS and WebSphere MQ JMS headers

JMS and WebSphere MQ JMS headers are Service Data Objects (SDO) that contain all the properties of the JMS (or WebSphere MQ JMS) message properties. These properties can be from the inbound message or they can be the properties that will be applied to the outbound message.

JMS and WebSphere MQ JMS headers are provided using the system programming interface; consequently, you set or get the headers by accessing the service component architecture message. You cannot set or get the header from a client program. You can, however, use an ESB mediation to retrieve the header values or set them on an outgoing request.

The properties that can be set on the JMSMessage Header are:
- **JMSType** and **JMSCorrelationID** – values of the specific predefined message header properties
- **JMSDeliveryMode** – values for delivery mode (Persistent or NonPersistent; default is Persistent)
- **JMSPriority** – priority value (0 to 9; default is JMS_Default_Priority)

The JMS and WebSphere MQ JMS headers and message properties are only used when the base service component architecture SCDL binding switch is turned on. When the switch is turned on, the JMS header information is propagated. By default, this switch is on. To prevent context information propagation, change the value to false.

When context propagation is enabled, header information is allowed to flow to the message or to the target component. To turn on and off context propagation, specify true or false for the contextPropagationEnabled attribute of the import and export bindings. For example:

```
<esbBinding xsi:type="eis:JMSImportBinding" contextProgagationEnabled="true">
```

The default is true.

> **Related concepts**
>
> "Generic JMS headers" on page 174
> Generic JMS headers are Service Data Objects (SDO) that contain all the properties of the generic JMS message properties. These properties can be from the inbound message or they can be the properties that will be applied to the outbound message.
>
> "WebSphere MQ headers" on page 190
> WebSphere MQ headers incorporate certain conventions for conversion to the service component architecture (SCA) messages.

External clients

The server can send messages to, or receive messages from, external clients using JMS bindings.

An external client such as a web portal or an enterprise information system can send a message to an SCA module in the server, or it can be invoked by a component from within the server.

The JMS export components deploy message driven beans (MDBs) to listen to requests incoming to the receive destination specified in the export binding. The destination specified in the send field is used to send the reply to the inbound request if the invoked application provides a reply. Thus, an external client is able to invoke applications via the export binding.

JMS imports bind to, and can deliver messages to, external clients. This message might or might not demand a response from the external client.

Working with external clients

An external client (that is, outside of the server) might need to interact with an application installed in the server.

About this task

Consider a very simple scenario in which an external client wishes to interact with a generic application on the server. The figure depicts a typical simple scenario.
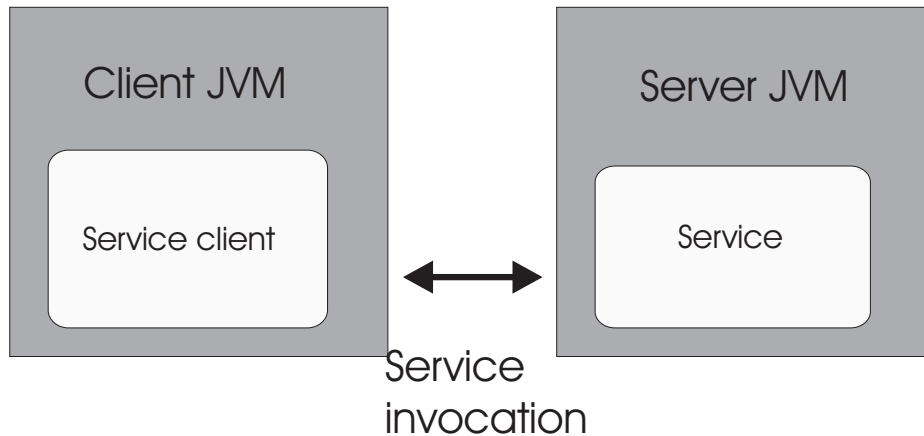
Figure 9. Simple use-case scenario: external client interacts with server application

The application is exported using a JMS binding; this makes the application available to external clients.

When you have an external client in a Java virtual machine (JVM) separate from your server, there are several steps you must take in order to make a connection and interact with a JMS export. The client obtains an InitialContext with the correct values, then looks up the resources through JNDI. The client then uses the JMS 1.1 spec client to access the destinations and the send and receive messages on the destinations.

The default JNDI names of the resources created automatically by the runtime are listed in the configuration topic of this section. However, if you have pre-created resources, then use those JNDI names.

Procedure

1. Configure JMS destinations and connection factory to send the message.
2. Make sure that the JNDI context, the port for the SIB resource adapter, and the messaging bootstrapping port are correct.

   The server uses some default ports but if there are more servers installed in that machine, alternate ports are created at installation time to avoid conflicts with other server instances. You can determine which ports your server is employing with the administrative console. Go to **Servers** ⊁ **Application Servers** ⊁ *your_server_ name* ⊁ **Configuration** and click **Ports** under **Communication**. You can then edit the port being used.
3. The client obtains an initial context with the correct values, then looks up the resources through JNDI.
4. Using JMS 1.1 specifications, the client accesses the destinations, and the send and receive messages on the destinations.

Generic JMS bindings

The generic JMS import and export bindings provide connectivity to third-party JMS 1.1 compliant providers. Its operation is similar to that of JMS bindings.

The service provided through a JMS binding allows a Service Component Architecture (SCA) module to make calls or receive messages from external systems. The system can be an external JMS system.

The generic JMS binding application provides integration with non-JCA 1.5-compliant JMS providers that support JMS 1.1 and implement the optional JMS Application Server Facility. The generic JMS binding supports those JMS providers that do not support JCA 1.5 but do support the Application Server Facility of the JMS 1.1 specification, including Oracle AQ, TIBCO, SonicMQ, WebMethods, BEA WebLogic, and WebSphere MQ. SIB is not supported, as SIB is a JCA 1.5 JMS provider.

A user would use this generic application when integrating with a non-JCA 1.5-compliant JMS-based system within an SCA environment. The target external applications can then receive messages and send messages to integrate with an SCA component.

Related tasks

"Enabling event sequencing: generic JMS exports" on page 214
Generic JMS export bindings support event sequencing. In order for events to be handled in the order in which they are received, you must configure properties of the binding.

Generic JMS bindings: a general perspective

Generic JMS bindings are non-JCA JMS bindings that provide connectivity between the Service Component Architecture (SCA) environment and external JMS systems that are compliant with JMS 1.1 and that implement the optional JMS Application Server Facility.

Generic JMS bindings

The major components of the bindings for both generic JMS imports and generic JMS exports are:

- Resource adapter: enables managed, bidirectional connectivity between enterprise information systems (EISs) and J2EE components.
- Connections: encapsulate a virtual connection between a client and a provider application.
- Destinations: used by clients to specify the target of messages it produces or the source of messages it consumes.
- Authentication data: used to secure access to the binding.

Generic JMS import bindings

Generic JMS import bindings provide outbound connectivity from SCA applications to non-JCA 1.5-compliant JMS providers.

The connection part of a JMS import is a connection factory. A connection factory, the object a client uses to create a connection to a provider, encapsulates a set of connection configuration parameters defined by an administrator. Each connection factory is an instance of the ConnectionFactory, QueueConnectionFactory, or TopicConnectionFactory interface.

Importing services from JMS utilizes a destination that the data will be sent to and a destination where the reply can be received.

Two types of usage scenario for JMS import binding are supported:

- One-way: a message is sent to the send destination specified in the import file. Nothing is sent to the ReplyTo field of the JMS header.

- Two-way (request-response): a message is sent to the send destination in the import file. The receive destination is set in the ReplyTo header property. A message driven bean (MDB) is deployed to listen on the receive destination; when a reply is received, the MDB invokes the callback object. For request-response scenarios, the response can copy the request messageId to the correlationId field of the response message (default), or the response can copy the request correlationId to the correlationId field of the response message.

For both one-way and two-way usage scenarios, dynamic and static header properties can be specified. Some of these properties have special meanings to the SCA JMS runtime.

Figure 10 illustrates how the import is linked to the external service.
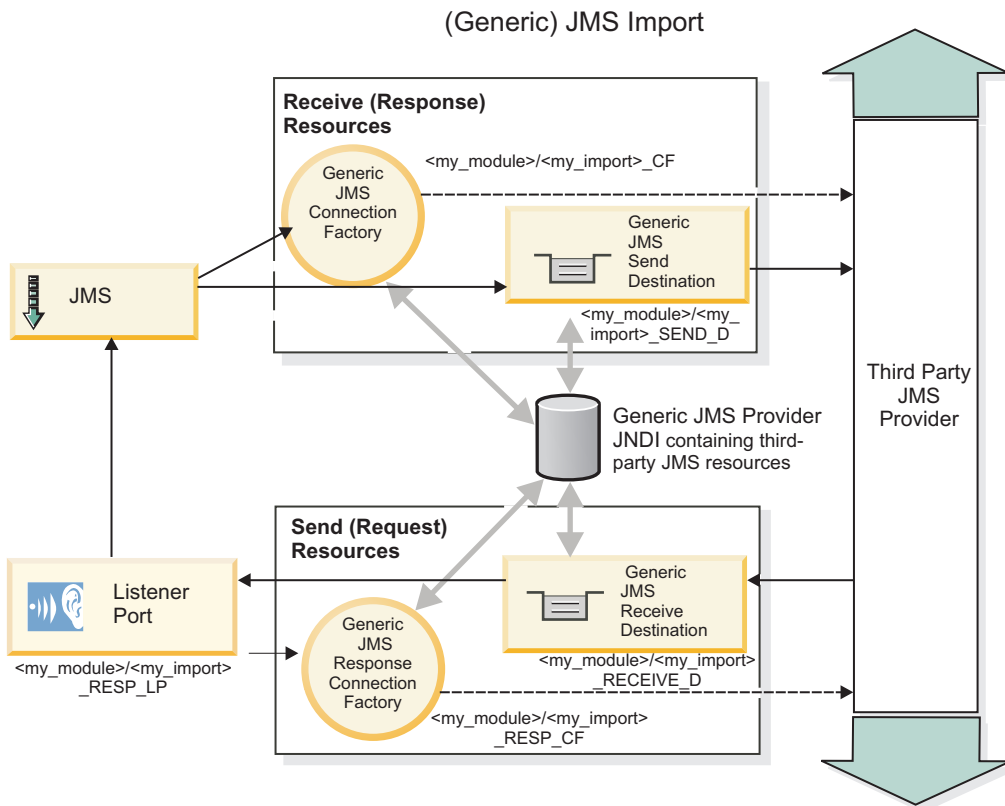
Figure 10. Generic JMS import binding resources

Generic JMS export bindings

Generic JMS export bindings provide inbound connectivity from JMS to the SCA system.

The connection part of a JMS export is composed of a ConnectionFactory and a ListenerPort.

A generic JMS export has send and receive destinations. The receive destination is where the incoming message for the target component should be placed. The send destination is where the reply will be sent, unless the incoming message has superseded this using the replyTo header property. An MDB is deployed to listen to requests incoming to the receive destination specified in the export binding.

- The destination specified in the send field is used to send the reply to the inbound request if the invoked application provides a reply.
- The destination specified in the replyTo field of the incoming message overwrites the destination specified in the *send* field.
- For request/response scenarios, the response can copy the request messageId to the correlationId field of the response message (default), or the response can copy the request correlationId to the to the correlationId field of the response message.

Figure 11 illustrates how the external requestor is linked to the export.
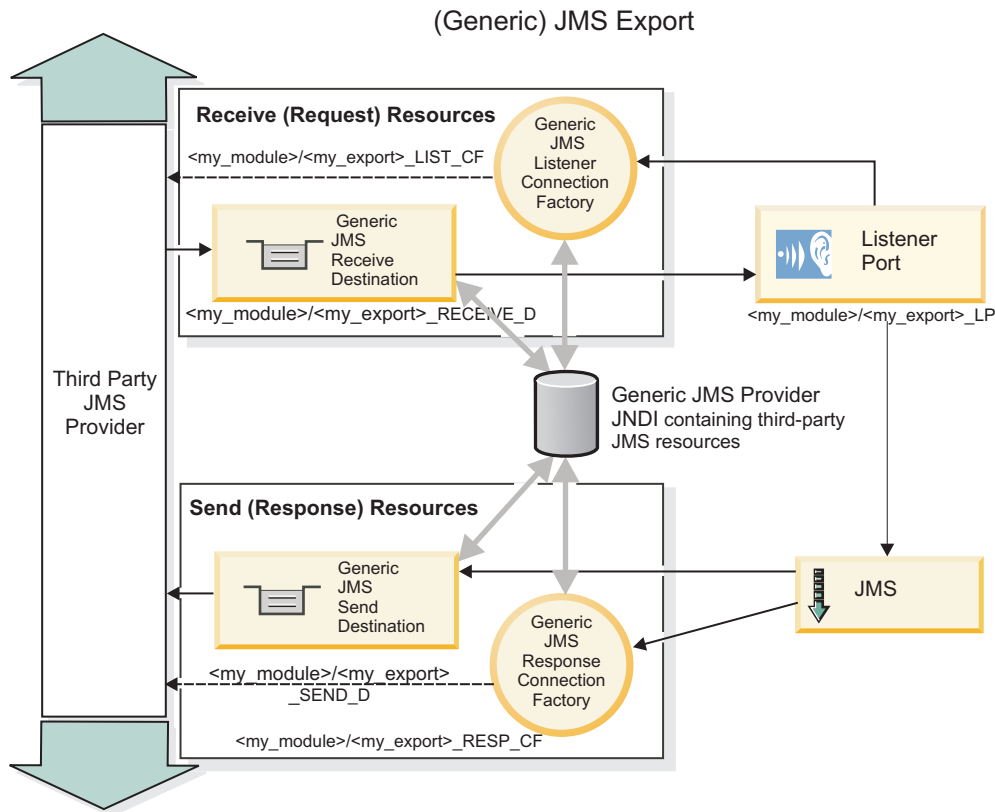
Figure 11. Generic JMS export binding resources

Key features of generic JMS bindings

The features of the generic JMS import and export binding are consistent with those of the WebSphere embedded JMS and MQ JMS import bindings. Key features include header defininitions and access to existing J2EE resources. However, due to its generic nature, there are no JMS provider-specific connectivity options, and this binding has limited capability to generate resources at deployment and installation.

Generic imports

Like the MQ JMS import application, the generic JMS implementation is asynchronous and supports three invocations: one-way, two-way (also known as request-response), and callback.

When the JMS import is deployed, a message driven bean (MDB) provided by the runtime environment is deployed. The MDB listens for replies to the request

message. The MDB is associated with (listens on) the destination sent with the request in the replyTo header field of the JMS message.

Generic exports

Generic JMS export bindings differ from EIS export bindings in their handling of the return of the result. A generic JMS export explicitly sends the response to the replyTo destination specified on the incoming message. If none is specified, the send destination is used.

When the generic JMS export is deployed, a message driven bean (a different MDB than the one used for generic JMS imports) is deployed. It listens for the incoming requests on the receive destination and then dispatches the requests to be processed by the SCA runtime.

Special headers

Special header properties are used in generic imports and exports to tell the target binding how to handle the message.

For example, the TargetFunctionName property is used by the default function selector to identify the name of the operation in the import or export's interface that is being invoked.

J2EE resources

A number of J2EE resources are created when a JMS binding is deployed to a J2EE environment.
- Listener port for listening on the receive (response) destination (two-way only) for imports, and on the receive (request) destination for exports
- Generic JMS connection factory for the outboundConnection (import) and inboundConnection (export)
- Generic JMS destination for the send (import) and receive (export; two-way only) destinations
- Generic JMS connection factory for the responseConnection (two-way only and optional; otherwise, outboundConnection is used for imports, and inboundConnection is used for exports)
- Generic JMS destination for the receive (import) and send (export) destination (two-way only)
- Default messaging provider callback JMS destination used to access the SIB callback queue destination (two-way only)
- Default messaging provider callback JMS connection factory used to access the callback JMS destination (two-way only)
- SIB callback queue destination used to store information about the request message for use during response processing (two-way only)

The installation task creates the ConnectionFactory, the three destinations, and theActivationSpec from the information in the import and export files.

Working with generic bindings

Generic JMS bindings can be administered from within the server.

You can configure and administer generic JMS imports and exports using the administrative console.

Detailed instructions on creating generic JMS imports and exports can be found in the WebSphere Integration Developer information center at **WebSphere Integration Developer** > **Developing integration applications** > **Accessing external services with messaging systems** > **Generic JMS**.

Parameters of generic JMS bindings

The generic JMS import and export bindings can be installed with all the necessary connection factories created, or they can be designed to point to a pre-configured set of artifacts on the server.

Typically, the generic JMS bindings are created in WebSphere Integration Developer. At development time you can either create the connections and destinations required for the JMS binding at the time the component is installed on your server, or you can specify the JNDI name of the resources on the server that you intend your JMS import or export to utilize.

Configuring the generic JMS binding depends upon which option was selected.

In the case where new message provider resources are created (i.e., the resources are created on the server at install time), the resources will exist and can be located and administered using the administrative console. The JNDI names of the generated artifacts are described in the following tables.

Table 9. Generic JMS imports: Names and JNDI names of resources created at installation on the server

Resource	Generated resource JNDI name
outboundConnection	[moduleName]/[importName]_CF
responseConnection	[moduleName]/[importName]_RESP_CF
send destination	[moduleName]/[importName]_SEND_D
receive destination	[moduleName]/[importName]_RECEIVE_D
callback destination	[moduleName]/[importName]_CALLBACK_D

Table 10. Generic JMS exports: Names and JNDI names of resources created at installation on the server

Resource	Generated resource JNDI name
inboundConnection	[moduleName]/[exportName]_LIS_CF
responseConnection	[moduleName]/[exportName]_RESP_CF
receive destination	[moduleName]/[exportName]_RECEIVE_D
send destination	[moduleName]/[exportName]_SEND_D
callback destination	[moduleName]/[exportName]_CALLBACK_D

Note: The resources are created at the server scope. The scope in the administrative console is initially set at "All scopes". You must set the scope to "cell" or "node" to create a new resource. You can select an existing resource from the default list.

In the alternate case, where the JMS import is expecting to find required resources on the server, you must have these resources installed and the import and export files must contain the JNDI names. The association between the JMS binding and the resources will then be made.

Setting up connectivity for the generic JMS binding

You must set up connectivity to and from a third-party JMS provider to use the generic JMS binding.

Before you begin

You must have permission to make and save changes to the profile on the administrative console. You must have the appropriate permissions to make and save changes in WebSphere Integration Developer and in WebSphere Application Server.

About this task

This task provides a procedural outline only; providing specific instructions for individual third-party JMS providers is beyond the scope of this topic. The application in this scenario contains a mediation component connection to other applications at both ends by means of the generic JMS binding; the application contains an interface with a single two-way operation.

Procedure

1. Configure your third-party JMS provider to create a queue manager, queues and JMS connection factories and destinations using the provider-specific tooling.
2. In WebSphere Application Server, you must define a generic messaging provider.
3. In WebSphere Integration Developer, you must perform the following tasks:
   a. Add an import and export to the application and connect them to a previously-implemented mediation component.
   b. Add a generic JMS binding to both the export and the import.
   c. Generate Binding –> Messaging binding –> Generic JMS binding.
   d. Set the genericMessagingProviderName property on both the import and export to match the properties previously defined to WebSphere Application Server.
   e. Set the ExternalJNDIName for the connections and send/receive destinations to match those defined in your third-party JMS provider tooling.
4. Deploy the application to a single server. Make sure that the third-party JMS provider queue manager is running and available for connection, and that the context to which the generic messaging provider definition points in WebSphere Application Server is available. Build and deploy your application using WebSphere Integration Developer or service deploy.
5. Start the application.
6. Run the application. The application can be executed by placing messages on the third-party JMS provider queue defined in the generic JMS export receive destination. Responses will be returned to the generic JMS export send destination. Similarly, the application will issue requests to the generic JMS import send destination and expect responses on the generic JMS import receive destination.

Configuring generic JMS bindings

You can administer generic JMS imports bindings to configure special features of the resource. The administrative tasks are performed using the administrative console.

Before you begin

You must have permission to make and save changes to the profile on the administrative console, and you must have completed the connectivity setup procedure.

About this task

The generic JMS import must be installed as part of a Service Component Architecture (SCA) application in your server profile.

Procedure

1. Select the SCA module. From the administrative console, select **Applications → SCA Modules** and then select the *modulename*. The configuration page opens.
2. Under **Module components**, expand **Imports** or **Exports**. This reveals a list of the installed imports or exports.
3. Expand the import or export and then expand **Binding**. This reveals a list of available bindings to administer.
4. Click on the binding to be administered. The general properties of the binding are displayed:
   - The **Send Resources** category contains the Connection Factory and the Send Destination.
   - The **Receive Resources** category contains the Response Connection Factory, the Listener Port, and the Activation Specification.
   - The **Advanced Resources** category contains Callback resources and other available resources.

   Note: You can also access a resource by typing the JNDI name in the text box. Doing so, however, allows you to enter the name of a resource that is not yet configured.
5. Administer the desired resource:
   a. Click the **Browse...** button to open a window with a list of JNDI names; then, select the desired JNDI name. The selected name will populate the appropriate text field.
   b. Click the **Configure...** button to open the corresponding panel referred to by the JNDI name. While most resources can be configured at cluster scope, selecting the **Configure...** option at Listener Port displays a panel showing all listener ports with the cluster Listener Port names for all the members of the given cluster; you can then select one listener port.

      When **Configure...** has been selected, the corresponding WebSphere Application Server panel will open.
6. When you are ready to administer your settings, click **Apply** or **OK**.

What to do next

The application will be automatically restarted for the JNDI bindings to take effect.

Generic JMS headers

Generic JMS headers are Service Data Objects (SDO) that contain all the properties of the generic JMS message properties. These properties can be from the inbound message or they can be the properties that will be applied to the outbound message.

The headers are provided using the system programming interface; consequently, you set or get the headers by accessing the service component architecture message. You cannot set or get the header from a client program. You can, however, use an ESB mediation to retrieve the header values or set them on an outgoing request.

The following properties are set statically on the methodBinding:
- JMSType
- JMSCorrelationID
- JMSDeliveryMode
- JMSPriority

The generic JMS binding also supports dynamic modification of JMS headers and properties in the same manner as the JMS and MQJMS bindings.

Some generic JMS providers place restrictions on which properties can be set by the application and in what combinations. You must consult your third-party product documentation for more information. However, an additional property has been added to the methodBinding, ignoreInvalidOutboundJMSProperties, which allows any exceptions to be propagated.

The generic JMS headers and message properties are only used when the base service component architecture SCDL binding switch is turned on. When the switch is turned on, context information is propagated. By default, this switch is on. To prevent context information propagation, change the value to false.

When context propagation is enabled, header information is allowed to flow to the message or to the target component. To turn on and off context propagation, specify true or false for the contextPropagationEnabled attribute of the import and export bindings. For example:

```
<esbBinding xsi:type="eis:JMSImportBinding" contextProgagationEnabled="true">
```

The default is true.

Related concepts

"JMS and WebSphere MQ JMS headers" on page 164
JMS and WebSphere MQ JMS headers are Service Data Objects (SDO) that contain all the properties of the JMS (or WebSphere MQ JMS) message properties. These properties can be from the inbound message or they can be the properties that will be applied to the outbound message.
"WebSphere MQ headers" on page 190
WebSphere MQ headers incorporate certain conventions for conversion to the service component architecture (SCA) messages.

WebSphere MQ JMS bindings

The WebSphere MQ JMS binding provides integration with WebSphere MQ JMS-based providers.

Use the WebSphere MQ JMS provider applications to integrate directly with external JMS or MQ JMS systems from your server environment. This eliminates the need to use MQ Link or Client Link features of the Service Integration Bus.

Importing *from* WebSphere MQ utilizes a destination that the data will be sent to and a destination where the reply can be received. Conversion for the data to and from a JMS message is accomplished through the JMS data binding edge component.

Exporting *to* WebSphere MQ utilizes a destination where the request can be received and the response can be sent. The conversion for the data to and from a JMS message is done through the JMS data binding. The function selector provides a mapping to the operation on the target component to be invoked.

Related concepts

"WebSphere MQ bindings" on page 183
The WebSphere MQ application provides Service Component Architecture connectivity with WebSphere MQ applications.

Related tasks

"Enabling event sequencing: WebSphere MQ JMS exports" on page 213
WebSphere MQ JMS export bindings support event sequencing. In order for events to be handled in the order in which they are received, you must configure properties of the binding.

WebSphere MQ JMS bindings: a general perspective

The WebSphere MQ JMS application provides integration with WebSphere MQ using the JMS API.

WebSphere MQ administrative tasks

The WebSphere MQ system administrator is expected to create the underlying WebSphere MQ Queue Manager, which the WebSphere MQ JMS bindings will utilize before running an application containing these bindings.

WebSphere MQ JMS import bindings

The WebSphere MQ JMS import binding provides outbound connectivity from Service Component Architecture (SCA) applications to WebSphere MQ JMS-based providers. You must be using a supported version of WebSphere MQ. Detailed hardware and software requirements can be found on the IBM support pages.

Two types of usage scenario for WebSphere MQ JMS import bindings are supported:

- One-way: a message is sent to the send destination in the import file. Nothing is sent to the replyTo field of the JMS header.
- Two-way (request-response): a message is sent to the send destination in the import file. The receive destination is is set in the replyTo header field. A message-driven bean (MDB) is deployed to listen on the receive destination, and when a reply is received the MDB invokes the callback object. For request-response scenarios, the response is expected to have the correlationId set to the sent message messageID for the default correlation scheme.

For both one-way and two-way usage scenarios, dynamic and static header properties can be specified. Some of these properties have special meanings to the SCA JMS runtime.

Figure 12 illustrates how the import is linked to the external service.

Figure 12. WebSphere MQ JMS import binding resources

WebSphere MQ JMS export bindings

The WebSphere MQ JMS export binding provides inbound connectivity from WebSphere MQ-based JMS provider to the SCA system.

An MDB is deployed to listen to requests incoming to the receive destination specified in the export binding. The destination specified in the send field is used to send the reply to the inbound request if the invoked application provides a reply. The destination specified in the replyTo field of the response message overwrites the destination specified in the send field.

Figure 13 on page 177 illustrates how the external requestor is linked to the export.

Figure 13. WebSphere MQ JMS export binding resources

WebSphere MQ JMS bindings support

The WebSphere MQ JMS application provides integration with WebSphere MQ JMS-based providers.

Use the WebSphere MQ JMS provider applications when you want to integrate directly with a WebSphere MQ JMS-based system within a J2EE programming environment. This eliminates the need for MQ Link or Client Link features of the Service Integration Bus.

Importing *from* WebSphere MQ JMS utilizes a destination that the data will be sent to and a destination where the reply can be received. Conversion for the data to and from a JMS message is accomplished through the JMS data binding edge component.

Exporting *to* WebSphere MQ JMS utilizes a destination where the request can be received and the response can be sent. The conversion for the data to and from a JMS message is done through the JMS data binding. The function selector serves the purpose of providing a mapping to the operation on the target component to be invoked.

The configuration of WebSphere MQ Queue Managers is left up to the WebSphere MQ system administrator.

> **Related concepts**
>
> "WebSphere MQ bindings support" on page 185
> The WebSphere MQ application provides integration with native MQ-based applications.

Key features of WebSphere MQ JMS bindings

Key features of WebSphere MQ JMS bindings include headers, J2EE artifacts and created J2EE resources.

Headers

A JMS message header contains a number of predefined fields containing values used by both clients and providers to identify and to route messages.

JMSCorrelationID
> Links to a related message. Typically, this field is set to the message identifier string of the message that is being replied to.

TargetFunctionName
> This header is used on the export binding to map from native method to operation method. Setting the string JMS header property TargetFunctionName will indicate to the JMSFunctionSelectorImpl of the export the native method that should be used to map to the operation method. To use this, specify the native method as in the method binding as the value. You must set this as a JMS header property in the import binding definition if you expect the export to use the JMSFunctionSelectorImpl.

Correlation schemes

The WebSphere MQ JMS bindings provide various correlation schemes that are used to determine how to correlate request messages with response messages.

RequestMsgIDToCorrelID
> The JMSMessageID is copied to the JMSCorrelationID field. This is the default setting.

RequestCorrelIDToCorrelID
> The JMSCorrelationID is copied to the JMSCorrelationID field.

J2EE resources

A number of J2EE resources are created when a MQ JMS import is deployed to a J2EE environment.

Parameters

MQ Connection Factory
> Used by clients to create a connection to the MQ JMS provider.

Response Connection Factory
> Used by the SCA MQ JMS runtime when the send destination is on a different Queue Manager than the receive destination.

Listener Port
> Specifies an association between a connection factory, a destination, and a

message-driven bean. This enables deployed message-driven beans associated with the port to retrieve messages from the destination.

Destinations
- Send destination:
    - Imports: Where the request or outgoing message is sent.
    - Exports: Where the response message will be sent if it is not superseded by the JMSReplyTo header field of the incoming message.
- Receive destination:
    - Imports: Where the response or incoming message should be placed.
    - Exports: Where the incoming or request message should be placed.

Working with WebSphere MQ JMS bindings

WebSphere MQ JMS bindings can be administered in WebSphere Process Server.

Use the administrative console to access WebSphere MQ JMS bindings.

For detailed instructions on generating WebSphere MQ JMS imports and exports, see "Working with MQ JMS bindings" in the WebSphere Integration Developer Information Center [WID]

Related concepts

"Working with WebSphere MQ bindings" on page 187
WebSphere MQ bindings can be administered from within the server.

Related tasks

"Enabling event sequencing: WebSphere MQ JMS exports" on page 213
WebSphere MQ JMS export bindings support event sequencing. In order for events to be handled in the order in which they are received, you must configure properties of the binding.

Parameters of MQ JMS bindings

MQ JMS bindings can be installed with all the necessary connection factories created, or they can be designed to point to a pre-configured set of artifacts on the server.

Typically, MQ JMS bindings are created in WebSphere Integration Developer. At development time you have an option to create the connections and destinations required for the MQ JMS binding at install time, i.e., when the component is installed on your server. Alternatively, you can specify the JNDI name of the resources on the server that you intend your MQ JMS binding to utilize.

Configuring the MQ JMS binding depends upon which option was selected.

In the case where new message provider resources are created (i.e., the resources are created on the server at install time), the resources will exist and can be located and administered using the administrative console. The JNDI names of the generated artifacts are described in the following tables.

Table 11. MQ JMS imports: Names and JNDI names of resources created at installation on the server

Resource	Module name	Import name	Resource global JNDI name
Outbound Connection Factory	mqjms.module	my/import	mqjms.module/my/import_MQ_CF

Table 11. MQ JMS imports: Names and JNDI names of resources created at installation on the server (continued)

Resource	Module name	Import name	Resource global JNDI name
Response Listener Port	mqjms.module	my/import	mqjms.module.my.import_RESP_LP (Note: This is only a name, not JNDI)
Response Connection Factory	mqjms.module	my/import	mqjms.module/my/import_RESP_CF
Send	mqjms.module	my/import	mqjms.module/my/import_MQ_SEND_D
Receive	mqjms.module	my/import	mqjms.module/my/export_MQ_RECEIVE_D
SIB Callback Destination	mqjms.module	my/import	mqjms.module/my/import_MQ_CALLBACK_D
SIB Callback Connection Factory	All modules	my/import	SCA.MQJMS/Callback_CF

Table 12. MQ JMS exports: Names and JNDI names of resources created at installation on the server

Resource	Module name	Export name	Resource global JNDI name
Listener Port	mqjms.module	my/export	mqjms.module.my.export_LP (Note: This is just a name, not JNDI)
Inbound Connection Factory	mqjms.module	my/export	mqjms.module/my/export_LIS_CF
Response Connection Factory	mqjms.module	my/export	mqjms.module/my/export_RESP_CF
Receive	mqjms.module	my/export	mqjms.module/my/export_MQ_RECEIVE_D
Send	mqjms.module	my/export	mqjms.module/my/export_MQ_SEND_D
SIB Callback Destination	mqjms.module	my/export	mqjms.module/my/export_MQ_CALLBACK_D
SIB Callback Connection Factory	All modules	my/export	SCA.MQJMS/Callback_CF

Note:

- The resources are created at the server scope. The default scope in the administrative console is cell. You must change the scope in order to locate and administer the resources.
- The SIB callback destination and SIB callback connection factory are SIB JMS resources. The other entries in the table are MQ JMS resources. The two types of resources are administered.

In the alternate case, where the MQ JMS import or export is expecting to find on the server resources that it will use, you must have these resources installed and the import file must contain their JNDI names. The association between the MQ JMS import and the resources will then be made.

Configuring MQ JMS bindings

You can administer MQ JMS bindings to configure special features of the resource. The administrative tasks are performed using the administrative console.

Before you begin

You must have permission to make and save changes to the profile on the administrative console.

The queue and queue manager are not automatically generated; they must be created in WebSphere MQ by your WebSphere MQ administrator.

About this task

The MQ JMS import or export must be installed as part of a Service Component Architecture (SCA) application in your server profile.

Procedure

1. Select the SCA module. From the administrative console, click **Applications** → **SCA Modules** and then click the *modulename*. The configuration page opens.
2. Under **Module components**, expand **Imports** or **Exports**. This reveals a list of the installed imports or exports.
3. Expand the import or export and then expand **Binding**. This reveals a list of available bindings to administer.
4. Click on the binding to be administered. The general properties of the binding are displayed:
   - The **Send Resources** category contains the Connection Factory and the Send Destination.
   - The **Receive Resources** category contains the Response Connection Factory, the Listener Port, and the Activation Specification.
   - The **Advanced Resources** category contains Callback resources and other available resources.

   Note: You can also access a resource by typing the JNDI name in the text box. Doing so, however, allows you to enter the name of a resource that is not yet configured.
5. Administer the desired resource:
   a. Click the **Browse...** button to open a window with a list of JNDI names; then select the desired JNDI name. The selected name will populate the appropriate text field.
   b. Click the **Configure...** button to open the corresponding panel referred to by the JNDI name. While most resources can be configured at cluster scope, selecting the **Configure...** option at Listener Port displays a panel showing all listener ports with the cluster Listener Port names for all the members of the given cluster; you can then select one listener port.

      When **Configure...** has been selected, the corresponding WebSphere Application Server panel will open.
6. When you are ready to administer your settings, click **Apply** or **OK**.

What to do next

The application will be automatically restarted for the JNDI bindings to take effect.

JMS and WebSphere MQ JMS headers

JMS and WebSphere MQ JMS headers are Service Data Objects (SDO) that contain all the properties of the JMS (or WebSphere MQ JMS) message properties. These properties can be from the inbound message or they can be the properties that will be applied to the outbound message.

JMS and WebSphere MQ JMS headers are provided using the system programming interface; consequently, you set or get the headers by accessing the service component architecture message. You cannot set or get the header from a client program. You can, however, use an ESB mediation to retrieve the header values or set them on an outgoing request.

The properties that can be set on the JMSMessage Header are:
- **JMSType** and **JMSCorrelationID** – values of the specific predefined message header properties
- **JMSDeliveryMode** – values for delivery mode (Persistent or NonPersistent; default is Persistent)
- **JMSPriority** – priority value (0 to 9; default is JMS_Default_Priority)

The JMS and WebSphere MQ JMS headers and message properties are only used when the base service component architecture SCDL binding switch is turned on. When the switch is turned on, the JMS header information is propagated. By default, this switch is on. To prevent context information propagation, change the value to false.

When context propagation is enabled, header information is allowed to flow to the message or to the target component. To turn on and off context propagation, specify true or false for the contextPropagationEnabled attribute of the import and export bindings. For example:

```
<esbBinding xsi:type="eis:JMSImportBinding" contextProgagationEnabled="true">
```

The default is true.

> **Related concepts**
>
> "Generic JMS headers" on page 174
> Generic JMS headers are Service Data Objects (SDO) that contain all the properties of the generic JMS message properties. These properties can be from the inbound message or they can be the properties that will be applied to the outbound message.
>
> "WebSphere MQ headers" on page 190
> WebSphere MQ headers incorporate certain conventions for conversion to the service component architecture (SCA) messages.

External clients

The server can send messages to, or receive messages from, external clients using WebSphere MQ JMS bindings.

An external client, for example a web portal or an EIS, needs to send a message to an SCA module in the server or needs to be invoked by a component from within the server.

The WebSphere MQ JMS export components deploy message driven beans (MDBs) to listen to requests incoming to the receive destination specified in the export binding. The destination specified in the send field is used to send the reply to the

inbound request if the invoked application provides a reply. Thus an external client is able to invoke applications via the export binding.

WebSphere MQ JMS imports bind to and can deliver message to external clients. This message may or may not demand a response from the external client.

More information on how to interact with external clients using WebSphere MQ can be found at the WebSphere MQ information center.

WebSphere MQ bindings

The WebSphere MQ application provides Service Component Architecture connectivity with WebSphere MQ applications.

Use the WebSphere MQ provider applications to integrate directly with a WebSphere MQ-based system from your server environment. This eliminates the need to use MQ Link or Client Link features of the Service Integration Bus.

Importing *from* WebSphere MQ utilizes a destination that the data will be sent to and a destination where the reply can be received. Conversion for the data to and from an MQ message is accomplished through the MQ header and body data binding.

Exporting *to* WebSphere MQ utilizes a destination where the request can be received and the response can be sent. The conversion for the data to and from an MQ message is done through the MQ header and body data binding. The function selector provides a mapping to the operation on the target component to be invoked.

Related concepts

"WebSphere MQ JMS bindings" on page 174
The WebSphere MQ JMS binding provides integration with WebSphere MQ JMS-based providers.

WebSphere MQ bindings: a general perspective

The WebSphere MQ application provides integration with native MQ-based applications.

WebSphere MQ administrative tasks

The WebSphere MQ system administrator is expected to create the underlying WebSphere MQ Queue Manager, which the WebSphere MQ bindings will use before running an application containing these bindings.

WebSphere MQ import bindings

The WebSphere MQ import binding provides outbound connectivity from Service Component Architecture (SCA) applications to WebSphere MQ-based applications. You must be using a supported version of WebSphere MQ. Detailed hardware and software requirements can be found on the IBM support pages.

Two types of WebSphere MQ import bindings are supported:

- One-way: a message is sent to the destination specified as send in the import file. Nothing is sent to the replyTo field of the MQMD header.

- Two-way (request-response): a message is sent to the destination specified as send in the import file. The destination specified as receive is set in the replyTo MQMD header field. A message-driven bean (MDB) is deployed to listen on the receive destination, and when a reply is received the MDB invokes the callback object. For request-response scenarios, the response is expected to have the correlationId set to the sent message messageID for the default correlation scheme.

Figure 14 illustrates how the import is linked to the external service.
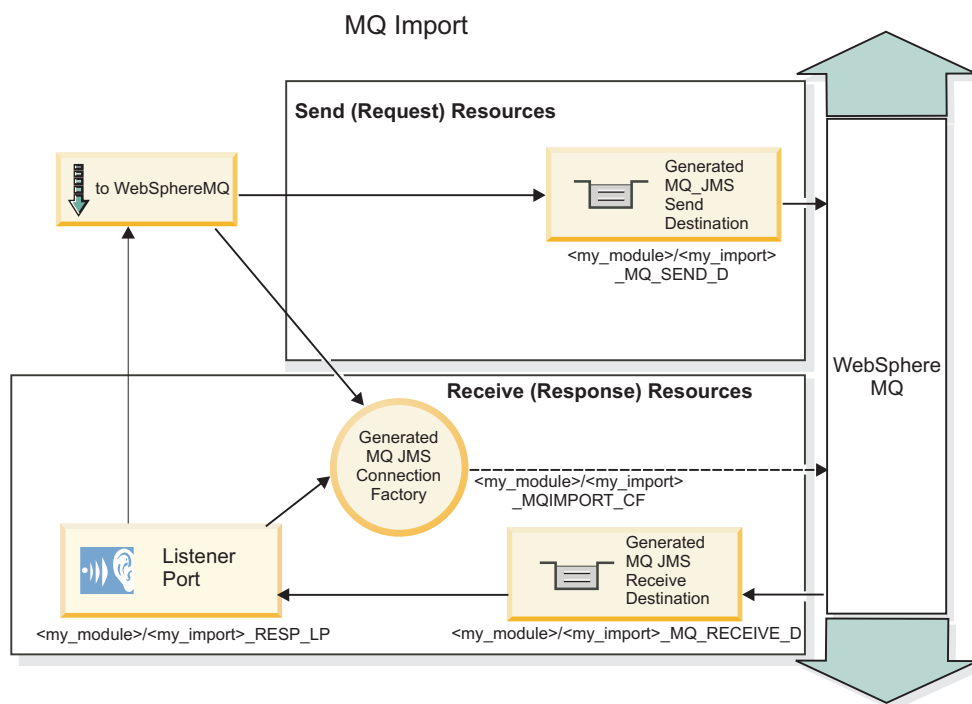
Figure 14. WebSphere MQ import binding resources

WebSphere MQ export bindings

The WebSphere MQ export binding provides inbound connectivity from WebSphere MQ-based applications to the SCA system.

An MDB is deployed to listen to requests incoming to the receive destination specified in the export binding. The destination specified in the send field is used to send the reply to the inbound request if the invoked application provides a reply. The destination specified in the replyTo field of the response message overwrites the destination specified in the send field.

Figure 15 on page 185 illustrates how the external requestor is linked to the export.
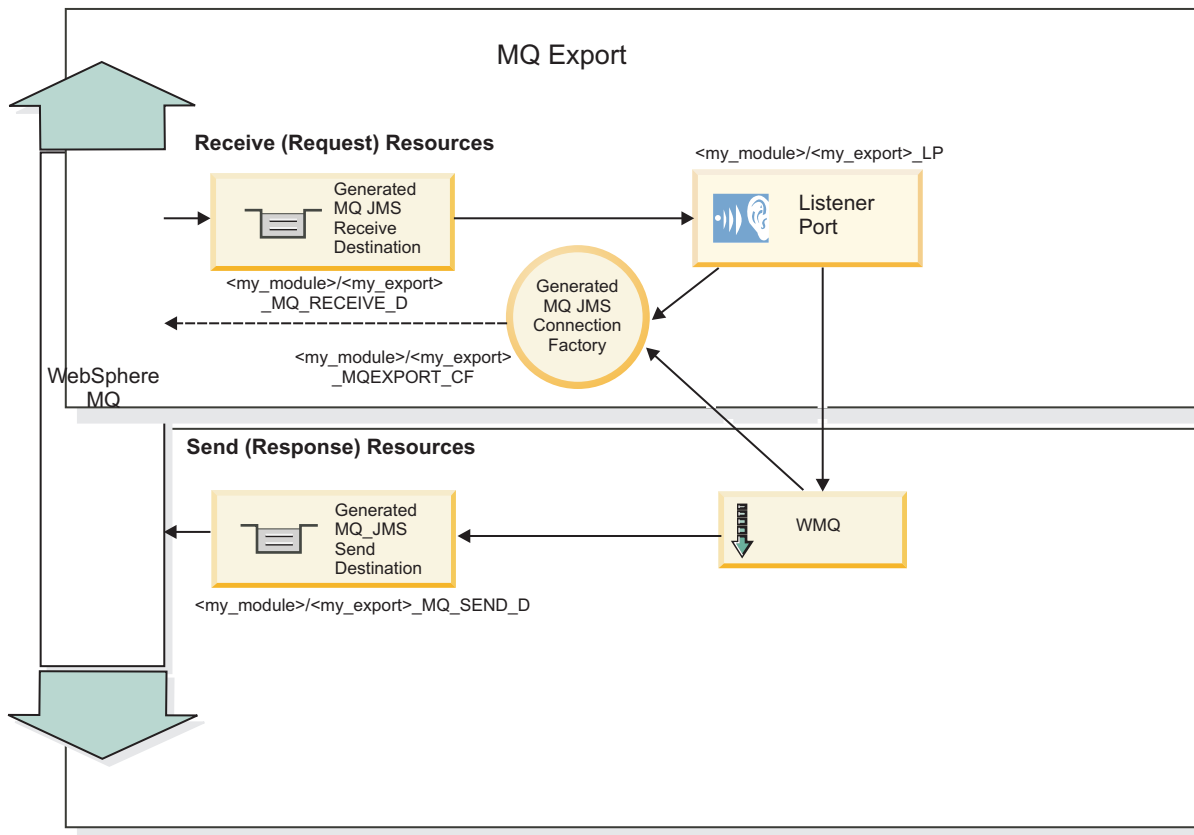
Figure 15. WebSphere MQ export binding resources

WebSphere MQ bindings support

The WebSphere MQ application provides integration with native MQ-based applications.

Use the WebSphere MQ applications when you want to integrate directly with a native MQ-based system within a J2EE programming environment. This eliminates the need for MQ Link or Client Link features of the Service Integration Bus.

Importing *from* WebSphere MQ utilizes a destination that the data will be sent to and a destination where the reply can be received. Conversion for the data to and from an MQ message is accomplished through the MQ header and body data bindings.

Exporting *to* WebSphere MQ utilizes a destination where the request can be received and the response can be sent. The conversion for the data to and from an MQ message is done through the MQ header and body data bindings. The function selector serves the purpose of providing a mapping to the operation on the target component to be invoked.

The configuration of WebSphere MQ Queue managers is left up to the WebSphere MQ system administrator. However, you must change the MQ_INSTALL_ROOT environment variable in WebSphere to the WebSphere MQ version supported by the server, and restart the server. This ensures that the libraries of a supported

version of WebSphere MQ are being used. Detailed hardware and software requirements can be found on the IBM support pages.

Related concepts

"WebSphere MQ JMS bindings support" on page 177
The WebSphere MQ JMS application provides integration with WebSphere MQ JMS-based providers.

Key features of a WebSphere MQ binding

Key features of a WebSphere MQ binding include headers, J2EE artifacts and created J2EE resources.

Correlation schemes

A WebSphere MQ request/reply application can use one of a number of techniques to correlate response messages with requests, built around the MQMD's MessageID and CorrelID fields. In the vast majority of cases, the requestor lets the queue manager select a MessageID and expects the responding application to copy this into the CorrelID of the response. In most cases, the requestor and responding application implicitly know which correlation technique is in use. Occasionally the responding application will honor various flags in the Report field of the request that describe how to handle these fields.

Export bindings for WebSphere MQ messages can be configured with the following options:

Response MsgId options:

New MsgID
    Allows the queue manager to select a unique MsgId for the response (default).

Copy from Request MsgID
    Copies the MsgId field from the MsgId field in the request.

Copy from SCA message
    Sets the MsgId to be that carried in WebSphere MQ headers in the SCA response message; or let the queue manager define a new Id if the value does not exist.

As Report Options
    Inspects the Report field of the MQMD in the request for a hint as to how to handle the MsgId. The MQRO_NEW_MSG_ID and MQRO_PASS_MSG_ID options are supported, and behave like New MsgId and Copy from Request MsgID, respectively

Response CorrelId options:

Copy from Request MsgID
    Copies the CorrelId field from the MsgId field in the request (default).

Copy from Request CorrelID
    Copies the CorrelId field from the CorrelId field in the request.

Copy from SCA message
    Sets the CorrelId to be that carried in WebSphere MQ headers in the SCA response message; left blank if the value doesn't exist.

As Report Options
    Inspects the Report field of the MQMD in the request for a hint as to how to handle the CorrelId. The MQRO_COPY_MSG_ID_TO_CORREL_ID and

MQRO_PASS_CORREL_ID options are supported, and behave like Copy from Request MsgID and Copy from Request CorrelID, respectively

Import bindings for WebSphere MQ messages can be configured with the following options:

Request MsgId options:

New MsgID
Allows the queue manager to select a unique MsgId for the request (default)

Copy from SCA message
Sets the MsgId to be that carried in WebSphere MQ headers in the SCA request message; or let the queue manager define a new Id if the value doesn't exist.

Response correlation options:

Response has CorrelID copied from MsgId
Expects the response message to have a CorrelId field set as per the request's MsgId (default).

Response has MsgID copied from MsgId
Expects the response message to have a MsgId field set as per the request's MsgId.

Response has CorrelID copied from CorrelId
Expects the response message to have a CorrelId field set as per the request's CorrelId.

J2EE resources

A number of J2EE resources are created when a WebSphere MQ binding is deployed to a J2EE environment.

Parameters

MQ Connection Factory
Used by clients to create a connection to the MQ JMS provider.

Response Connection Factory
Used by the SCA MQ runtime when the send destination is on a different Queue Manager than the receive destination.

Listener Port
Specifies an association between a connection factory, a destination, and a message-driven bean. This enables deployed message-driven beans associated with the port to retrieve messages from the destination.

Destinations
- Send destination: where the request or outgoing message is sent (import); where the response message will be sent (export), if not superseded by the MQMD ReplyTo header field in the incoming message.
- Receive destination: where the response/request or incoming message should be placed.

Working with WebSphere MQ bindings

WebSphere MQ bindings can be administered from within the server.

Use the administrative console to access WebSphere MQ bindings.

Detailed instructions on creating WebSphere MQ imports and exports can be found in the WebSphere Integration Developer information center at **WebSphere Integration Developer** > **Developing integration applications** > **Accessing external services with messaging systems** > **WebSphere MQ > Working with MQ bindings**.

> **Related concepts**
>
> "Working with WebSphere MQ JMS bindings" on page 179
> WebSphere MQ JMS bindings can be administered in WebSphere Process Server.
>
> **Related tasks**
>
> "Enabling event sequencing: WebSphere MQ exports" on page 214
> WebSphere MQ export bindings support event sequencing. In order for events to be handled in the order in which they are received you must configure properties of the binding.

Parameters of WebSphere MQ bindings

The WebSphere MQ binding can be installed with all the necessary connection factories created, or it can be designed to point to a pre-configured set of artifacts on the server.

Typically, WebSphere MQ import and export bindings are created in WebSphere Integration Developer. At development time you have an option to create the connections and destinations required for the WebSphere MQ import at install time, i.e., when the component is installed on your server. Alternatively, you can specify the JNDI name of the resources on the server that you intend your WebSphere MQ binding to utilize.

Configuring the WebSphere MQ binding depends upon which option was selected.

In the case where new message provider resources are created (i.e., the resources are created on the server at install time), the resources will exist and can be located and administered using the administrative console. The JNDI names of the generated artifacts are described in the following tables.

Table 13. WebSphere MQ import: Names and JNDI names of resources created at installation on the server

Resource	Module name	Import name	Resource global JNDI name
Outbound Connection Factory	mq.module	my/import	mq.module/my/import_MQ_CF
Response Listener Port	mq.module	my/import	mq.module.my.import_RESP_LP (Note: This is only a name, not JNDI)
Response Connection Factory	mq.module	my/import	mq.module/my/import_RESP_CF
Send	mq.module	my/import	mq.module/my/import_MQ_SEND_D
Receive	mq.module	my/import	mq.module/my/export_MQ_RECEIVE_D
SIB Callback Destination	mq.module	my/import	mq.module/my/import_MQ_CALLBACK_D
SIB Callback Connection Factory	All modules	my/import	SCA.MQ/Callback_CF

Table 14. WebSphere MQ export: Names and JNDI names of resources created at installation on the server

Resource	Module name	Export name	Resource global JNDI name
Listener Port	mq.module	my/export	mq.module.my.export_LP (Note: This is just a name, not JNDI)
Inbound Connection Factory	mq.module	my/export	mq.module/my/export_LIS_CF
Response Connection Factory	mq.module	my/export	mq.module/my/export_RESP_CF
Receive	mq.module	my/export	mq.module/my/export_MQ_RECEIVE_D
Send	mq.module	my/export	mq.module/my/export_MQ_SEND_D
SIB Callback Destination	mq.module	my/export	mq.module/my/export_MQ_CALLBACK_D
SIB Callback Connection Factory	All modules	my/export	SCA.MQ/Callback_CF

Note:

- The resources are created at the server scope. The default scope in the administrative console is cell. You must change the scope in order to locate and administer the resources.
- The SIB Callback Destination and SIB Callback Connection Factory are SIB JMS resources. The other entries in the table are WebSphere MQ JMS resources. The two types of resources are administered.

In the alternate case, where the WebSphere MQ binding is expecting to find resources on the server that it will use, you must have these resources installed and the import or export file must contain their JNDI names. The association between the WebSphere MQ binding and the resources will then be made.

Configuring WebSphere MQ bindings

You can administer WebSphere MQ import and export bindings to tune, or set, special features of the resource. The administrative tasks are performed using the administrative console.

Before you begin

You must have permission to make and save changes to the profile on the administrative console.

The queue and queue manager are not automatically generated and must be created in WebSphere MQ by your WebSphere MQ administrator.

About this task

The WebSphere MQ import must be installed as part of a Service Component Architecture (SCA) application in your server profile.

Procedure

1. Open the default messaging provider settings panel in the administrative console.

   Expand JMS Providers, click WebSphere MQ.

2. Optional: Administer WebSphere MQ connection factories.

   Click on WebSphere MQ connection factory, in the list of additional properties. This panel shows a list of WebSphere MQ connection factories with a summary of their configuration properties. Click on the MQ connection factory that you want to administer, or click New to create a new connection factory.

   Use the subsequent panel to browse or change the configuration properties of the selected connection factory for use with WebSphere MQ as a JMS provider. These configuration properties control how connections are created to associated queues.

   You set these properties in the bindings for the resource reference of the application. If you do not want to modify the bindings for an existing application, locate this connection factory in the J2C panels where you can find these properties.

3. Optional: Administer WebSphere MQ queue connection factories.

   Click on WebSphere MQ queue connection factories, in the list of addition properties. This panel shows a list of WebSphere MQ queue connection factories with a summary of their configuration properties. Click on the WebSphere MQ queue connection factory that you want to administer, or click New to create a new queue connection factory.

   Use the subsequent panel to browse or change the configuration of the selected queue connection factory for use with the WebSphere MQ JMS provider. These configuration properties control how connections are created to associated queues.

   A WebSphere MQ queue connection factory is used to create JMS connections to queues provided by WebSphere MQ for point-to-point messaging. Use WebSphere MQ queue connection factory administrative objects to manage queue connection factories for the WebSphere MQ JMS provider.

4. Optional: Administer WebSphere MQ queue destinations.

   Click on WebSphere MQ queue destinations, in the list of additional properties. This panel shows a list of the WebSphere MQ queue destinations with a summary of their configuration properties. Click on the queue destination that you want to administer, or click New to create a new WebSphere MQ queue destination.

   Use the subsequent panel to browse or change the configuration properties of the selected queue destination for point-to-point messaging with WebSphere MQ as a messaging provider.

   A WebSphere MQ queue destination is used to configure the properties of a queue. Connections to the queue are created by the associated queue connection factory for WebSphere MQ as a messaging provider.

5. Save your changes to the master profile and if necessary restart the server.

WebSphere MQ headers

WebSphere MQ headers incorporate certain conventions for conversion to the service component architecture (SCA) messages.

The representation of WebSphere MQ headers in the SCA model incorporates the following conventions:

- WebSphere MQ messages consist of a system header (the MQMD) and a message body.

- The message body can contain zero or more message headers. If multiple message headers exist in the message, the order of the headers is significant.
- Each header contains information describing the structure of the following header or, for the last header, of the body. The MQMD describes the first header or, if there is no message header, the body.

WebSphere MQ headers use the following attributes:
- MQMD represents the contents of the WebSphere MQ message description, except for information determining the structure and encoding of the body.
- MQControl contains information determining the structure and encoding of a message header or body.
- MQHeader is a container of header data that extends MQControl but contains a value element of anyType.
- MQHeaders contain a list of MQHeader objects.

Three SCA message headers are defined:
- An SCA header containing an MQMD DataObject
- An SCA header containing an MQControl DataObject, describing the st5ructure and encoding of the contents of the message body
- An optional SCA header containing an MQHeaders DataObject, a list of MQHeader DataObjects.

  Related concepts

  "JMS and WebSphere MQ JMS headers" on page 164
  JMS and WebSphere MQ JMS headers are Service Data Objects (SDO) that contain all the properties of the JMS (or WebSphere MQ JMS) message properties. These properties can be from the inbound message or they can be the properties that will be applied to the outbound message.

  "Generic JMS headers" on page 174
  Generic JMS headers are Service Data Objects (SDO) that contain all the properties of the generic JMS message properties. These properties can be from the inbound message or they can be the properties that will be applied to the outbound message.

Service Component Architecture modules and WebSphere MQ

SCA modules and WebSphere MQ queues can be connected to provide services to one another.

Service Component Architecture (SCA) modules can communicate with WebSphere MQ applications much in the same way as they do other SCA modules. A module that wants to send requests to a WebSphere MQ application uses an import configured with the correct response and request queues associated with that application. Similarly, an SCA module can provide services to a WebSphere MQ application using an export configured with the appropriate application request and response queues. You define the connections between the SCA modules and the WebSphere MQ queues when you build your module.

From the WebSphere MQ queue manager perspective, the SCA module looks as if it were a normal MQ client. From the SCA module end, the WebSphere MQ queue looks like any other service. You can even further shield the SCA module from the WebSphere MQ queues by using a mediation module between the SCA module and the WebSphere MQ queue and let the mediation transform the original SCA request to the correct format for the target queue and handle the response when it becomes available.

Restrictions: When configuring WebSphere MQ for imports and exports, keep in mind the following:

- You must configure queue destinations with the target client set to **MQ**.
- You must use a TCP/IP client connection to connect to WebSphere MQ. See WebSphere MQ Intercommunication for details.
- You cannot use channel compression.
- You must set any necessary channel exits for correct data handling: click **WebSphere MQ messaging provider** → **WebSphere MQ connection factory objects** → **Custom Properties** and set the following custom properties:
  - **SENDEXIT** to the value com.ibm.ws.sca.internal.mq.exit.MQInternalSendExitImpl.
  - **RECEXIT** to the value com.ibm.ws.sca.internal.mq.exit.MQInternalReceiveExitImpl.
  - **SENDEXITINIT** to a text string of your choice. Setting this property helps you identify the exit in a trace.
  - **RECEXITINIT** to a text string of your choice. Setting this property helps you identify the exit in a trace.
- You cannot use the same connection factories for WebSphere MQ and Java Message Service (JMS) MQ bindings as the exits will inhibit the JMS MQ bindings.

External clients

WebSphere Process Server can send messages to, or receive messages from, external clients using WebSphere MQ bindings.

An external client, for example a web portal or an EIS, needs to send a message to an SCA module in the server or needs to be invoked by a component from within the server.

The WebSphere MQ export components deploy message driven beans (MDBs) to listen to requests incoming to the receive destination specified in the export binding. The destination specified in the send field is used to send the reply to the inbound request if the invoked application provides a reply. Thus an external client is able to invoke applications via the export binding.

WebSphere MQ imports bind to, and can deliver message to, external clients. This message might or might not demand a response from the external client.

More information on how to interact with external clients using WebSphere MQ can be found at the WebSphere MQ information center.

HTTP bindings

The HTTP binding is designed to provide Service Component Architecture (SCA) connectivity to HTTP. This allows existing or newly-developed HTTP applications to participate in Service Oriented Architecture (SOA) environments.

In addition, a network of SCA runtime environments can communicate across an existing HTTP infrastructure.

The HTTP binding exposes several HTTP features:

- Messages are presented to mediation components in a manner that preserves HTTP format and message header information. This provides a more familiar view to HTTP application programmers, users and administrators.
- An existing data binding framework is extended for HTTP conventions and provides mapping between SCA messages and HTTP message headers and bodies.
- Imports and exports can be configured to support a range of common HTTP features.
- When you install an SCA module containing HTTP imports or exports, the runtime environment is automatically configured appropriately to allow connectivity to HTTP.

Detailed instructions on creating HTTP imports and exports can be found in the information center at **WebSphere Integration Developer** > **Developing integration applications** > **HTTP data binding**.

Related tasks

"Working with imports" on page 282
You can list the imports of mediation modules that have been deployed to WebSphere Process Server. You can also display import interfaces and change the details of import bindings.

"Working with exports" on page 285
You can list the exports of mediation modules that have been deployed to WebSphere Process Server. You can also display export interfaces and export bindings.

Displaying HTTP bindings

After deploying an application, you may want to examine the HTTP bindings to make sure they are correct.

Before you begin

You must be at the administrative console.

Required security role for this task: When security and role-base authorization are enabled, you must log in as an operator, administrator or configurator to perform this task.

About this task

Display HTTP bindings as the first step in changing any existing HTTP bindings.

Procedure
1. Display the Service Component Architecture (SCA) applications. Expand **Applications** and click **SCA Modules**.
   The system displays the installed applications.
2. Click the *module_name* for which you want to display the bindings.
   The system displays the module configuration.
3. Expand **Imports** or **Exports** depending on which binding you are displaying.
4. Expand the *path* and then expand **Bindings** to display the bindings for the imports or exports contained by the module.
   The system displays all of the contained import or export bindings.
5. Click the binding name to display the binding configuration.

Results

The system displays the configuration of the binding for the import or export you selected.

What to do next

Change the import or export, if needed.

Changing HTTP export bindings

The administrative console allows you to change the configuration of HTTP export bindings without changing the original source and then redeploying the application.

Before you begin

Display the HTTP bindings for the module as described in "Displaying HTTP bindings."

Required security role for this task: When security and role-base authorization are enabled, you must log in as an administrator or configurator to perform this task.

About this task

Change HTTP export bindings when you need to change whether a method on a binding is pingable or to change the encodings a method or binding supports.

Procedure

1. Display the module bindings as described in "Displaying HTTP bindings."
2. Select the binding you are changing. Click on the binding name.
3. Change the binding configuration.

Option	Description
To change the configuration on the binding scope	Click the **Binding Scope** tab.
To change the configuration at the method scope	Click the **Method Scope** tab.

   When both configurations exist, the method scope configuration takes precedence over the binding scope configuration.
4. Change the configuration and click **Apply**, to remain on the same page, or **OK**, to return to the previous page.

Results

The configuration for the binding is changed.

Restriction: If the module is redeployed, the configuration is replaced with the new configuration. To ensure this change remains with the module across deployments, you must make the change in the source code for the module using WebSphere Integration Developer.

Changing HTTP import bindings

The administrative console allows you to change the configuration of HTTP import bindings without changing the original source and then redeploying the application.

Before you begin

Display the HTTP bindings for the module as described in "Displaying HTTP bindings."

Required security role for this task: When security and role-base authorization are enabled, you must log in as an administrator or configurator to perform this task.

About this task

Change HTTP import bindings when the binding properties of an HTTP application used by a Service Component Architecture (SCA) module change.

Procedure

1. Display the module bindings as described in "Displaying HTTP bindings."
2. Select the binding you are changing. Click on the binding name.
3. Change the binding configuration.

Option	Description
To change the configuration on the binding scope	Click the **Binding Scope** tab.
To change the configuration at the method scope	Click the **Method Scope** tab.

When both configurations exist, the method scope configuration takes precedence over the binding scope configuration.

4. Change the configuration and click **Apply**, to remain on the same page, or **OK**, to return to the previous page.

Results

The configuration for the binding is changed.

Restriction: If the module is redeployed, the configuration is replaced with the new configuration. To ensure this change remains with the module across deployments, you must make the change in the source code for the module using WebSphere Integration Developer.

HTTP headers

HTTP import and export bindings allow configuration of HTTP headers and their values to be used for outbound messages. The HTTP import uses these headers for requests, and the HTTP export uses them for responses.

Statically configured headers and control information take precedence over values dynamically set at runtime. However, the dynamic override URL, Version, and Method control values override the static values, which are otherwise considered defaults.

The binding supports dynamicity of the HTTP import URL by determining the value of HTTP target URL, Version and Method at runtime. These values are determined by extracting the value of the dynamic override URL, Version and Method from HTTP control parameters section of Service Component Architecture (SCA) message.

The control and header information for outbound messages under HTTP export and import bindings is processed in the following order:

1. Header and control information excluding HTTP dynamic override URL, Version and Method from the SCA Message. (lowest priority)
2. Simple Configuration Definition Language (SCDL) instance export/import level
3. Changes from the administrative console on the export/import level
4. SCDL instance export/import method level
5. Changes from the administrative console on the method level of the export/import
6. HTTP dynamic override URL, Version and Method from the Service Component Architecture (SCA) message
7. Headers and control information from the data binding (highest priority)

The HTTP export and import will only populate inbound direction headers and control parameters with data from the incoming message (HTTPExportRequest and HTTPImportResponse) if contextPropagationEnabled is set to True. Inversely, the HTTP export and import will only read and process outbound headers and control parameters (HTTPExportResponse and HTTPImportRequest) if contextPropagationEnabled is set to True. Configured headers and control parameters from SCDL will apply regardless of the value of contextPropagationEnabled property.

Supplied HTTP header structures and support

Table 15 itemizes the request/response parameters for HTTPImport and HTTPExport requests and responses.

Table 15. Supplied HTTP header information

Control name	HTTPImport request	HTTPImport response	HTTPExport request	HTTPExport response
URL	Ignored	Not set	Read from the request message. **Note:** Query string is also part of the URL control parameter.	Ignored
Version (possible values: 1.0, 1.1; default is 1.1)	Ignored	Not set	Read from the request message	Ignored
Method	Ignored	Not set	Read from the request message	Ignored

Table 15. Supplied HTTP header information (continued)

Control name	HTTPImport request	HTTPImport response	HTTPExport request	HTTPExport response
Dynamic Override URL	If set, overrides the HTTP Import URL. Written to the message in request line. **Note:** Query string is also part of the URL control parameter.	Not set	Not set	Ignored
Dynamic Override Version	If set, overrides the HTTP Import Version. Written to the message in request line.	Not set	Not set	Ignored
Dynamic Override Method	If set, overrides the HTTP Import Method. Written to the message in request line.	Not set	Not set	Ignored
Media Type (This control parameter carries part of the value of Content-Type HTTP header.)	If present, written to the message as part of Content-Type header. **Note:** This control element value should be provided by data binding.	Read from the response message, Content-Type header	Read from the request message, Content-Type header	If present, written to the message as part of Content-Type header. **Note:** This control element value should be provided by data binding.
Charset (default: UTF-8)	If present, written to the message as part of Content-Type header. **Note:** This control element value should be provided by data binding.	Read from the response message, Content-Type header	Read from the request message, Content-Type header	Supported; written to the message as part of Content-Type header. **Note:** This control element value should be provided by data binding.
Transfer Encoding (Possible values: chunked, identity; default is identity)	If present, written to the message as a header and controls how the message transformation is encoded.	Read from the response message	Read from the request message	If present, written to the message as a header and controls how the message transformation is encoded.

Table 15. Supplied HTTP header information (continued)

Control name	HTTPImport request	HTTPImport response	HTTPExport request	HTTPExport response
Content Encoding (Possible values: gzip, x-gzip, deflate, identity; default is identity)	If present, written to the message as a header and controls how the payload is encoded.	Read from the response message	Read from the request message	If present, written to the message as a header and controls how the payload is encoded.
Content-Length	If present, written to the message a header (HTTP header value ignored) **Note:** This should be set by data binding in the control parameters.	Read from the response message	Read from the request message	If present, written to the message a header (HTTP header value ignored) **Note:** This should be set by data binding in the control parameters.
StatusCode (default: 200)	Not supported	Read from the response message	Not supported	If present, written to the message in response line
ReasonPhrase (default: OK)	Not supported	Read from the response message	Not supported	Control value ignored. The message response line value is generated from the StatusCode.
Authentication (contains multiple properties)	If present, used to construct Basic Authentication header. **Note:** The value for this header will be encoded only on the HTTP protocol. In the SCA, it will be decoded and passed as clear text.	Not applicable	Read from the request message Basic Authentication header. The presence of this header does not indicate the user has been authenticated. Authentication should be controlled in the servlet configuration. **Note:** The value for this header will be encoded only on the HTTP protocol. In the SCA, it will be decoded and passed as clear text.	Not applicable

Table 15. Supplied HTTP header information (continued)

Control name	HTTPImport request	HTTPImport response	HTTPExport request	HTTPExport response
Proxy (contains multiple properties: Host, Port, Authentication)	If present, used to establish connection through proxy.	Not applicable	Not applicable	Not applicable
SSL (contains multiple properties : Keystore, Keystore Password, Trustore, Trustore Password, ClientAuth)	If populated and destination url is HTTPS, it is used to establish connection through SSL.	Not applicable	Not applicable	Not applicable

HTTP function selectors

A function selector is used to identify a "native" operation name, which in turn identifies the method binding. You can use one of the two supplied function selectors, or you can use a custom function selector by supplying a java implementation of the HTTPFunctionSelector interface.

The FunctionSelector uses information from the HTTP headers, from the invoked HTTP export URL, and in some cases from the HTTP message payload.

When the HTTP export receives a request message, it uses the function selector to determine the "native" method name, i.e., the HTTP identifier(s) for the method being invoked. Method-specific binding configuration (known as a "method binding") is then used to determine the name of the operation to invoke, the format (i.e., the data binding class) to use for the request, and the format (i.e., the data binding class) to use for any response.

The HTTP import also uses method bindings to determine the request and (optionally) response data bindings. There is a method binding for each operation on the import's interfaces.

When constructing a HTTP message on behalf of an import or export, the data binding has access to the header Data Objects and has full control over these objects. When reading an inbound HTTP message, both the FunctionSelector and data binding have full access to the list of HTTP headers; any updates will be reflected in the message passed into SCA.

The two function selectors that are provided are:
- HTTP Export Context Path + HTTP Method

  Since a combination of an HTTP Export Binding Context path (part of the received request on the HTTP export) and the HTTP method is unique for a given servlet instance, the function selector will be able to use that as the native operation name which in turn will identify the method binding.

  For example, if the HTTP export receives a request with url as "http://www.ibm.com/store/shoppingCart" and request HTTP method as "POST", the above function selector will extract "/store/shoppingCart" as the

export context path and append "POST" with "@" character as the delimiter. The resulting native method that the export will be looking for would need to be "/store/shoppingCart@POST". Please note that export context path is configured at design time and not necessarily will be equal to the relative path of the request url.

- HTTP Header

  This function selector reads the native operation from the TargetFunctionName HTTP custom header.

The binding relies on the fact that all native method names in the export are unique so during runtime, it is enough to find the first match to determine the method binding. Validators will ensure that native methods are unique.

Implementing the custom HTTP function selector

In order to implement custom function selector, extend the HTTPFunctionSelector abstract class and implement the generateEISFunctionName method.

HTTP data bindings

For each different mapping of data between a Service Component Architecture (SCA) message and HTTP protocol message, an HTTP data binding must be configured. Three data binding classes are supplied; you can also write custom data bindings.

Three data bindings are provided for use with HTTP imports and HTTP exports: binary data binding, XML data binding, and SOAP data binding. Response data binding is not required for the one-way operations.

A data binding is represented by the name of a Java class whose instances can convert both from HTTP to ServiceDataObject and vice versa. A function selector must be used on an export which, in conjunction with method bindings, can determine which data binding is used and which operation is invoked.

The supplied data bindings are:

- Binary data binding treats the body as unstructured binary data. The implementation of binary data binding XSD schema is as follows:

```
<xsd:schema elementFormDefault="qualified"
  targetNamespace="http://com.ibm.websphere.http.data.bindings/schema"
  xmlns:tns="http://com.ibm.websphere.http.data.bindings/schema"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:complexType name="HTTPBaseBody">
    <xsd:sequence/>
  </xsd:complexType>

  <xsd:complexType name="HTTPBytesBody">
    <xsd:complexContent>
      <xsd:extension base="tns:HTTPBaseBody">
        <xsd:sequence>
          <xsd:element name="value" type="xsd:hexBinary"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
```

- XML data binding supports the body as XML data. The implementation of XML data binding is similar to JMS XML data binding, and has no restrictions on the interface schema.

- SOAP data binding supports the body as SOAP data. The implementation of SOAP data binding has no restrictions on the interface schema.

Implementing custom HTTP data binding

HTTPStreamDataBinding is the principle interface for handling custom HTTP messages. The interface is designed to allow handling of large payloads. However, in order for such implementation to work, this Data Binding must return the control information and headers before writing the message into stream.

The methods and their order of execution, listed below, must be implemented by the custom data binding.

HTTP binding will invoke Data Binding in the following order:

Outbound processing (SDO to Native format):
1. setDataObject(...)
2. setHeaders(...)
3. setControlParameters(...)
4. setBusinessException...)
5. convertToNativeData()
6. getControlParameters()
7. getHeaders()
8. write(...)

Inbound processing (Native format to SDO):
1. setControlParameters(...)
2. setHeaders(...)
3. convertFromNativeData(...)
4. isBusinessException()
5. getDataObject()
6. getControlParameters()
7. getHeaders()

Data bindings

A data binding describes mapping between data passed to the SCA application and the enterprise information system (EIS). In other words, it describes binding of the SCA data representation to the EIS-specific form.

Detailed instructions on working with data bindings can be found in the WebSphere Integration Developer information center at **WebSphere Integration Developer** > **Developing integration applications** > **Working with configurations that access external services** > **Data binding configuration**.

Data bindings for EIS application

Data bindings allow EIS import or EIS export implementation to convert argument data to the native format expected by the J2C or JMS implementation.

You can define the data binding implementation class or classes in the import or export files.

The DataBinding is the root data binding interface. It allows for setting of a data object and is applicable to both J2C and JMS.

Predefined data bindings for EIS applications

You can create your own data bindings and data handlers, but several are provided as part of the product.

A data binding takes an incoming stream of data in a native format and converts it to a business object to be used by services created with the service component architecture.

A data handler works with a data binding. The data binding delegates the transformation logic to the data handler. The data binding works as a mediator between the transport-dependent protocol (for example, JMS or MQ) and the transport-independent protocol of the data handler.

The following data bindings are predefined when using data bindings for the Email, FTP and Flat Files adapters:

Table 16. Predefined data bindings

Data binding	Description
EmailSimpleDataBinding	This data binding serializes the business object to a simple e-mail data type. It cannot be used with a data handler. It is used for simple email alerts.
EmailFixedStructure DataBinding	This data binding serializes the business object to an Email adapter user defined type. It can be used with a data handler. This data binding corresponds to a business object structure that you define. Email attachments must be in the same order as the attributes in the business object.
EmailWrapperDataBinding	This data binding serializes the business object to a generic e-mail data type or generic e-mail with business graph data type. It can optionally be used with a data handler. It is used with generic email and generic email with business graph data types.
FlatFileBaseDataBinding	This data binding serializes the business object to the format expected by the Flat Files adapter. It is configurable with a data handler. It does not need to be configured for some uses such as a passthrough scenario.
FTPFileBaseDataBinding	This data binding serializes the business object to the format expected by the FTP adapter. It is configurable with a data handler. It does not need to be configured for some uses such as a passthrough scenario.
WTX data binding	This data binding lets you use WebSphere Transformation Extender, a universal validation and transformation engine, to convert business objects to many data formats and vice versa.

For other data formats, you must employ custom data bindings and data handlers.

The following data handlers are predefined for use with data bindings for the Email, FTP and Flat Files adapters:

Table 17. Predefined data handlers

Data Handler	Description
Delimited Data Handler	This data handler should be used when you have serialized data in a delimited format like comma separated format as input into an export or need to generate serialized data in a delimited format for an import.
FixedWidth Data Handler	This data handler should be used when you have serialized data in a fixed width format as input into an export or need to generate serialized data in a fixed width format for an import.
JSON Data Handler	This data handler should be used when you have serialized data in a JavaScript Object Notation (JSON) format as input into an export or need to generate serialized data in a JSON format for an import.
WTX MapSelection Data Handler	This data handler lets you use WebSphere Transformation Extender, a universal validation and transformation engine, to convert business objects to many data formats and vice versa. This data handler should be used if you are creating a configuration that makes use of the Email, FTP and Flat Files adapters.
XML Data Handler	This data handler should be used when you have serialized XML data as input into your export (for inbound processing at runtime) or need to generate serialized XML data for an import (for outbound processing at runtime).

JCA 1.5 record-based data bindings

You can define the data binding implementation class or classes in the import or export files. Specific interfaces must be implemented if a record-based data binding is used.

The RecordDataBinding interface extends the basic interface and adds to it the javax.resource.cci.Record. If the data binding is to be used with the J2EE connector architecture-based service, it must implement this interface. As a record, it can be passed to the execute method of the resource adapter interaction with input and output.

This interface is not sufficient if the resource adapter supports an input-only variant of the execute method on the interaction. This limitation arises because an input-only execution returns the result as the CCI Record. In this instance the record needs to be set on the data binding and then the converted data object can be retrieved. If the resource adapter supports the input-only variant of the execute method, the data binding provider needs to implement the RecordHolderDataBinding interface.

You must also implement the RecordHolderDataBinding interface for an inbound J2C communication. The listener interface is invoked by the resource adapter and the MDB implementing it is passed as native data. This data needs to be set on the data binding to be able to retrieve from it business object. If the listener argument and return are typed as javax.resource.cci.Record, the RecordHolderDataBinding is sufficient. If the listener argument(s) or return type is other than CCI Record, the utility interface InboundNativeDataRecord is provided.

Both RecordDataBinding and RecordHolderDataBinding can support three-argument invocation. Whether the resource adapter requires two or three

arguments is defined in the ResourceAdapterMetadata, which is queried at
runtime to determine how many arguments the RecordDataBinding and
RecordHolderDataBinding require

An arbitrary number and types of arguments to the listener can be set on the
InboundNativeDataRecord interface, as an object array. The implementation, a CCI
Record, can then be set on the RecordHolderDataBinding interface thus passing all
the listener arguments and allowing the data binding to retrieve arguments and
create the business object. On the return, the data binding implementation sets the
return value in the symmetric manner. The runtime then uses the get method, also
returning an object array; the argument at index 0 is the return value to be
returned to the invoker (the resource adapter).

Data binding for JMS, generic JMS, and WebSphere MQ JMS applications

JMS messages use data binding classes to map between JMS messages and the
internal data representation (service data object).

Six general-purpose data bindings that support any message body are supplied for
the JMS, generic JMS and WebSphere MQ JMS applications: a defined message
class, which can have properties but no payload, and five sub-classes of message,
each of which has a different body type. The operation on the runtime interface
used to process a message is the FunctionSelector, which uses the JMSType
property of the message to select the operation name.

The five sub-classes of message, with their properties to carry the message
payload, are:

- TextMessage – the message body is a Java string. JMSTextDataBinding has one
  string property.
- BytesMessage – the message body is a byte array. JMSBytesDataBinding has one
  hexBinary property.
- ObjectMessage – the message body is a serialized Java object.
  JMSObjectDataBinding has one hexBinary property.
- StreamMessage – the message body is a stream (sequence) of simple Java types.
  JMSStreamDataBinding has a list of entry elements where each entry has a type
  and value property.
- MapMessage – the message body is a set of name/value pairs that can be
  addressed by *name* with the value being a simple Java type.
  JMSMapDataBinding has a list of entry elements where each entry has a name,
  type and value property.

The FunctionSelector is exposed at the import and export nodes. When an export
node has been created and an interface assigned to it, the bindings for it can be
generated. When JMS Binding is selected, a wizard will offer a drop-down
selection of three options at Serialization type: Object, Text, and User Supplied.

Clear the check box at JMS function selector to type in the class name of the new
user-supplied function selector in the text box.

The data bindings can also be selected or changed on the binding properties page
of the export or import.

Predefined data bindings and data handlers for JMS, MQ JMS and generic JMS bindings

You can create your own data bindings and data handlers, but several are provided as part of the product.

A data binding takes an incoming stream of data in a native format and converts it to a business object to be used by services created with the service component architecture.

A data handler works with a data binding. The data binding delegates the transformation logic to the data handler. The data binding works as a mediator between the transport-dependent protocol (for example, JMS or MQ) and the transport-independent protocol of the data handler.

The following data bindings are predefined for use with JMS, MQ JMS or generic JMS bindings:

Table 18. Predefined data bindings

Data binding	Description
Business Object XML Using JMSTextMessage or JMS BytesMessage	This data binding serializes the business object to and from XML and uses a JMS text message or a message in bytes to communicate with the JMS client. All character encoding sets (CCSIDs) are supported.
JMS adapter language data binding generator	This data binding generator is needed if your message body contains a message in either C or COBOL or PL/I.
Serialized Business Object using JMSObjectMessage	This data binding sends and receives the business object, whose underlying implementation is a service data object (SDO) in a JMS object message, to and from the JMS client. This serialization type is preferred when dealing with other SOA components using JMS.
Simple JMS BytesMessage Data Binding	This data binding sends and receives the JMSBytesBody business object in a BytesMessage message body to the client. The message body type is a byte array.
Simple JMS MapMessage Data Binding	This data binding sends and receives the JMSMapBody business object in a MapMessage message body to the client. The message body type is a set of name/value pairs that can be addressed by name. The data type of the value is a Java primitive.
Simple JMS Message Data Binding	This data binding does not have a message body. It can be used to notify the JMS client of an event or to send some simple data in the message headers or properties.
Simple JMS ObjectMessage Data Binding	This data binding sends and receives the JMSObjectBody business object in an ObjectMessage message body to the client. The message body type is a serialized Java Object.
Simple JMS StreamMessage Data Binding	This data binding sends and receives the JMSStreamBody business object in a StreamMessage message body to the client. The message body type is a stream of Java primitives.
Simple JMS TextMessage Data Binding	This data binding sends and receives the JMSTextBody business object in a TextMessage message body to the client. The message body type is a String.
Delimited data binding for JMS	This data binding should be used when you have serialized data in a delimited format like comma separated format as input into an export or need to generate serialized data in a delimited format for an import.

Table 18. Predefined data bindings (continued)

Data binding	Description
FixedWidth data binding for JMS	This data binding should be used when you have serialized data in a fixed width format as input into an export or need to generate serialized data in a fixed width format for an import.
JSON data binding for JMS	This data binding should be used when you have serialized data in a JavaScript Object Notation (JSON) format as input into an export or need to generate serialized data in a JSON format for an import.
WTX data binding	This data binding lets you use WebSphere Transformation Extender, a universal validation and transformation engine, to convert business objects to many data formats and vice versa.

For other data formats, you must employ custom data bindings and data handlers.

The following data handlers are predefined for use with JMS, MQ JMS and generic JMS bindings:

Table 19. Predefined data handlers

Data Handler	Description
Delimited Data Handler	This data handler should be used when you have serialized data in a delimited format like comma separated format as input into an export or need to generate serialized data in a delimited format for an import.
FixedWidth Data Handler	This data handler should be used when you have serialized data in a fixed width format as input into an export or need to generate serialized data in a fixed width format for an import.
JSON Data Handler	This data handler should be used when you have serialized data in a JSON format as input into an export or need to generate serialized data in a JSON format for an import.
WTX MapSelection Data Handler	This data handler lets you use WebSphere Transformation Extender, a universal validation and transformation engine, to convert business objects to many data formats and vice versa.
XML Data Handler	This data handler should be used when you have serialized XML data as input into your export (for inbound processing at runtime) or need to generate serialized XML data for an import (for outbound processing at runtime).

Serialization

The JMS and MQ JMS applications can provide default implementations for data bindings that provide serialization of the DataObjects to and from JMS or MQ JMS formats.

The JMS application and export implementation is able to provide default implementations for the following JMS data bindings:

- JMSDataBindingImplJava – supports JMSObjectMessage and serializes the DataObject to or from the object field of the JMSMessage. The object contents of the message must implement the Serializable interface. The

JMSDataBindingImplJava will also implement the JMSObjectBinding which allows it to serialize java.lang.Object to the message.

- JMSDataBindingImplXML – supports JMSTextMessage. This binding serializes the DataObject into XML format and sets it to the text field of the JMSMessage. The implementation of the serialization is using native SDO object serialization exposed in the SCA implementation (class com.ibm.ws.sca.data.impl.DataBindingImplXML).

When an XML schema has a type defined without a global element, the JMS bindings (JMSDataBindingImplXML and JMSDataBindingImplJava) cannot resolve the type to an element.

Schema:

```
 <?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
 targetNamespace="http://www.ibm.com" xmlns:tns="http://www.ibm.com">
<!-- global element required but missing -->

<complexType name="Quote">
<sequence>
<element name="symbol" type="string"></element>
<element name="price" type="float"></element>
</sequence>
</complexType>

</schema>
```

If you receive an exception such as:

```
com.ibm.websphere.sca.ServiceRuntimeException:
caused by: java.lang.IllegalArgumentException:
{Quote}Quote is not corresponding to a global element.
```

Or

```
[8/25/06 10:20:40:938 PDT] 00000054 FFDC Z
com.ibm.ws.sca.databinding.impl.DataBindingImplXML
com.ibm.ws.sca.databinding.impl.DataBindingImplXML#002 Exception:
org.eclipse.emf.ecore.xmi.FeatureNotFoundException:
Feature 'Quote' not found. (sca:/dataObject.xml, 2, 126)
```

This could mean that you need to define a global element:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
 targetNamespace="http://www.ibm.com" xmlns:tns="http://www.ibm.com">

<element name="Quote" type="tns:Quote"></element> <!-- global element required -->

<complexType name="Quote">
<sequence>
<element name="symbol" type="string"></element>
<element name="price" type="float"></element>
</sequence>
</complexType>

</schema>
```

Data binding for the WebSphere MQ application

Data bindings are provided for a WebSphere MQ message as a set of header data bindings and four types of body data bindings.

Both body and header data bindings use the same API to read from, and write to, WebSphere MQ messages. The read() method passes an MQDataInputStream object, and the write() method passes an MQDataOutputStream objeect.

Header data bindings are chosen based on the format identifier in control information associated with the header for both reading and writing. When reading a WebSphere MQ message, if no header binding can process the message, the body binding is called. When writing a WebSphere MQ message, the body data binding can overwrite the format identifier written into the WMQ message.

You can select from four body data bindings for use with a WebSphere MQ import or export. The string and binary data bindings treat the body as unstructured textual and binary data. The other two body data bindings store a serialized form of the SCA message body in the native MQ message: one using an XML form and one using Java object serialization. When either of these body data bindings are used with an import, the import will include an MQRFH2 header in the outbound native MQ message with a property TargetFunctionName in the <usr> folder. This property will be set to the name of the invoked operation.

Use the FunctionSelector on the export to select a body data binding type. Four function selectors are provided: Two function selectors return format information as the native operation and can be used when the body data binding should be dependent upon format data. A third function selector reads the native operation from the TargetFunctionName property of the <usr> folder of an MQRFH2 header, allowing interoperability with JMS imports. A fourth function selector always returns handleMessage regardless of the contents of the WebSphere MQ message.

Predefined data bindings and data handlers for MQ bindings

You can create your own data bindings and data handlers, but several are provided as part of the product.

A data binding takes an incoming stream of data in a native format and converts it to a business object to be used by services created with the service component architecture.

A data handler works with a data binding. The data binding delegates the transformation logic to the data handler. The data binding works as a mediator between the transport-dependent protocol (for example, JMS or MQ) and the transport-independent protocol of the data handler.

The following data bindings are predefined for use with MQ bindings.

Table 20. Predefined data bindings

Data binding	Description
JMS adapter language data binding generator	This data binding generator is needed if your message body contains a message in either C or COBOL or PL/I.
MQ serialized business object XML	This data binding serializes a business object to an XML document and deserializes an XML document to a business object. Your wrapped data objects must be a complex type; they can not be a simple type.
MQ serialized Java object	This data binding serializes a business object to a Java object and deserializes a Java object to a business object.

Table 20. Predefined data bindings (continued)

Data binding	Description
MQ unstructured binary message	This data binding sets the incoming bytes into a business object property called "value" on the inbound message and gets the bytes from the business object property called "value" and sets it in the output stream on the outbound message.
MQ unstructured text message	This data binding sets the incoming text message into a business object property called "value" on the inbound message and gets the text message from the business object property called "value" and sets it in the output stream on the outbound message.
MQRFH Header Data Binding	This data binding contains fixed-size portions and variable-sized portions of information specified by the MQRFH header standard.
MQRFH2 Header Data Binding	This data binding contains fixed-size portions and variable-sized portions of information specified by the MQRFH2 header standard.
Delimited data binding for MQ	This data binding should be used when you have serialized data in a delimited format like comma separated format as input into an export or need to generate serialized data in a delimited format for an import.
FixedWidth data binding for MQ	This data binding should be used when you have serialized data in a fixed width format as input into an export or need to generate serialized data in a fixed width format for an import.
JSON data binding for MQ	This data binding should be used when you have serialized data in a JavaScript Object Notation (JSON) format as input into an export or need to generate serialized data in a JSON format for an import.
WTX data binding	This data binding lets you use WebSphere Transformation Extender, a universal validation and transformation engine, to convert business objects to many data formats and vice versa.

For other data formats, you must employ custom data bindings and data handlers.

The following data handlers are predefined for use with MQ bindings:

Table 21. Predefined data handlers

Data Handler	Description
Delimited Data Handler	This data handler should be used when you have serialized data in a delimited format like comma separated format as input into an export or need to generate serialized data in a delimited format for an import.
FixedWidth Data Handler	This data handler should be used when you have serialized data in a fixed width format as input into an export or need to generate serialized data in a fixed width format for an import.
JSON Data Handler	This data handler should be used when you have serialized data in a JSON format as input into an export or need to generate serialized data in a JSON format for an import.

Table 21. Predefined data handlers (continued)

Data Handler	Description
WTX MapSelection Data Handler	This data handler lets you use WebSphere Transformation Extender, a universal validation and transformation engine, to convert business objects to many data formats and vice versa.
XML Data Handler	This data handler should be used when you have serialized XML data as input into your export (for inbound processing at runtime) or need to generate serialized XML data for an import (for outbound processing at runtime).

Predefined data bindings and data handlers for HTTP bindings

You can create your own data bindings and data handlers, but several are provided as part of the product.

A data binding takes an incoming stream of data in a native format and converts it to a business object to be used by services created with the service component architecture.

A data handler works with a data binding. The data binding delegates the transformation logic to the data handler. The data binding works as a mediator between the transport-dependent protocol (for example, JMS or MQ) and the transport-independent protocol of the data handler.

The following data bindings are predefined for use with HTTP bindings.

Table 22. Predefined data bindings

Data binding	Description
HTTP bytes message data binding	This data binding sets the incoming bytes in a business object property called "value" on the inbound message and gets the bytes from the business object property called "value" and sets them in the HTTP output stream.
HTTP SOAP message binding	This data binding serializes a business object to a SOAP message and deserializes a SOAP message to a business object.
HTTP XML message binding	This data binding serializes a business object to an XML document and deserializes an XML document to a business object.
Delimited data binding for HTTP	This data binding should be used when you have serialized data in a delimited format like comma separated format as input into an export or need to generate serialized data in a delimited format for an import.
FixedWidth data binding for HTTP	This data binding should be used when you have serialized data in a fixed width format as input into an export or need to generate serialized data in a fixed width format for an import.
JSON data binding for HTTP	This data binding should be used when you have serialized data in a JavaScript Object Notation (JSON) format as input into an export or need to generate serialized data in a JSON format for an import.

Table 22. Predefined data bindings (continued)

Data binding	Description
WTX data binding	This data binding lets you use WebSphere Transformation Extender, a universal validation and transformation engine, to convert business objects to many data formats and vice versa.

For other data formats, you must employ custom data bindings and data handlers.

The following data handlers are predefined for use with HTTP bindings:

Table 23. Predefined data handlers

Data Handler	Description
Delimited Data Handler	This data handler should be used when you have serialized data in a delimited format like comma separated format as input into an export or need to generate serialized data in a delimited format for an import.
FixedWidth Data Handler	This data handler should be used when you have serialized data in a fixed width format as input into an export or need to generate serialized data in a fixed width format for an import.
JSON Data Handler	This data handler should be used when you have serialized data in a JSON format as input into an export or need to generate serialized data in a JSON format for an import.
WTX MapSelection Data Handler	This data handler lets you use WebSphere Transformation Extender, a universal validation and transformation engine, to convert business objects to many data formats and vice versa.
XML Data Handler	This data handler should be used when you have serialized XML data as input into your export (for inbound processing at runtime) or need to generate serialized XML data for an import (for outbound processing at runtime).

Export bindings and event sequencing

You can stipulate that your exports process and deliver messages in the order in which they are received if your export bindings are appropriately configured.

Event sequencing provides the ability to sequence incoming events to an SCA component. Export bindings are the entry points to the target SCA components. In order for event sequencing to be enabled, the exports must process and deliver messages in the same order in which those messages are received.

The specific settings required to permit event sequencing depend on the type of export binding you are using.

Enabling event sequencing: EIS exports

Event sequencing is supported for EIS export bindings. The activation specification of the binding must be appropriately configured.

About this task

WebSphere adapters provide a mechanism to allow ordered events by specifying an activation specification property; see the documentation for more details. For JCA 1.5 resource adapters, consult the specific provider documentation for details on how to configure the adapter to enable the ordered or sequencing of events.

In general, if event sequencing is required in a network deployment environment, the module that has the export should be moved to a standalone server or to a cluster that has only one active server that is enabled for high availability.

Related concepts

"SCA EIS bindings" on page 152
Service Component Architecture (SCA) enterprise information system (EIS) bindings provide connectivity between SCA components and external systems. This communication is mediated by EIS exports and EIS imports.

Enabling event sequencing: JMS exports

JMS export bindings support event sequencing. In order for events to be handled in the order in which they are received, you must configure properties of the binding.

Before you begin

Event sequencing for JMS export bindings is supported in a clustered environment only when the destinations are not partitioned. For event sequencing to function in a network deployment environment with clusters, there can be only one active messaging engine per cluster. For event sequencing in a network deployment environment with servers that are not in a cluster, each server can have an active messaging engine.

About this task

Event sequencing requires events to acquire a lock before being dispatched to the target component for execution. When execution is complete, the event releases the lock. If an event cannot acquire a lock, the execution of the invocation is suspended. If the event subsequently acquires a lock, it will be dispatched.

You declare that event sequencing is required on a particular method for a particular component by adding an event sequencing qualifier to the method in the component definition.

- The keySpecification attribute defines the key that will be used to identify the events that need to be sequenced.
- The parameter attribute specifies the parameter from which the key attributes will be extracted.
- The name attribute is the name of the parameter.
- The xpath attribute is applied to the parameter to extract a value that will be part of the key.

You must specify a parameter element for each parameter that is going to contribute to the key.

The **esadmin** command line utility can be used to list locks and delete locks, both active and queued. Usage takes the form **esAdmin** *[-h hostname] [-p soapPortNumber]* method *[parameters]*. An absent hostname defaults to *localhost*. An absent soapPortNumber defaults to 8880. When security is enabled, valid userId and password are required.

Usage of **esadmin -help** takes the form of **esAdmin** *[-h hostname] [-p soapPortNumber] [-u userId] [-p password]* method *[parameters].* Available methods are:

- listAll
- listlocks moduleName
- listLocks moduleName componentName
- listLocks moduleName componentName methodName
- deleteLocks moduleName
- unlock lockId

where lockId and sequenceId are integers, and moduleName, componentName and methodName are strings. The sequenceId appears in the output and reflects the order in which the lock requests are queued.

Attention: The *delete* command should be used with caution. Use the *unlock* command instead of the *delete* command where possible. The *delete* command performs a database delete. The module must be stopped before this command is executed. Restart of the module is required.

Event sequencing for a JMS export is enabled from WebSphere Integration Developer.

Procedure

1. Set the eventSequencing property of your WebSphere MQ JMS export binding to True.
2. Save the changes to your master configuration.

Results

Event sequencing is enabled for your binding.

Note: Removing the exception destination means that any failure will stop all incoming messages.

> **Related concepts**
> "JMS bindings" on page 158
> A Java Message Service (JMS) provider enables messaging based on the Java Messaging Service. It provides J2EE connection factories to create connections for JMS destinations.

Enabling event sequencing: WebSphere MQ JMS exports

WebSphere MQ JMS export bindings support event sequencing. In order for events to be handled in the order in which they are received, you must configure properties of the binding.

Before you begin

If event sequencing for WebSphere MQ JMS export bindings is required in a network deployment environment, the module that has the export should be moved to a standalone server or to a cluster that has only one active server which is enabled for high availability.

About this task

Event sequencing for a WebSphere MQ JMS export is enabled from WebSphere Integration Developer.

Procedure

1. Set the eventSequencing property of your WebSphere MQ JMS export binding to True.
2. Save the changes to your master configuration.

   Related concepts

   "WebSphere MQ JMS bindings" on page 174
   The WebSphere MQ JMS binding provides integration with WebSphere MQ JMS-based providers.

   "Working with WebSphere MQ JMS bindings" on page 179
   WebSphere MQ JMS bindings can be administered in WebSphere Process Server.

Enabling event sequencing: WebSphere MQ exports

WebSphere MQ export bindings support event sequencing. In order for events to be handled in the order in which they are received you must configure properties of the binding.

Before you begin

If event sequencing for WebSphere MQ export bindings is required in a network deployment environment, the module that has the export should be moved to a standalone server or to a cluster that has only one active server which is enabled for high availability.

About this task

Event sequencing for a WebSphere MQ export is enabled from WebSphere Integration Developer.

Procedure

1. Set the eventSequencing property of your native MQ export binding to True.
2. Save the changes to your master configuration.

   Related concepts

   "Working with WebSphere MQ bindings" on page 187
   WebSphere MQ bindings can be administered from within the server.

Enabling event sequencing: generic JMS exports

Generic JMS export bindings support event sequencing. In order for events to be handled in the order in which they are received, you must configure properties of the binding.

Before you begin

If event sequencing for generic JMS export bindings is required in a network deployment environment, the module that has the export should be moved to a standalone server or to a cluster that has only one active server which is enabled for high availability.

About this task

Event sequencing for a generic JMS export is enabled from WebSphere Integration Developer.

Procedure

1. Set the eventSequencing property of your generic JMS export binding to True.
2. Save the changes to your master configuration.

   Related concepts

   "Generic JMS bindings" on page 166
   The generic JMS import and export bindings provide connectivity to third-party JMS 1.1 compliant providers. Its operation is similar to that of JMS bindings.

Troubleshooting bindings

Various error conditions can occur with bindings that are specific to the type of binding.

About this task

The manner in which error conditions are handled depends upon the type of binding concerned.

Troubleshooting JMS and WebSphere MQ JMS bindings

You can diagnose and fix problems with JMS and WebSphere MQ JMS bindings.

Implementation exceptions
About this task

In response to various error conditions, the JMS and MQ JMS import and export implementation can return one of two types of exceptions:

- ServiceBusinessException: this exception is returned if exception specified on the service business interface (WSDL port type or Java interface) occurred.
- ServiceRuntimeException: raised in all other cases. In most cases, the cause exception will contain the original exception. In the J2C case that would be ResourceException and in the JMS case, it would be JMSException.

When an XML schema has a type defined without a global element, the JMS bindings (JMSDataBindingImplXML and JMSDataBindingImplJava) cannot resolve the type to an element.

Schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.ibm.com" xmlns:tns="http://www.ibm.com">
 <!-- global element required but missing -->

<complexType name="Quote">
  <sequence>
   <element name="symbol" type="string"></element>
   <element name="price" type="float"></element>
   </sequence>
 </complexType>
</schema>
```

If you receive one of the following exceptions:

```
com.ibm.websphere.sca.ServiceRuntimeException:
caused by: java.lang.IllegalArgumentException:
{Quote}Quote is not corresponding to a global element.
```

Or

```
[8/25/06 10:20:40:938 PDT] 00000054 FFDC          Z
com.ibm.ws.sca.databinding.impl.DataBindingImplXML
com.ibm.ws.sca.databinding.impl.DataBindingImplXML#002 Exception:

org.eclipse.emf.ecore.xmi.FeatureNotFoundException:
Feature 'Quote' not found. (sca:/dataObject.xml, 2, 126)
```

This might mean you need to define a global element:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.ibm.com" xmlns:tns="http://www.ibm.com">

<element name="Quote" type="tns:Quote"></element> <!-- global element required -->

<complexType name="Quote">
  <sequence>
   <element name="symbol" type="string"></element>
   <element name="price" type="float"></element>
  </sequence>
 </complexType>

</schema>
```

JMS-based SCA messages not appearing in the failed event manager

About this task

If SCA messages originated through a JMS interaction fail, you would expect to find these messages in the failed event manager. If such messages are not appearing in the failed event manager, ensure that the underlying SIB destination of the JMS Destination has a maximum failed deliveries value greater than 1. Setting this value to 2 or more enables interaction with the failed event manager during SCA invocations for the JMS bindings.

WebSphere MQ JMS-based SCA messages not appearing in the failed event manager

About this task

If SCA messages originated through a WebSphere MQ JMS interaction fail, you would expect to find these messages in the failed event manager. If such messages are not appearing in the failed event manager, ensure that the value of the maximum retries property on the underlying listener port is equal to or greater than 1. Setting this value to 1 or more enables interaction with the failed event manager during SCA invocations for the MQ JMS bindings.

Faults

About this task

The argument that is expected to be passed to the JMSDataBinding and JMSObjectBinding depends on the interface operation and the input, output and fault types.

For faults, the outDataBindingType specified on the method binding is used. If none is specified, the binding level dataBindingType is used for all serialization and deserialization.

If the fault type is simple, a string is set on the JMS DataBinding representing the fault message. In addition IsBusinessException is set to true.

If the fault type is a data object, then a data object is set on the JMS DataBinding which represents the fault message. This scenario requires the use of JMSDataBinding.

Messages containing faults are handled by the JMS data bindings. A Boolean header property IsBusinessException is intercepted by the data binding. If the value is true, the data binding informs the runtime that the payload contains fault data.

If you are working with a custom data binding, you need to take the following steps to handle faults correctly. Default implementations handle faults without user intervention.

Procedure
1. For JMS exports, use the setBusinessException (boolean isBusinessException) method on the JMSDataBinding interface to indicate that data object or object specified on the data binding is a fault object and the message created by the binding should be constructed accordingly. The Data Binding is then responsible to specify the isBusinessException appropriately.
2. For JMS imports use the isBusinessException() method on the JMSDataBinding interface to indicate whether the message contains a fault.

    The data binding gets the value of the header property indicating a fault defined in the payload. After the runtime passes the JMS message to the data binding, it invokes isBusinessException() on the data binding. If returned value is false, the message is processed normally, otherwise, the ServiceBusinessException is returned to the caller. The data object or object produced by the binding is set on a ServiceBusinessException and it is returned to the caller.

Misusage scenarios: comparison with WebSphere MQ bindings
About this task

The WebSphere MQ JMS binding is designed to interoperate with JMS applications deployed against WebSphere MQ, which exposes messages according to the JMS message model. The WebSphere MQ import and export, however, are principally designed to interoperate with native WebSphere MQ applications, and expose the full content of the WebSphere MQ message body to mediations.

The following scenarios should be built using the WebSphere MQ JMS binding, not the WebSphere MQ binding:
- Invoking a JMS message-driven bean (MDB) from an SCA module, where the MDB is deployed against the WebSphere MQ JMS provider. This would be implemented using a WebSphere MQ JMS import.
- Allowing the SCA module to be called from a J2EE component servlet or EJB via JMS. This would be implemented using a WebSphere MQ JMS export.
- Mediating the contents of a JMS MapMessage, in transit across WebSphere MQ. This would use WebSphere MQ JMS export and import in conjunction with the appropriate data binding.

There are situations in which the WebSphere MQ binding and WebSphere MQ JMS binding might be expected to interoperate. In particular, when you are bridging between J2EE and non-J2EE WebSphere MQ applications, use a WebSphere MQ export and WebSphere MQ JMS import (or vice versa) in conjunction with appropriate data bindings, mediation module, or both.

Troubleshooting generic JMS bindings

Certain failure conditions can occur with a generic JMS binding.

About this task

Various error conditions can occur with generic JMS bindings.

Troubleshooting generic JMS exceptions

In response to various error conditions, the generic JMS import and export implementation can return an exception.

About this task

In response to various error conditions, the generic JMS import and export implementation can return one of two types of exceptions:

- ServiceBusinessException – this exception is returned if the exception specified on the service business interface (WSDL port type or Java interface) occurred.
- ServiceRuntimeException – raised in all other cases. In most cases, the *cause* exception will contain the original exception. In the case of JMS, it would be JMSException.

Troubleshooting generic JMS message expiry

A request message by the JMS provider is subject to expiration.

About this task

Request expiry refers to the expiration of a request message by the JMS provider when the JMSExpiration time on the request message is reached. As with other JMS bindings, the generic JMS binding handles the request expiry by setting expiration on the callback message placed by the import to be the same as for the outgoing request. Notification of the expiration of the callback message will indicate that the request message has expired and the client should be notified by means of a business exception.

If the callback destination is moved to the third-party provider, however, this type of request expiry is not supported.

Response expiry refers to the expiration of a response message by the JMS provider when the JMSExpiration time on the response message is reached.

Response expiry for the generic JMS binding is not supported, because the exact expiry behavior of a third-party JMS provider is not defined. You can, however, check that the response is not expired if and when it is received.

For outbound request messages, the JMSExpiration value will be calculated from the time waited and from the requestExpiration values carried in the asyncHeader, if set.

Troubleshooting generic JMS connection factory errors

When you define certain types of Connection Factory in your generic JMS provider, you may receive an error message when you try to start an application. You can modify the external Connection Factory to avoid this problem.

About this task

When launching an application you may receive the following error message: "MDB Listener Port JMSConnectionFactory type does not match JMSDestination type"

This problem can arise when you are defining external connection factories. Specifically the exception can be thrown when you create a JMS 1.0.2 Topic Connection Factory, instead of a JMS 1.1 (unified) Connection Factory (i.e., one that is able to support both point-to-point and publish/subscribe communication).

To resolve this issue take the following steps:

Procedure

1. Access the generic JMS provider that you are using.
2. Replace the JMS 1.0.2 Topic Connection Factory that you defined with a JMS 1.1 (unified) Connection Factory.

Results

When you launch the application with the newly defined JMS 1.1 Connection Factory you should no longer receive an error message.

Troubleshooting WebSphere MQ bindings

You can diagnose and fix faults and failure conditions that occur with WebSphere MQ bindings.

About this task

The primary failure conditions of WebSphere MQ bindings are determined by transactional semantics, by WebSphere MQ configuration, or by reference to existing behavior in other components. The primary failure conditions include:

- Failure to connect to WebSphere MQ queue manager or queue: a failure to connect to WebSphere MQ, to receive messages, will result in the MDB ListenerPort failing to start. This condition will be logged in the WebSphere Application Server log. Persistent messages will remain on the WebSphere MQ queue until they are successfully retrieved (or expired by WebSphere MQ). A failure to connect to WebSphere MQ, to send outbound messages, will cause rollback of the transaction controlling the send.
- Failure to parse inbound message, or to construct outbound message: a failure in the data binding will cause rollback of the transaction controlling the work.
- Failure to send outbound message: a failure to send a message will cause rollback of the relevant transaction
- Multiple or unexpected response messages: the import expects only one response message for each request message; when a response has arrived, the record is deleted. If response messages arrive unexpectedly, they will be discarded as with the JMS import.

Misusage scenarios: comparison with WebSphere MQ JMS bindings

About this task

The WebSphere MQ import and export are principally designed to interoperate with native WebSphere MQ applications, and expose the full content of the WebSphere MQ message body to mediations. The WebSphere MQ JMS binding, however, is designed to interoperate with JMS applications deployed against WebSphere MQ, which exposes messages according to the JMS message model.

The following scenarios should be built using the WebSphere MQ JMS binding, not the WebSphere MQ binding:

- Invoking a JMS message-driven bean (MDB) from an SCA module, where the MDB is deployed against the WebSphere MQ JMS provider. This would be implemented using a WebSphere MQ JMS import.
- Allowing the SCA module to be called from a J2EE component servlet or EJB via JMS. This would be implemented using a WebSphere MQ JMS export.
- Mediating the contents of a JMS MapMessage, in transit across WebSphere MQ. This would use WebSphere MQ JMS export and import in conjunction with the appropriate data binding.

There are situations in which the WebSphere MQ binding and WebSphere MQ JMS binding might be expected to interoperate. In particular, when you are bridging between J2EE and non-J2EE WebSphere MQ applications, use a WebSphere MQ export and WebSphere MQ JMS import (or vice versa) in conjunction with appropriate data bindings and/or mediation module.

If WebSphere MQ cannot deliver a message to its intended destination, usually due to configuration errors, it will send messages instead to a nominated dead-letter queue. In doing so it adds a dead-letter header to the start of the message body; this header contains failure reasons, the original destination, and other information.

Key features of WebSphere MQ JMS bindings

Key features of WebSphere MQ JMS bindings include headers, J2EE artifacts and created J2EE resources.

Headers

A JMS message header contains a number of predefined fields containing values used by both clients and providers to identify and to route messages.

JMSCorrelationID
Links to a related message. Typically, this field is set to the message identifier string of the message that is being replied to.

TargetFunctionName
This header is used on the export binding to map from native method to operation method. Setting the string JMS header property TargetFunctionName will indicate to the JMSFunctionSelectorImpl of the export the native method that should be used to map to the operation method. To use this, specify the native method as in the method binding as the value. You must set this as a JMS header property in the import binding definition if you expect the export to use the JMSFunctionSelectorImpl.

Correlation schemes

The WebSphere MQ JMS bindings provide various correlation schemes that are used to determine how to correlate request messages with response messages.

RequestMsgIDToCorrelID
> The JMSMessageID is copied to the JMSCorrelationID field. This is the default setting.

RequestCorrelIDToCorrelID
> The JMSCorrelationID is copied to the JMSCorrelationID field.

J2EE resources

A number of J2EE resources are created when a MQ JMS import is deployed to a J2EE environment.

Parameters

MQ Connection Factory
> Used by clients to create a connection to the MQ JMS provider.

Response Connection Factory
> Used by the SCA MQ JMS runtime when the send destination is on a different Queue Manager than the receive destination.

Listener Port
> Specifies an association between a connection factory, a destination, and a message-driven bean. This enables deployed message-driven beans associated with the port to retrieve messages from the destination.

Destinations
- Send destination:
    - Imports: Where the request or outgoing message is sent.
    - Exports: Where the response message will be sent if it is not superseded by the JMSReplyTo header field of the incoming message.
- Receive destination:
    - Imports: Where the response or incoming message should be placed.
    - Exports: Where the incoming or request message should be placed.

Key features of a WebSphere MQ binding

Key features of a WebSphere MQ binding include headers, J2EE artifacts and created J2EE resources.

Correlation schemes

A WebSphere MQ request/reply application can use one of a number of techniques to correlate response messages with requests, built around the MQMD's MessageID and CorrelID fields. In the vast majority of cases, the requestor lets the queue manager select a MessageID and expects the responding application to copy this into the CorrelID of the response. In most cases, the requestor and responding application implicitly know which correlation technique is in use. Occasionally the responding application will honor various flags in the Report field of the request that describe how to handle these fields.

Export bindings for WebSphere MQ messages can be configured with the following options:

Response MsgId options:

New MsgID
　　Allows the queue manager to select a unique MsgId for the response (default).

Copy from Request MsgID
　　Copies the MsgId field from the MsgId field in the request.

Copy from SCA message
　　Sets the MsgId to be that carried in WebSphere MQ headers in the SCA response message; or let the queue manager define a new Id if the value does not exist.

As Report Options
　　Inspects the Report field of the MQMD in the request for a hint as to how to handle the MsgId. The MQRO_NEW_MSG_ID and MQRO_PASS_MSG_ID options are supported, and behave like New MsgId and Copy from Request MsgID, respectively

Response CorrelId options:

Copy from Request MsgID
　　Copies the CorrelId field from the MsgId field in the request (default).

Copy from Request CorrelID
　　Copies the CorrelId field from the CorrelId field in the request.

Copy from SCA message
　　Sets the CorrelId to be that carried in WebSphere MQ headers in the SCA response message; left blank if the value doesn't exist.

As Report Options
　　Inspects the Report field of the MQMD in the request for a hint as to how to handle the CorrelId. The MQRO_COPY_MSG_ID_TO_CORREL_ID and MQRO_PASS_CORREL_ID options are supported, and behave like Copy from Request MsgID and Copy from Request CorrelID, respectively

Import bindings for WebSphere MQ messages can be configured with the following options:

Request MsgId options:

New MsgID
　　Allows the queue manager to select a unique MsgId for the request (default)

Copy from SCA message
　　Sets the MsgId to be that carried in WebSphere MQ headers in the SCA request message; or let the queue manager define a new Id if the value doesn't exist.

Response correlation options:

Response has CorrelID copied from MsgId
　　Expects the response message to have a CorrelId field set as per the request's MsgId (default).

Response has MsgID copied from MsgId
　　Expects the response message to have a MsgId field set as per the request's MsgId.

Response has CorrelID copied from CorrelId
  Expects the response message to have a CorrelId field set as per the request's CorrelId.

J2EE resources

A number of J2EE resources are created when a WebSphere MQ binding is deployed to a J2EE environment.

Parameters

MQ Connection Factory
  Used by clients to create a connection to the MQ JMS provider.

Response Connection Factory
  Used by the SCA MQ runtime when the send destination is on a different Queue Manager than the receive destination.

Listener Port
  Specifies an association between a connection factory, a destination, and a message-driven bean. This enables deployed message-driven beans associated with the port to retrieve messages from the destination.

Destinations
  - Send destination: where the request or outgoing message is sent (import); where the response message will be sent (export), if not superseded by the MQMD ReplyTo header field in the incoming message.
  - Receive destination: where the response/request or incoming message should be placed.

Key features of generic JMS bindings

The features of the generic JMS import and export binding are consistent with those of the WebSphere embedded JMS and MQ JMS import bindings. Key features include header defininitions and access to existing J2EE resources. However, due to its generic nature, there are no JMS provider-specific connectivity options, and this binding has limited capability to generate resources at deployment and installation.

Generic imports

Like the MQ JMS import application, the generic JMS implementation is asynchronous and supports three invocations: one-way, two-way (also known as request-response), and callback.

When the JMS import is deployed, a message driven bean (MDB) provided by the runtime environment is deployed. The MDB listens for replies to the request message. The MDB is associated with (listens on) the destination sent with the request in the replyTo header field of the JMS message.

Generic exports

Generic JMS export bindings differ from EIS export bindings in their handling of the return of the result. A generic JMS export explicitly sends the response to the replyTo destination specified on the incoming message. If none is specified, the send destination is used.

When the generic JMS export is deployed, a message driven bean (a different MDB than the one used for generic JMS imports) is deployed. It listens for the incoming requests on the receive destination and then dispatches the requests to be processed by the SCA runtime.

Special headers

Special header properties are used in generic imports and exports to tell the target binding how to handle the message.

For example, the TargetFunctionName property is used by the default function selector to identify the name of the operation in the import or export's interface that is being invoked.

J2EE resources

A number of J2EE resources are created when a JMS binding is deployed to a J2EE environment.

- Listener port for listening on the receive (response) destination (two-way only) for imports, and on the receive (request) destination for exports
- Generic JMS connection factory for the outboundConnection (import) and inboundConnection (export)
- Generic JMS destination for the send (import) and receive (export; two-way only) destinations
- Generic JMS connection factory for the responseConnection (two-way only and optional; otherwise, outboundConnection is used for imports, and inboundConnection is used for exports)
- Generic JMS destination for the receive (import) and send (export) destination (two-way only)
- Default messaging provider callback JMS destination used to access the SIB callback queue destination (two-way only)
- Default messaging provider callback JMS connection factory used to access the callback JMS destination (two-way only)
- SIB callback queue destination used to store information about the request message for use during response processing (two-way only)

The installation task creates the ConnectionFactory, the three destinations, and theActivationSpec from the information in the import and export files.

Key features of JMS bindings

Key features of JMS import and export bindings include headers and created J2EE resources.

Differences from EIS

Imports

The JMS import application differs from an EIS import in its runtime invocation with respect to handling of responses. The JMS implementation is asynchronous, and the SCA programming model requires it to support three invocations: one-way, two-way (request-response), and callback.

When the JMS import is deployed, a message driven bean (MDB), provided by the runtime environment, is deployed to listen for replies to the request message. The MDB is associated with (listens on) the destination sent with the request in the replyTo header field of the JMS message.

Exports

JMS export bindings differ from EIS export bindings in their handling of the return of the result. A JMS export explicitly sends the response to the JMSReplyTo destination specified on the incoming message. If none is specified, the send destination is used.

When the JMS export is deployed, a message driven bean (a different MDB than the one used for JMS imports) is deployed. It listens for the incoming requests on the receive destination and then dispatches the requests to be processed by the SCA runtime.

Special headers

Special header properties are used in JMS imports and exports to tell the target how to handle the message.

For example, TargetFunctionName maps from the native method to the operation method. It is used by the default function selector (JMSFunctionSelectorImpl) to extract the native method name from the JMS message.

J2EE resources

A number of J2EE resources are created when JMS imports and exports are deployed to a J2EE environment.

ConnectionFactory
Used by clients to create a connection to the JMS provider.

ActivationSpec
Imports use this for receiving the response to a request; exports use it when configuring the message endpoints that represent MDBs in their interactions with the messaging system.

Destinations
- Send destination: on an import, this is where the request or outgoing message is sent; on an export, this is the destination where the response message will be sent, if not superseded by the JMSReplyTo header field in the incoming message.
- Receive destination: where the incoming message should be placed; with imports, this is a response; with exports, this is a request.
- Callback destination: SCA JMS system destination used to store correlation information. Do not read or write to this destination.

The installation task creates the ConnectionFactory and three destinations. It also creates the ActivationSpec to enable the runtime MDB to listen for replies on the receive destination. The properties of these resources are specified in the import or export file.

JMS bindings: a general perspective

JMS bindings provide connectivity between the Service Component Architecture (SCA) environment and external JMS systems.

JMS bindings

The major components of both JMS import and JMS export bindings are:

- Resource adapter: enables managed, bidirectional connectivity between an SCA module and external JMS systems.
- Connections: encapsulate a virtual connection between a client and a provider application.
- Destinations: used by clients to specify the target of messages it produces or the source of messages it consumes.
- Authentication data: used to secure access to the binding.

JMS import bindings

JMS import bindings allow you to import an external JMS application to be used inside your SCA module.

Connections to the associated JMS provider of JMS destinations are created by using a JMS connection factory. Use connection factory administrative objects to manage JMS connection factories for the default messaging provider.

Importing services from external JMS systems utilizes a destination that the data will be sent to and a destination where the reply can be received.

Two types of usage scenario for the JMS import binding are supported:

- One-way: a message is sent to the send destination in the import file. Nothing is set in the replyTo field of the JMS header.
- Two-way (request-response): a message is sent to the send destination specified in the import file. The receive destination is set in the replyTo header property of the outbound message. A message driven bean (MDB) is deployed to listen on the receive destination, and when a reply is received the MDB invokes the callback object. For request-response scenarios, the response can copy the request messageId to the correlationId field of the response message (default), or the response can copy the request correlationId to the correlationId field of the response message.

For both one-way and two-way usage scenarios, dynamic and static header properties can be specified. Some of these properties have special meanings to the SCA JMS runtime.

Figure 7 on page 159 illustrates how the import is linked to the external service.
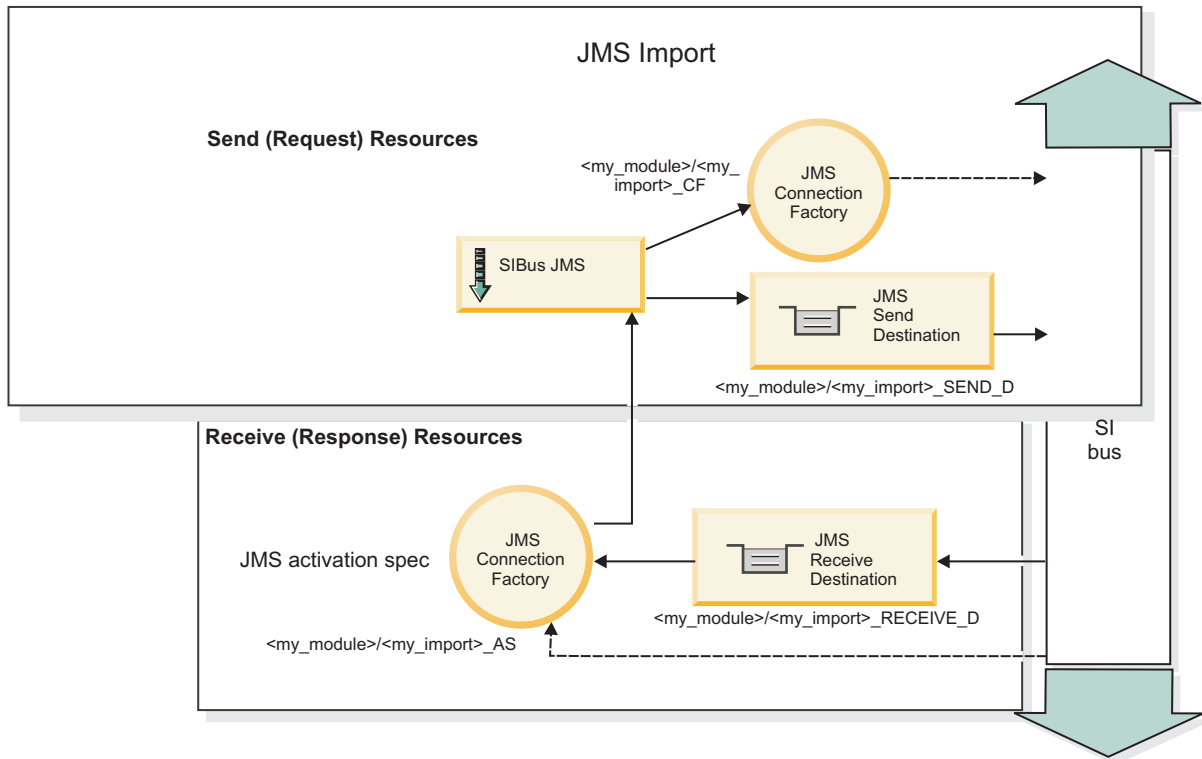
Figure 16. JMS import binding resources

JMS export bindings

JMS export bindings provide inbound connectivity from JMS to the SCA system.

The connection that is part of a JMS export is a configurable activation specification.

A JMS export has send and receive destinations. The receive destination is where the incoming message for the target component should be placed. The send destination is where the reply will be sent, unless the incoming message has superseded this using the replyTo header property.

An MDB is deployed to listen to requests incoming to the receive destination specified in the export binding. The destination specified in the send field is used to send the reply to the inbound request if the invoked application provides a reply. The destination specified in the replyTo field of the incoming message overwrites the destination specified in the send.

Figure 8 on page 160 illustrates how the external requestor is linked to the export.

Figure 17. JMS export binding resources

Generic JMS bindings: a general perspective

Generic JMS bindings are non-JCA JMS bindings that provide connectivity between the Service Component Architecture (SCA) environment and external JMS systems that are compliant with JMS 1.1 and that implement the optional JMS Application Server Facility.

Generic JMS bindings

The major components of the bindings for both generic JMS imports and generic JMS exports are:

- Resource adapter: enables managed, bidirectional connectivity between enterprise information systems (EISs) and J2EE components.
- Connections: encapsulate a virtual connection between a client and a provider application.
- Destinations: used by clients to specify the target of messages it produces or the source of messages it consumes.
- Authentication data: used to secure access to the binding.

Generic JMS import bindings

Generic JMS import bindings provide outbound connectivity from SCA applications to non-JCA 1.5-compliant JMS providers.

The connection part of a JMS import is a connection factory. A connection factory, the object a client uses to create a connection to a provider, encapsulates a set of connection configuration parameters defined by an administrator. Each connection factory is an instance of the ConnectionFactory, QueueConnectionFactory, or TopicConnectionFactory interface.

Importing services from JMS utilizes a destination that the data will be sent to and a destination where the reply can be received.

Two types of usage scenario for JMS import binding are supported:

- One-way: a message is sent to the send destination specified in the import file. Nothing is sent to the ReplyTo field of the JMS header.
- Two-way (request-response): a message is sent to the send destination in the import file. The receive destination is set in the ReplyTo header property. A message driven bean (MDB) is deployed to listen on the receive destination; when a reply is received, the MDB invokes the callback object. For request-response scenarios, the response can copy the request messageId to the correlationId field of the response message (default), or the response can copy the request correlationId to the correlationId field of the response message.

For both one-way and two-way usage scenarios, dynamic and static header properties can be specified. Some of these properties have special meanings to the SCA JMS runtime.

Figure 10 on page 168 illustrates how the import is linked to the external service.
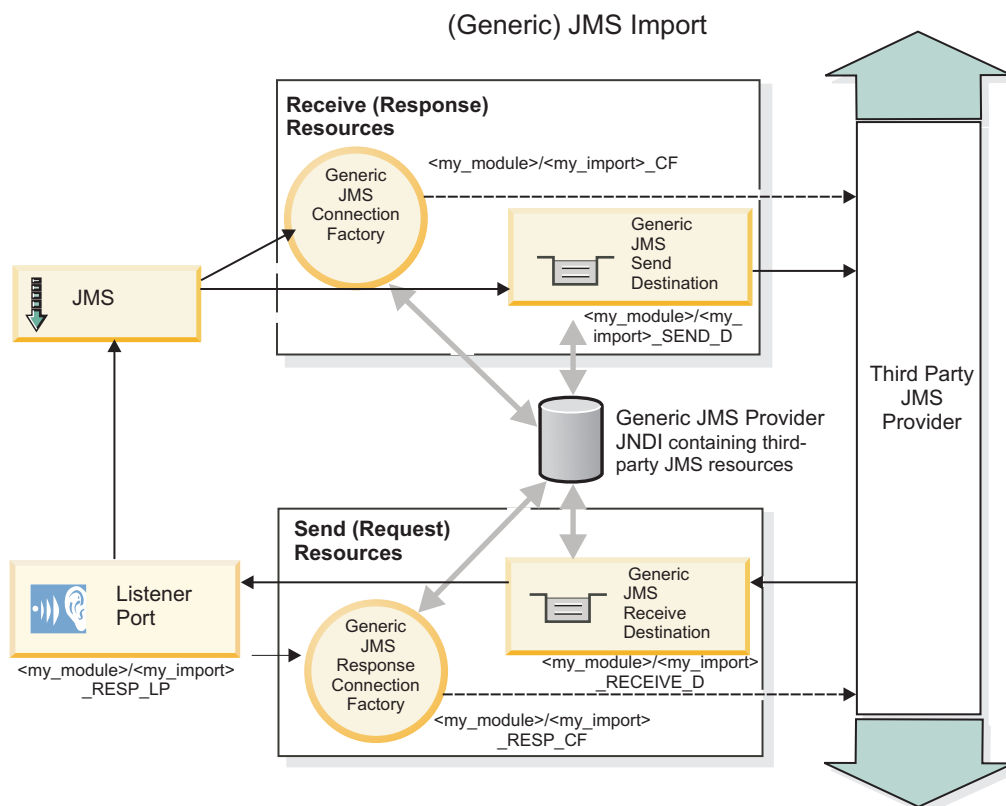
Figure 18. Generic JMS import binding resources

Generic JMS export bindings

Generic JMS export bindings provide inbound connectivity from JMS to the SCA system.

The connection part of a JMS export is composed of a ConnectionFactory and a ListenerPort.

A generic JMS export has send and receive destinations. The receive destination is where the incoming message for the target component should be placed. The send destination is where the reply will be sent, unless the incoming message has superseded this using the replyTo header property. An MDB is deployed to listen to requests incoming to the receive destination specified in the export binding.

- The destination specified in the send field is used to send the reply to the inbound request if the invoked application provides a reply.
- The destination specified in the replyTo field of the incoming message overwrites the destination specified in the *send* field.
- For request/response scenarios, the response can copy the request messageId to the correlationId field of the response message (default), or the response can copy the request correlationId to the to the correlationId field of the response message.

Figure 11 on page 169 illustrates how the external requestor is linked to the export.
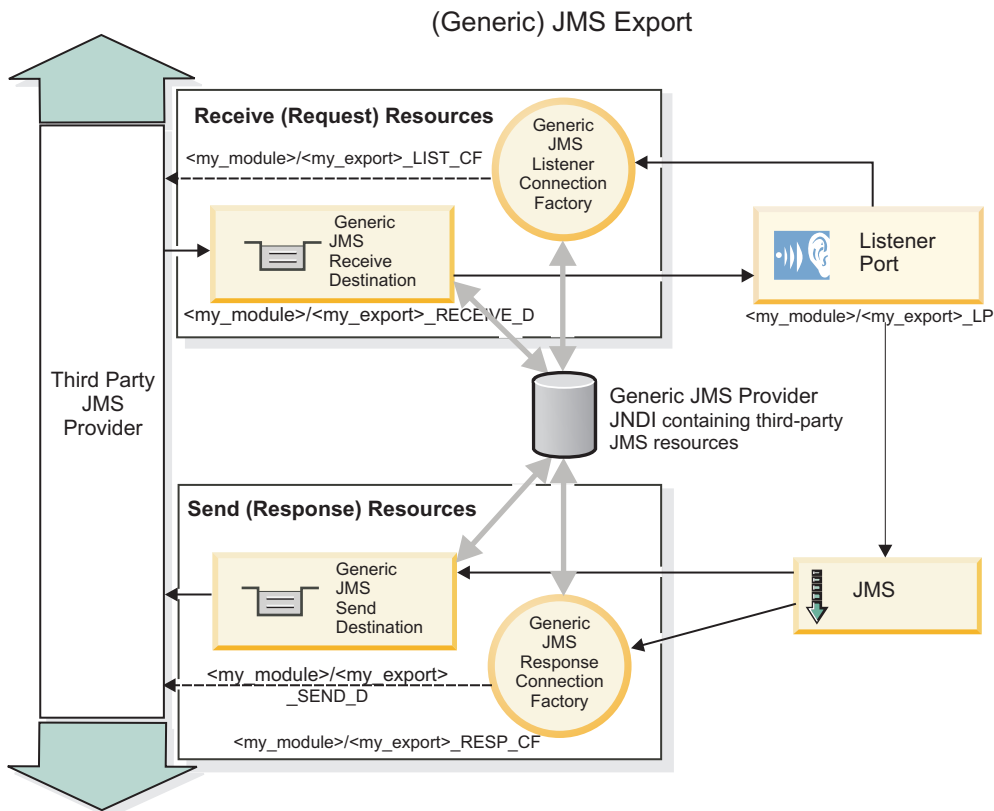
Figure 19. Generic JMS export binding resources

WebSphere MQ JMS bindings: a general perspective

The WebSphere MQ JMS application provides integration with WebSphere MQ using the JMS API.

WebSphere MQ administrative tasks

The WebSphere MQ system administrator is expected to create the underlying WebSphere MQ Queue Manager, which the WebSphere MQ JMS bindings will utilize before running an application containing these bindings.

WebSphere MQ JMS import bindings

The WebSphere MQ JMS import binding provides outbound connectivity from Service Component Architecture (SCA) applications to WebSphere MQ JMS-based providers. You must be using a supported version of WebSphere MQ. Detailed hardware and software requirements can be found on the IBM support pages.

Two types of usage scenario for WebSphere MQ JMS import bindings are supported:
- One-way: a message is sent to the send destination in the import file. Nothing is sent to the replyTo field of the JMS header.
- Two-way (request-response): a message is sent to the send destination in the import file. The receive destination is is set in the replyTo header field. A message-driven bean (MDB) is deployed to listen on the receive destination, and when a reply is received the MDB invokes the callback object. For request-response scenarios, the response is expected to have the correlationId set to the sent message messageID for the default correlation scheme.

For both one-way and two-way usage scenarios, dynamic and static header properties can be specified. Some of these properties have special meanings to the SCA JMS runtime.

Figure 12 on page 176 illustrates how the import is linked to the external service.
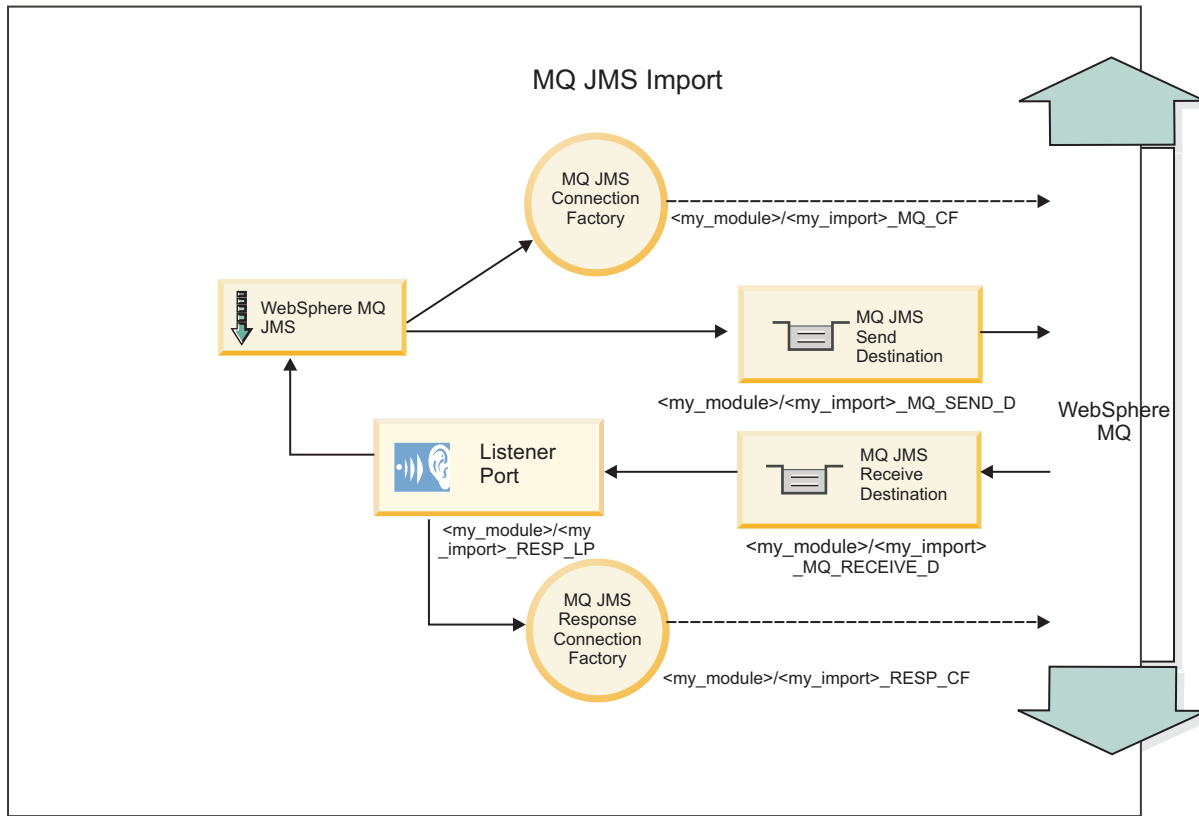
Figure 20. WebSphere MQ JMS import binding resources

WebSphere MQ JMS export bindings

The WebSphere MQ JMS export binding provides inbound connectivity from WebSphere MQ-based JMS provider to the SCA system.

An MDB is deployed to listen to requests incoming to the receive destination specified in the export binding. The destination specified in the send field is used to send the reply to the inbound request if the invoked application provides a reply. The destination specified in the replyTo field of the response message overwrites the destination specified in the send field.

Figure 13 on page 177 illustrates how the external requestor is linked to the export.

MQ JMS Export
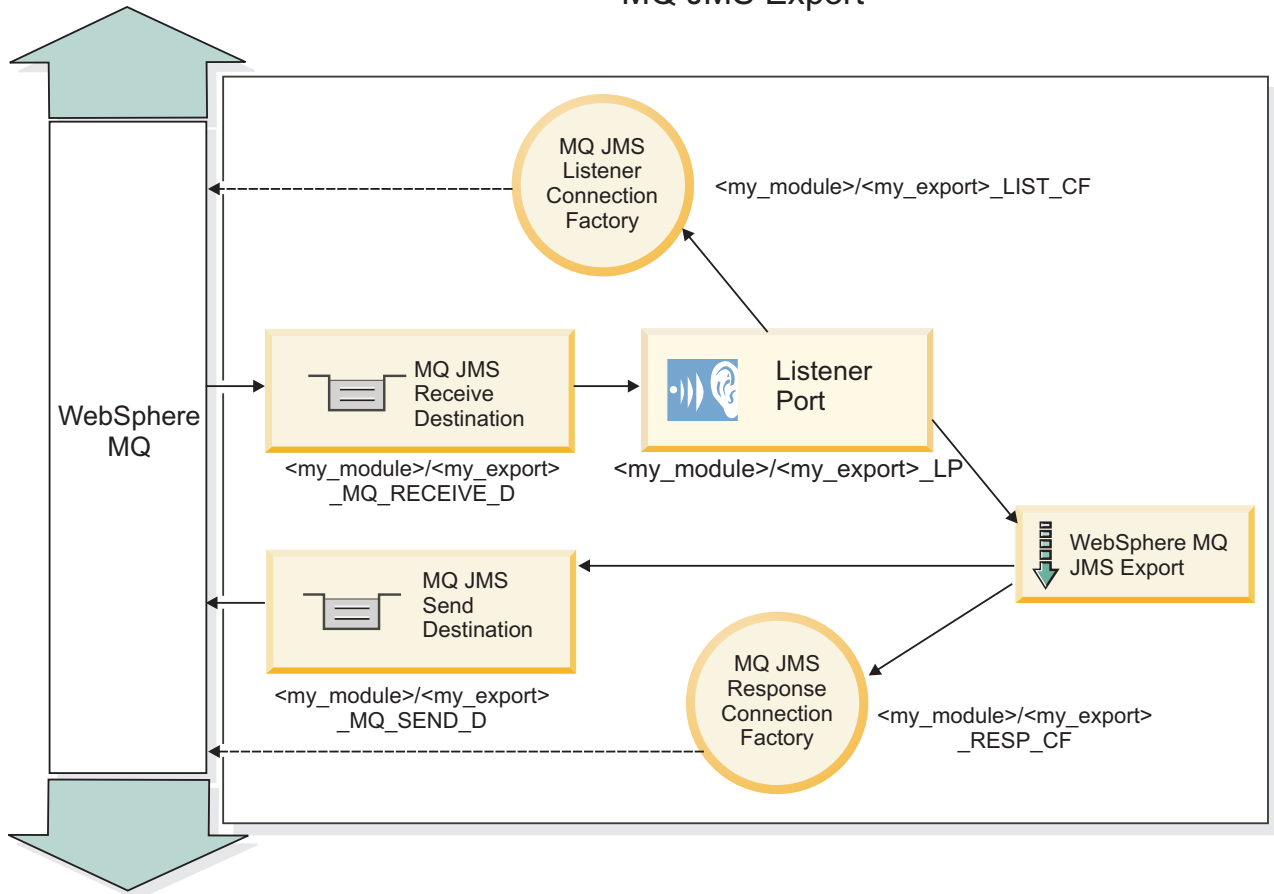
Figure 21. WebSphere MQ JMS export binding resources

WebSphere MQ bindings: a general perspective

The WebSphere MQ application provides integration with native MQ-based applications.

WebSphere MQ administrative tasks

The WebSphere MQ system administrator is expected to create the underlying WebSphere MQ Queue Manager, which the WebSphere MQ bindings will use before running an application containing these bindings.

WebSphere MQ import bindings

The WebSphere MQ import binding provides outbound connectivity from Service Component Architecture (SCA) applications to WebSphere MQ-based applications. You must be using a supported version of WebSphere MQ. Detailed hardware and software requirements can be found on the IBM support pages.

Two types of WebSphere MQ import bindings are supported:
- One-way: a message is sent to the destination specified as send in the import file. Nothing is sent to the replyTo field of the MQMD header.

- Two-way (request-response): a message is sent to the destination specified as send in the import file. The destination specified as receive is set in the replyTo MQMD header field. A message-driven bean (MDB) is deployed to listen on the receive destination, and when a reply is received the MDB invokes the callback object. For request-response scenarios, the response is expected to have the correlationId set to the sent message messageID for the default correlation scheme.

Figure 14 on page 184 illustrates how the import is linked to the external service.
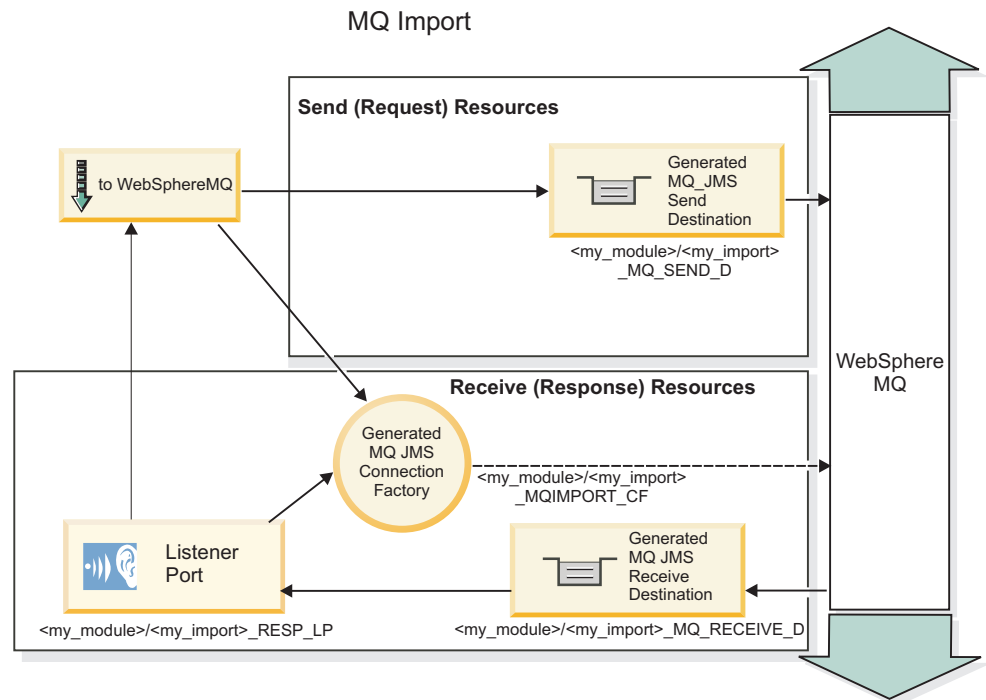
Figure 22. WebSphere MQ import binding resources

WebSphere MQ export bindings

The WebSphere MQ export binding provides inbound connectivity from WebSphere MQ-based applications to the SCA system.

An MDB is deployed to listen to requests incoming to the receive destination specified in the export binding. The destination specified in the send field is used to send the reply to the inbound request if the invoked application provides a reply. The destination specified in the replyTo field of the response message overwrites the destination specified in the send field.

Figure 15 on page 185 illustrates how the external requestor is linked to the export.
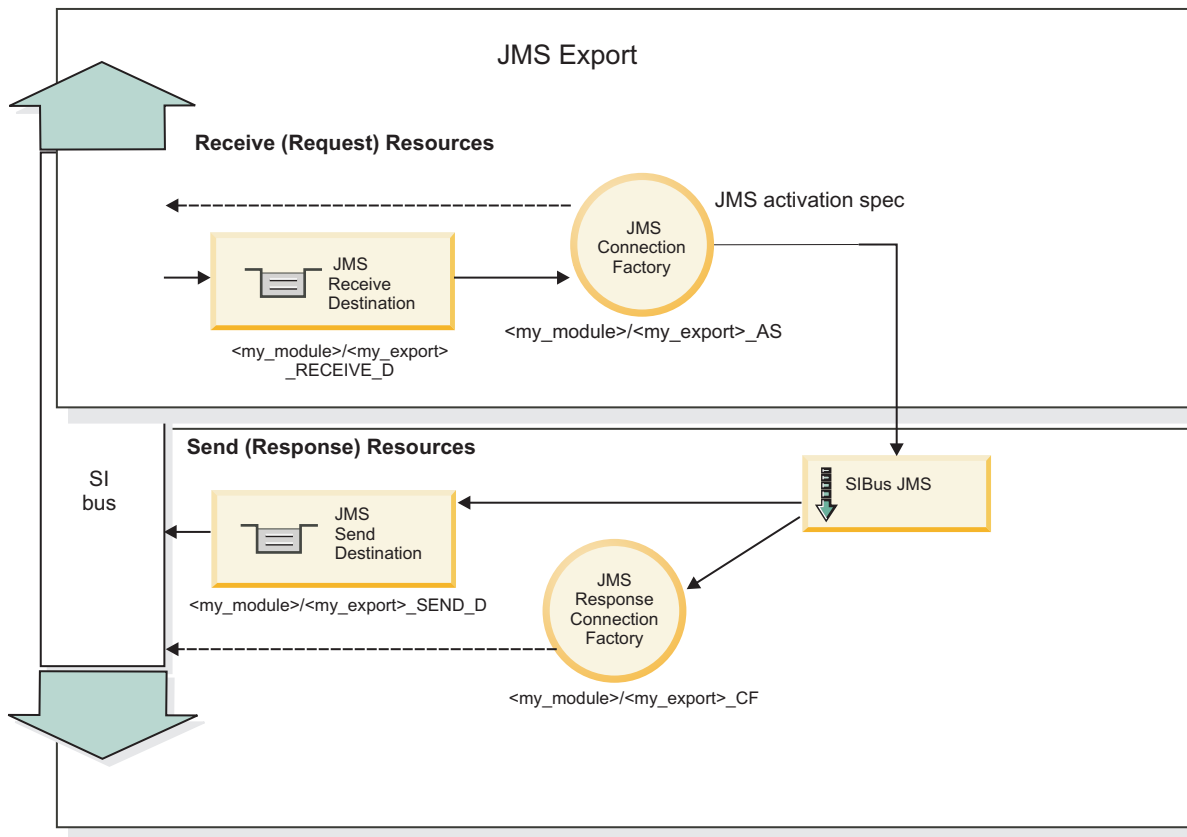
MQ Export

Receive (Request) Resources

Generated MQ JMS Receive Destination

<my_module>/<my_export> _MQ_RECEIVE_D

Generated MQ JMS Connection Factory

<my_module>/<my_export> _MQEXPORT_CF

<my_module>/<my_export>_LP

Listener Port

WebSphere MQ

Send (Response) Resources

Generated MQ_JMS Send Destination

<my_module>/<my_export>_MQ_SEND_D

WMQ
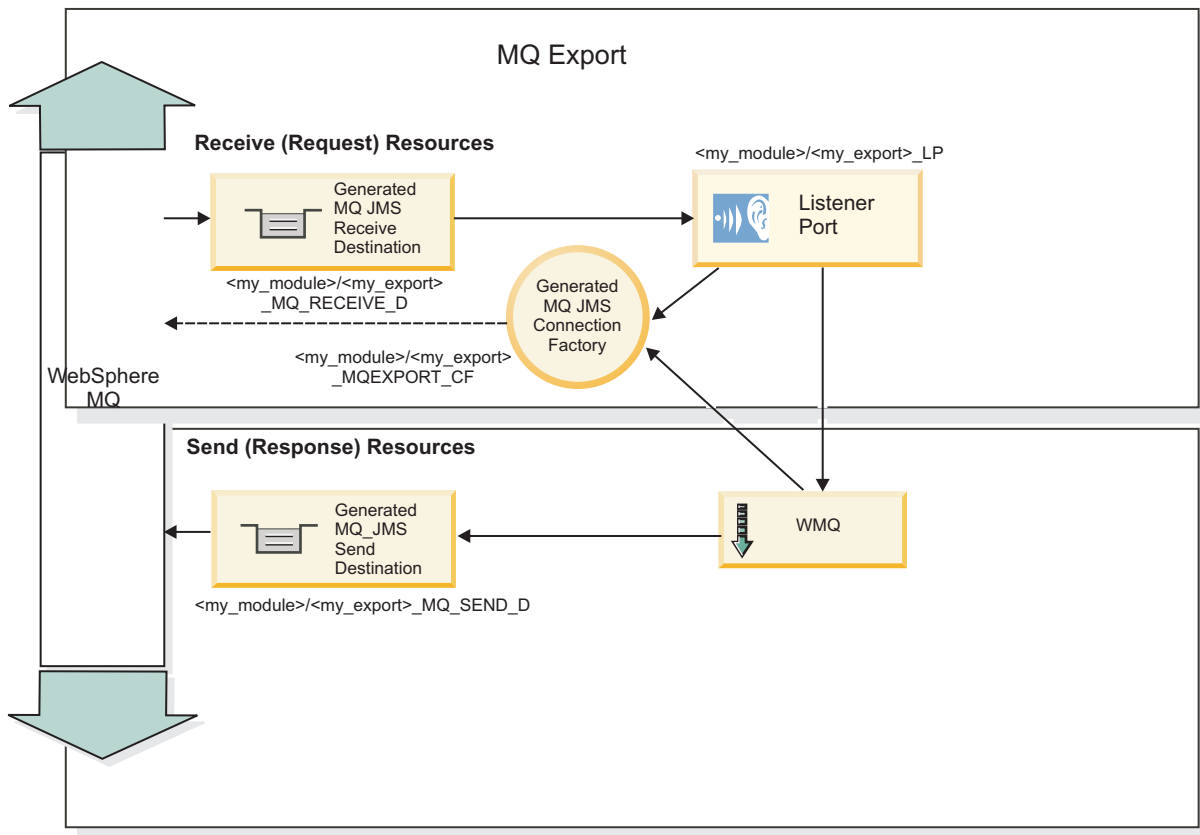
Figure 23. WebSphere MQ export binding resources

Parameters of JMS bindings

The JMS import and export bindings can be installed with all the necessary connection factories created, or they can be designed to point to a pre-configured set of artifacts on the server.

Typically, JMS import and export bindings are created in WebSphere Integration Developer. At development time you have an option to create the connections and destinations required for the JMS binding at install time, i.e., when the component is installed on your server. Your other option is to specify the JNDI name of the resources on the server that you intend your JMS imports and exports to utilize.

Configuring the JMS binding depends upon which option was selected.

In the case where new message provider resources are created (that is, the resources are created on the server at install time), the resources will exist and can be located and administered using the administrative console. The JNDI names of the generated artifacts for imports and exports are described in the following tables.

Table 24. JMS import bindings: Names and JNDI names of resources created at installation on the server

Resource	Module name	Import name	Resource global JNDI name
Connection Factory	jms.module	my/import	jms.module/my/import_CF
Activation Spec	jms.module	my/import	jms.module/my/import_AS
Destination	jms.module	my/import	jms.module/my/import_SEND_D, jms.module/my/import_RECEIVE_D, jms.module/my/import_CALLBACK_D
SIB Destinations	jms.module	my/import	jms.module.my.import_SEND_D_SIB, jms.module.my.import_RECEIVE_D_SIB, jms.module.my.import_CALLBACK_D_SIB

Table 25. JMS export bindings: Names and JNDI names of resources created at installation on the server

Resource	Module name	Export name	Resource global JNDI name
Connection Factory	jms.module	my/export	jms.module/my/export_CF
Activation Spec	jms.module	my/export	jms.module/my/export_AS
Destination	jms.module	my/export	jms.module/my/export_SEND_D, jms.module/my/export_RECEIVE_D, jms.module/my/export_CALLBACK_D
SIB Destinations	jms.module	my/export	jms.module.my.export_SEND_D_SIB, jms.module.my.export_RECEIVE_D_SIB, jms.module.my.export_CALLBACK_D_SIB

Note: The resources are created at the server scope. The scope in the administrative console is initially set at "All scopes". You must set the scope to "cell" or "node" to create a new resource. You can select an existing resource from the default list.

If you select the other option and the binding is expecting to find on the server resources for its use, you must have these resources installed and the import and binding files must contain the appropriate JNDI names. The association between the JMS binding and the resources will then be made.

Parameters of generic JMS bindings

The generic JMS import and export bindings can be installed with all the necessary connection factories created, or they can be designed to point to a pre-configured set of artifacts on the server.

Typically, the generic JMS bindings are created in WebSphere Integration Developer. At development time you can either create the connections and destinations required for the JMS binding at the time the component is installed on your server, or you can specify the JNDI name of the resources on the server that you intend your JMS import or export to utilize.

Configuring the generic JMS binding depends upon which option was selected.

In the case where new message provider resources are created (i.e., the resources are created on the server at install time), the resources will exist and can be located and administered using the administrative console. The JNDI names of the generated artifacts are described in the following tables.

Table 26. Generic JMS imports: Names and JNDI names of resources created at installation on the server

Resource	Generated resource JNDI name
outboundConnection	[moduleName]/[importName]_CF
responseConnection	[moduleName]/[importName]_RESP_CF
send destination	[moduleName]/[importName]_SEND_D
receive destination	[moduleName]/[importName]_RECEIVE_D
callback destination	[moduleName]/[importName]_CALLBACK_D

Table 27. Generic JMS exports: Names and JNDI names of resources created at installation on the server

Resource	Generated resource JNDI name
inboundConnection	[moduleName]/[exportName]_LIS_CF
responseConnection	[moduleName]/[exportName]_RESP_CF
receive destination	[moduleName]/[exportName]_RECEIVE_D
send destination	[moduleName]/[exportName]_SEND_D
callback destination	[moduleName]/[exportName]_CALLBACK_D

Note: The resources are created at the server scope. The scope in the administrative console is initially set at "All scopes". You must set the scope to "cell" or "node" to create a new resource. You can select an existing resource from the default list.

In the alternate case, where the JMS import is expecting to find required resources on the server, you must have these resources installed and the import and export files must contain the JNDI names. The association between the JMS binding and the resources will then be made.

Parameters of WebSphere MQ bindings

The WebSphere MQ binding can be installed with all the necessary connection factories created, or it can be designed to point to a pre-configured set of artifacts on the server.

Typically, WebSphere MQ import and export bindings are created in WebSphere Integration Developer. At development time you have an option to create the connections and destinations required for the WebSphere MQ import at install time, i.e., when the component is installed on your server. Alternatively, you can specify the JNDI name of the resources on the server that you intend your WebSphere MQ binding to utilize.

Configuring the WebSphere MQ binding depends upon which option was selected.

In the case where new message provider resources are created (i.e., the resources are created on the server at install time), the resources will exist and can be located and administered using the administrative console. The JNDI names of the

generated artifacts are described in the following tables.

Table 28. WebSphere MQ import: Names and JNDI names of resources created at installation on the server

Resource	Module name	Import name	Resource global JNDI name
Outbound Connection Factory	mq.module	my/import	mq.module/my/import_MQ_CF
Response Listener Port	mq.module	my/import	mq.module.my.import_RESP_LP (Note: This is only a name, not JNDI)
Response Connection Factory	mq.module	my/import	mq.module/my/import_RESP_CF
Send	mq.module	my/import	mq.module/my/import_MQ_SEND_D
Receive	mq.module	my/import	mq.module/my/export_MQ_RECEIVE_D
SIB Callback Destination	mq.module	my/import	mq.module/my/import_MQ_CALLBACK_D
SIB Callback Connection Factory	All modules	my/import	SCA.MQ/Callback_CF

Table 29. WebSphere MQ export: Names and JNDI names of resources created at installation on the server

Resource	Module name	Export name	Resource global JNDI name
Listener Port	mq.module	my/export	mq.module.my.export_LP (Note: This is just a name, not JNDI)
Inbound Connection Factory	mq.module	my/export	mq.module/my/export_LIS_CF
Response Connection Factory	mq.module	my/export	mq.module/my/export_RESP_CF
Receive	mq.module	my/export	mq.module/my/export_MQ_RECEIVE_D
Send	mq.module	my/export	mq.module/my/export_MQ_SEND_D
SIB Callback Destination	mq.module	my/export	mq.module/my/export_MQ_CALLBACK_D
SIB Callback Connection Factory	All modules	my/export	SCA.MQ/Callback_CF

Note:

- The resources are created at the server scope. The default scope in the administrative console is cell. You must change the scope in order to locate and administer the resources.
- The SIB Callback Destination and SIB Callback Connection Factory are SIB JMS resources. The other entries in the table are WebSphere MQ JMS resources. The two types of resources are administered.

In the alternate case, where the WebSphere MQ binding is expecting to find resources on the server that it will use, you must have these resources installed

and the import or export file must contain their JNDI names. The association between the WebSphere MQ binding and the resources will then be made.

Parameters of MQ JMS bindings

MQ JMS bindings can be installed with all the necessary connection factories created, or they can be designed to point to a pre-configured set of artifacts on the server.

Typically, MQ JMS bindings are created in WebSphere Integration Developer. At development time you have an option to create the connections and destinations required for the MQ JMS binding at install time, i.e., when the component is installed on your server. Alternatively, you can specify the JNDI name of the resources on the server that you intend your MQ JMS binding to utilize.

Configuring the MQ JMS binding depends upon which option was selected.

In the case where new message provider resources are created (i.e., the resources are created on the server at install time), the resources will exist and can be located and administered using the administrative console. The JNDI names of the generated artifacts are described in the following tables.

Table 30. MQ JMS imports: Names and JNDI names of resources created at installation on the server

Resource	Module name	Import name	Resource global JNDI name
Outbound Connection Factory	mqjms.module	my/import	mqjms.module/my/import_MQ_CF
Response Listener Port	mqjms.module	my/import	mqjms.module.my.import_RESP_LP (Note: This is only a name, not JNDI)
Response Connection Factory	mqjms.module	my/import	mqjms.module/my/import_RESP_CF
Send	mqjms.module	my/import	mqjms.module/my/import_MQ_SEND_D
Receive	mqjms.module	my/import	mqjms.module/my/export_MQ_RECEIVE_D
SIB Callback Destination	mqjms.module	my/import	mqjms.module/my/import_MQ_CALLBACK_D
SIB Callback Connection Factory	All modules	my/import	SCA.MQJMS/Callback_CF

Table 31. MQ JMS exports: Names and JNDI names of resources created at installation on the server

Resource	Module name	Export name	Resource global JNDI name
Listener Port	mqjms.module	my/export	mqjms.module.my.export_LP (Note: This is just a name, not JNDI)
Inbound Connection Factory	mqjms.module	my/export	mqjms.module/my/export_LIS_CF
Response Connection Factory	mqjms.module	my/export	mqjms.module/my/export_RESP_CF

Table 31. MQ JMS exports: Names and JNDI names of resources created at installation on the server (continued)

Resource	Module name	Export name	Resource global JNDI name
Receive	mqjms.module	my/export	mqjms.module/my/export_MQ_RECEIVE_D
Send	mqjms.module	my/export	mqjms.module/my/export_MQ_SEND_D
SIB Callback Destination	mqjms.module	my/export	mqjms.module/my/export_MQ_CALLBACK_D
SIB Callback Connection Factory	All modules	my/export	SCA.MQJMS/Callback_CF

Note:

- The resources are created at the server scope. The default scope in the administrative console is cell. You must change the scope in order to locate and administer the resources.
- The SIB callback destination and SIB callback connection factory are SIB JMS resources. The other entries in the table are MQ JMS resources. The two types of resources are administered.

In the alternate case, where the MQ JMS import or export is expecting to find on the server resources that it will use, you must have these resources installed and the import file must contain their JNDI names. The association between the MQ JMS import and the resources will then be made.

Configuring JMS bindings

You can configure JMS import and export bindings to apply special features of the resource. The administrative tasks are performed using the WebSphere administrative console.

Before you begin

You must have permission to make and save changes to the profile on the administrative console.

About this task

The JMS binding must be installed as part of a Service Component Architecture (SCA) application in your server profile.

Procedure

1. Select the SCA module. From the administrative console, click **Applications** → **SCA Modules** and then click the *modulename*. The configuration page opens.
2. Under **Module components**, expand **Imports** or **Exports**. This reveals a list of the installed imports or exports.
3. Expand the import or export and then expand **Binding**. This reveals a list of available bindings to administer.
4. Click on the binding to be administered. The general properties of the binding are displayed:
    - The **Send Resources** category contains the Connection Factory and the Send Destination.

- The **Receive Resources** category contains the Response Connection Factory, the Listener Port, and the Activation Specification.
- The **Advanced Resources** category contains Callback resources and other available resources.

Note: You can also access a resource by typing the JNDI name in the text box. Doing so, however, allows you to enter the name of a resource that is not yet configured.

5. Administer the desired resource:

    a. Click the **Browse...** button to open a window with a list of JNDI names; then, choose the desired JNDI name and click Select.

    b. Click the **Configure...** button to open the corresponding panel referred to by the JNDI name. While most resources can be configured at cluster scope, selecting the **Configure...** option at Listener Port displays a panel showing all listener ports with the cluster Listener Port names for all the members of the given cluster; you can then select one listener port.

    When **Configure...** has been selected, the corresponding server panel will open.

6. When you are ready to administer your settings, click **Apply** or **OK**.

What to do next

The application will be automatically restarted for the JNDI bindings to take effect.

Configuring generic JMS bindings

You can administer generic JMS imports bindings to configure special features of the resource. The administrative tasks are performed using the administrative console.

Before you begin

You must have permission to make and save changes to the profile on the administrative console, and you must have completed the connectivity setup procedure.

About this task

The generic JMS import must be installed as part of a Service Component Architecture (SCA) application in your server profile.

Procedure

1. Select the SCA module. From the administrative console, select **Applications** → **SCA Modules** and then select the *modulename*. The configuration page opens.

2. Under **Module components**, expand **Imports** or **Exports**. This reveals a list of the installed imports or exports.

3. Expand the import or export and then expand **Binding**. This reveals a list of available bindings to administer.

4. Click on the binding to be administered. The general properties of the binding are displayed:

    - The **Send Resources** category contains the Connection Factory and the Send Destination.

- The **Receive Resources** category contains the Response Connection Factory, the Listener Port, and the Activation Specification.
- The **Advanced Resources** category contains Callback resources and other available resources.

Note: You can also access a resource by typing the JNDI name in the text box. Doing so, however, allows you to enter the name of a resource that is not yet configured.

5. Administer the desired resource:
   a. Click the **Browse...** button to open a window with a list of JNDI names; then, select the desired JNDI name. The selected name will populate the appropriate text field.
   b. Click the **Configure...** button to open the corresponding panel referred to by the JNDI name. While most resources can be configured at cluster scope, selecting the **Configure...** option at Listener Port displays a panel showing all listener ports with the cluster Listener Port names for all the members of the given cluster; you can then select one listener port.

      When **Configure...** has been selected, the corresponding WebSphere Application Server panel will open.

6. When you are ready to administer your settings, click **Apply** or **OK**.

What to do next

The application will be automatically restarted for the JNDI bindings to take effect.

Configuring MQ JMS bindings

You can administer MQ JMS bindings to configure special features of the resource. The administrative tasks are performed using the administrative console.

Before you begin

You must have permission to make and save changes to the profile on the administrative console.

The queue and queue manager are not automatically generated; they must be created in WebSphere MQ by your WebSphere MQ administrator.

About this task

The MQ JMS import or export must be installed as part of a Service Component Architecture (SCA) application in your server profile.

Procedure

1. Select the SCA module. From the administrative console, click **Applications** → **SCA Modules** and then click the *modulename*. The configuration page opens.
2. Under **Module components**, expand **Imports** or **Exports**. This reveals a list of the installed imports or exports.
3. Expand the import or export and then expand **Binding**. This reveals a list of available bindings to administer.
4. Click on the binding to be administered. The general properties of the binding are displayed:

- The **Send Resources** category contains the Connection Factory and the Send Destination.
- The **Receive Resources** category contains the Response Connection Factory, the Listener Port, and the Activation Specification.
- The **Advanced Resources** category contains Callback resources and other available resources.

   Note: You can also access a resource by typing the JNDI name in the text box. Doing so, however, allows you to enter the name of a resource that is not yet configured.

5. Administer the desired resource:

   a. Click the **Browse...** button to open a window with a list of JNDI names; then select the desired JNDI name. The selected name will populate the appropriate text field.

   b. Click the **Configure...** button to open the corresponding panel referred to by the JNDI name. While most resources can be configured at cluster scope, selecting the **Configure...** option at Listener Port displays a panel showing all listener ports with the cluster Listener Port names for all the members of the given cluster; you can then select one listener port.

      When **Configure...** has been selected, the corresponding WebSphere Application Server panel will open.

6. When you are ready to administer your settings, click **Apply** or **OK**.

What to do next

The application will be automatically restarted for the JNDI bindings to take effect.

Configuring WebSphere MQ bindings

You can administer WebSphere MQ import and export bindings to tune, or set, special features of the resource. The administrative tasks are performed using the administrative console.

Before you begin

You must have permission to make and save changes to the profile on the administrative console.

The queue and queue manager are not automatically generated and must be created in WebSphere MQ by your WebSphere MQ administrator.

About this task

The WebSphere MQ import must be installed as part of a Service Component Architecture (SCA) application in your server profile.

Procedure

1. Open the default messaging provider settings panel in the administrative console.

   Expand JMS Providers, click WebSphere MQ.

2. Optional: Administer WebSphere MQ connection factories.

   Click on WebSphere MQ connection factory, in the list of additional properties. This panel shows a list of WebSphere MQ connection factories with a summary

of their configuration properties. Click on the MQ connection factory that you want to administer, or click New to create a new connection factory.

Use the subsequent panel to browse or change the configuration properties of the selected connection factory for use with WebSphere MQ as a JMS provider. These configuration properties control how connections are created to associated queues.

You set these properties in the bindings for the resource reference of the application. If you do not want to modify the bindings for an existing application, locate this connection factory in the J2C panels where you can find these properties.

3. Optional: Administer WebSphere MQ queue connection factories.

Click on WebSphere MQ queue connection factories, in the list of addition properties. This panel shows a list of WebSphere MQ queue connection factories with a summary of their configuration properties. Click on the WebSphere MQ queue connection factory that you want to administer, or click New to create a new queue connection factory.

Use the subsequent panel to browse or change the configuration of the selected queue connection factory for use with the WebSphere MQ JMS provider. These configuration properties control how connections are created to associated queues.

A WebSphere MQ queue connection factory is used to create JMS connections to queues provided by WebSphere MQ for point-to-point messaging. Use WebSphere MQ queue connection factory administrative objects to manage queue connection factories for the WebSphere MQ JMS provider.

4. Optional: Administer WebSphere MQ queue destinations.

Click on WebSphere MQ queue destinations, in the list of additional properties. This panel shows a list of the WebSphere MQ queue destinations with a summary of their configuration properties. Click on the queue destination that you want to administer, or click New to create a new WebSphere MQ queue destination.

Use the subsequent panel to browse or change the configuration properties of the selected queue destination for point-to-point messaging with WebSphere MQ as a messaging provider.

A WebSphere MQ queue destination is used to configure the properties of a queue. Connections to the queue are created by the associated queue connection factory for WebSphere MQ as a messaging provider.

5. Save your changes to the master profile and if necessary restart the server.

JMS bindings

A Java Message Service (JMS) provider enables messaging based on the Java Messaging Service. It provides J2EE connection factories to create connections for JMS destinations.

JMS bindings include:

• Service integration bus (SIB) provider binding compliant with JMS JCA 1.5 (*JMS binding*)

• Generic, non-JCA JMS binding, compliant with JMS 1.1 (*generic binding*)

• WebSphere MQ JMS binding, providing JMS provider support for WebSphere MQ and allowing J2EE application interoperability (*WebSphere MQ JMS binding*)

The applications provided through a JMS binding allow a Service Component Architecture (SCA) module to make calls to, and receive messages from, external JMS systems.

Also supported are WebSphere MQ bindings (*WebSphere MQ binding*) which allow native MQ users to handle arbitrary incoming and outgoing message formats (WebSphere MQ required).

The JMS application provides integration through an available JMS JCA 1.5-based resource adapter. The SIB JMS resource adapter is an example of a JCA 1.5-based resource adapter, and complete support is provided for JMS integration of this adapter.

Alternative JMS provider JCA 1.5-based JMS resource adapters are not supported.

Related tasks

"Enabling event sequencing: JMS exports" on page 212
JMS export bindings support event sequencing. In order for events to be handled in the order in which they are received, you must configure properties of the binding.

Generic JMS bindings

The generic JMS import and export bindings provide connectivity to third-party JMS 1.1 compliant providers. Its operation is similar to that of JMS bindings.

The service provided through a JMS binding allows a Service Component Architecture (SCA) module to make calls or receive messages from external systems. The system can be an external JMS system.

The generic JMS binding application provides integration with non-JCA 1.5-compliant JMS providers that support JMS 1.1 and implement the optional JMS Application Server Facility. The generic JMS binding supports those JMS providers that do not support JCA 1.5 but do support the Application Server Facility of the JMS 1.1 specification, including Oracle AQ, TIBCO, SonicMQ, WebMethods, BEA WebLogic, and WebSphere MQ. SIB is not supported, as SIB is a JCA 1.5 JMS provider.

A user would use this generic application when integrating with a non-JCA 1.5-compliant JMS-based system within an SCA environment. The target external applications can then receive messages and send messages to integrate with an SCA component.

Related tasks

"Enabling event sequencing: generic JMS exports" on page 214
Generic JMS export bindings support event sequencing. In order for events to be handled in the order in which they are received, you must configure properties of the binding.

WebSphere MQ JMS bindings

The WebSphere MQ JMS binding provides integration with WebSphere MQ JMS-based providers.

Use the WebSphere MQ JMS provider applications to integrate directly with external JMS or MQ JMS systems from your server environment. This eliminates the need to use MQ Link or Client Link features of the Service Integration Bus.

Importing *from* WebSphere MQ utilizes a destination that the data will be sent to and a destination where the reply can be received. Conversion for the data to and from a JMS message is accomplished through the JMS data binding edge component.

Exporting *to* WebSphere MQ utilizes a destination where the request can be received and the response can be sent. The conversion for the data to and from a JMS message is done through the JMS data binding. The function selector provides a mapping to the operation on the target component to be invoked.

Related concepts

"WebSphere MQ bindings" on page 183
The WebSphere MQ application provides Service Component Architecture connectivity with WebSphere MQ applications.

Related tasks

"Enabling event sequencing: WebSphere MQ JMS exports" on page 213
WebSphere MQ JMS export bindings support event sequencing. In order for events to be handled in the order in which they are received, you must configure properties of the binding.

WebSphere MQ bindings

The WebSphere MQ application provides Service Component Architecture connectivity with WebSphere MQ applications.

Use the WebSphere MQ provider applications to integrate directly with a WebSphere MQ-based system from your server environment. This eliminates the need to use MQ Link or Client Link features of the Service Integration Bus.

Importing *from* WebSphere MQ utilizes a destination that the data will be sent to and a destination where the reply can be received. Conversion for the data to and from an MQ message is accomplished through the MQ header and body data binding.

Exporting *to* WebSphere MQ utilizes a destination where the request can be received and the response can be sent. The conversion for the data to and from an MQ message is done through the MQ header and body data binding. The function selector provides a mapping to the operation on the target component to be invoked.

Related concepts

"WebSphere MQ JMS bindings" on page 174
The WebSphere MQ JMS binding provides integration with WebSphere MQ JMS-based providers.

The WebSphere Transformation Extender data binding

Data that is brought into an SCA module via an export can be in a wide range of formats, such as COBOL or EDI. These data must be converted into a data object before processing with any server component. Similarly, data passing from your server through an import must be converted to a format understood by the external system. WebSphere Transformation Extender can be used to perform these transformations.

The exports and imports at the edges of your modules are responsible for converting native data to data objects and vice versa. Exports and imports contain

data bindings for this purpose. All exports and imports have data bindings that convert XML data to and from Data Objects. If you have a format other than XML you can specify your own custom data binding for JMS, EIS, HTTP and Native MQ exports and imports. For SCA and Web Service imports and exports, the data binding is not configurable.

To transform native data in your imports and exports to a variety of data formats, you can write your own data binding, or you can use the WebSphere Transformation Extender data binding to convert native data to data object, and vice versa, at the edges of your modules.

WebSphere Transformation Extender is a transaction-oriented data integration solution that automates the transformation of high-volume, complex transactions without the need for hand-coding. It supports real-time integration of data from multiple applications, databases, messaging middleware and communications technologies across the enterprise. WebSphere Transformation Extender provides a meta-data driven, tool-oriented approach to data transformation.

WebSphere Transformation Extender is a separately licensed product and is not included as part of WebSphere Process Server.

For WebSphere Process Server to work with WebSphere Transformation Extender you must use the WebSphere Transformation Extender installer for WebSphere Process Server.

The scheme below represents configuring your export to use WebSphere Transformation Extender with the WebSphere Transformation Extender data binding to convert the incoming data to a Data Object.
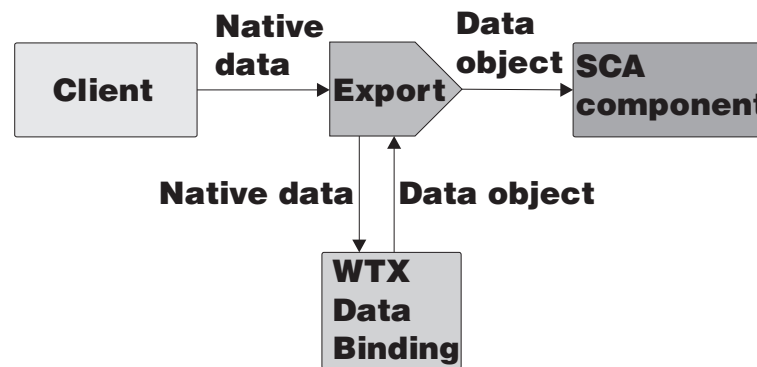
The client delivers the data to the export in some native format. The export then passes this data in the same native format to WebSphere Transformation Extender via the WebSphere Transformation Extender Data Binding. WebSphere Transformation Extender converts the data to Data Object format and returns it to the export. The export passes the Data Object on to the relevant SCA component.

The scheme below represents configuring your import to use WebSphere Transformation Extender with the WebSphere Transformation Extender Data Binding to convert the outgoing data from Data Object format into the native format required by the client.
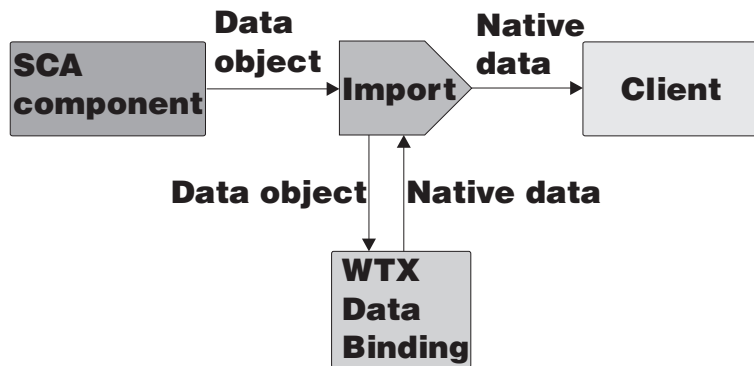
The SCA component sends a Data Object to the import. The import passes this Data Object to WebSphere Transformation Extender via the WebSphere Transformation Extender Data Binding. WebSphere Transformation Extender converts the information to the native format of the client and this information is returned to the import. The import sends the native data to the client.

Note: For more information about using WebSphere Transformation Extender see the WebSphere Transformation Extender product library.

Support for the WebSphere Transformation Extender data binding

WebSphere Transformation Extender can be used to convert data between a variety of formats and the Data Object required by the server. The details of when such a binding is supported are provided.

Supported platforms

The WebSphere Transformation Extender data binding is supported only on platforms that are supported by both WebSphere Transformation Extender and WebSphere Process Server. See the supported platform lists for both products here: WebSphere Transformation Extender system requirements and WebSphere Process Server.

Note: WebSphere Transformation Extender is a separately licensed product and is not included as part of WebSphere Process Server.

i5/OS WebSphere Transformation Extender is not supported on i5/OS and so this solution does not apply to the i5/OS platform.

WebSphere Transformation Extender must be at version 8.2 or above.

Imports and Exports

The following imports and exports support the use of WebSphere Transformation Extender for data transform:
- JMS
- Generic JMS
- WebSphere MQ JMS
- Native MQ (body data binding only)
- EIS Flat File
- EIS FTP

- EIS Email
- HTTP

The WebSphere Transformation Extender data binding is applicable to all imports and exports listed above. These data bindings can be customized. See the WebSphere Integration Developer documentation for more information.

The WebSphere Transformation Extender Data Binding

The WebSphere Transformation Extender Data Binding invokes WebSphere Transformation Extender. WebSphere Transformation Extender uses a map to convert data from one format to another. The map provides the system with the details of the conversion. You can utilize existing WebSphere Transformation Extender maps if they are applicable to the solution.

The WebSphere Transformation Extender data binding invokes one of your WebSphere Transformation Extender maps in the context of a WebSphere Transformation Extender Engine. The data binding sends the WebSphere Transformation Extender map name and input data to the WebSphere Transformation Extender engine which performs the conversion from one format to another. The result of the transformation is sent back to the binding which publishes it to the export or import.

On the export side, the WebSphere Transformation Extender data binding invokes a WebSphere Transformation Extender map to transform native data to XML. The data binding then deserializes the XML to a data object. An import also uses the data binding in this way on the response message in a request/response scenario.

On the import end, the data binding serializes a data object to XML and feeds it to a WebSphere Transformation Extender map that converts it to native data, which is then published. An export also uses the data binding in this way when

For more information on creating WebSphere Transformation Extender maps please refer to the WebSphere Transformation Extender documentation.

For more information on creating map naming conventions, usage of maps in SCA modules and configuring the WebSphere Transformation Extender Data Binding refer to the WebSphere Integration Developer documentation.

Considerations before using the WebSphere Transformation Extender data binding

The WebSphere Transformation Extender data binding is an ideal choice when you have non-XML data coming into, or leaving, your WebSphere Process Server. Note that you must have WebSphere Transformation Extender available.

You must install WebSphere Process Server before you install the WebSphere Transformation Extender for WebSphere Process Server. You must have a valid license for WebSphere Transformation Extender. WebSphere Transformation Extender is an independent product and has to be installed separately.

For your server to work with WebSphere Transformation Extender you must use the WebSphere Transformation Extender installer for WebSphere Process Server. In addition to installing WebSphere Transformation Extender, this process also installs the WebSphere Transformation Extender Java client libraries as an OSGi bundle in the WebSphere Process Server product so that it is accessible to WebSphere Process Server.

Installing WebSphere Transformation Extender for WebSphere Process Server is a two-step process:

1. Run the WebSphere Transformation Extender for the Software Developers Kit (SDK). This installs the SDK and the OSGi bundle for WebSphere Transformation Extender in the WebSphere Process Server plugins.

2. Run the installer for WebSphere Transformation Extender Design Studio. This step provides you with all the tools required to design and create tree-types and maps.

Note: You must perform the first step on every node where you will utilize the WebSphere Transformation Extender data binding. The second step is only required on nodes where you want to create and edit maps.

The installation of WebSphere Transformation Extender should have configured the system path to point to WebSphere Transformation Extender. Verify that the installation location (e.g., C:\Program Files\IBM\WebSphere Transformation Extender 8.2) is part of your system path.

The WebSphere Transformation Extender data binding calls WebSphere Transformation Extender using Java Native Interface (JNI). This may have performance implications on your system.

The WebSphere Transformation Extender data binding has the following memory requirements:

- For transforming from native data to business object, the memory required is at least twice the size of the native data plus twice the size of the serialized business object.

- For transforming from business object to native data, the memory required is at least twice the size of the serialized business object plus twice the size of native data.

The WebSphere Transformation Extender data binding is an ideal choice when you have non-XML data entering, or leaving, your WebSphere Process Server environment. For XML data you should use the XML data binding for JMS, WebSphere MQ and HTTP imports and exports, and use the XMLDataHandler for EIS bindings.

Additional considerations for using the WebSphere Transformation Extender data binding.

- If your data is in an industry standard format supported by WebSphere Transformation Extender then the WebSphere Transformation Extender data binding is an ideal solution. See related links for a list of industry packs.

- For formats such as delimited, csv and fixed-width, use the data bindings and data handlers that are supplied as part of WebSphere Process Server rather than the WebSphere Transformation Extender data binding.

- For COBOL, use the COBOL data binding generator to generate a COBOL data binding.

- For any other custom format make an evaluation based on the memory requirements and the ease of developing a WebSphere Transformation Extender map compared to developing a custom data binding.

Setting the data binding descriptor

Use the WebSphere Transformation Extender data binding descriptor instead of the configuration properties of WebSphere Transformation Extender data binding when you want to configure the WebSphere Transformation Extender data binding at an instance level. You must set the WebSphere Transformation Extender data binding descriptor in specific places for the various imports and exports.

About this task

The key to the invocation of WebSphere Transformation Extender is to determine the map name to be invoked. You can do this by configuring the properties of the WebSphere Transformation Extender data binding in WebSphere Integration Developer. The map set in the properties of the binding is then applicable to all instances of data that enter the export or import for that operation. For example, consider an export with an operation `Create(Customer)`. If you want every invocation of this operation to use the same WebSphere Transformation Extender map for each customer, then configuring the properties of the WebSphere Transformation Extender data binding in WebSphere Integration Developer is an appropriate choice.

However, you might want to invoke a different map depending on the incoming customer object. Such a scenario cannot be configured in the properties of the data binding and you must instead use the data binding descriptor.

You must put the WebSphere Transformation Extender data binding descriptor in the following locations, depending on what type of import or export you are employing:

- The WebSphere Transformation Extender data binding for JMS expects the descriptor to be set in the **JMSType** or in a **JMS property** called **DataBindingDescriptor**.

  For JMS Exports:

  – For a request, the client is required to either set the **JMSType** in the incoming message or set a **JMS property** called **DataBindingDescriptor**.

    These properties are set when the client sends a message to your server.

  – For a response, you cannot configure the binding at the instance level; you must use configuration properties to invoke WebSphere Transformation Extender.

  For JMS Imports:

  – For a request, you cannot configure the binding at the instance level; you must use configuration properties to invoke WebSphere Transformation Extender.

  – For a response, the client is required to either set the **JMSType** in the incoming message or set a **JMS property** called **DataBindingDescriptor**.

    These properties are set when the client sends a message to your server.

- The WebSphere Transformation Extender Native MQ data binding expects the descriptor to be set in the MQRFH2 headers.

  For Native MQ Exports:

  – For a request, the client must set the **DatabindingDescriptor** property in the MQRFH2 header.

  – For a response, you cannot configure the binding at the instance level; you must use configuration properties of the binding itself to invoke WebSphere Transformation Extender.

For Native MQ Imports:

- For a request, you cannot configure the binding at the instance level; you must use configuration properties of the binding itself to invoke WebSphere Transformation Extender.
- For a response, the client must set the **DatabindingDescriptor** property in the MQRFH2 header.

- The WebSphere Transformation Extender HTTP data binding expects a custom header called **DatabindingDescriptor**. If the header is not present, the binding looks in the parameters of the URL of the BusinessObject and StreamType or the map name parameter. For example: https://host[*port*]/*sca module name*/*exportname*/?businessObject=*value1*&contentType=*value2*

  For HTTP Exports:

  - For a request, the custom header **DatabindingDescriptor** must be set by the client. Alternatively, you can set custom properties on the URL.
  - For a response, you cannot configure the binding at the instance level and must use configuration properties of the binding itself to invoke WebSphere Transformation Extender.

  For HTTP Imports:

  - For a request, you cannot configure the binding at the instance level and must use configuration properties of the binding itself to invoke WebSphere Transformation Extender.
  - For response, the custom header **DatabindingDescriptor** must be set by the client.

WebSphere Transformation Extender data binding descriptor

The WebSphere Transformation Extender data binding descriptor is a URI that describes the incoming message.

Purpose

The WebSphere Transformation Extender data binding descriptor is set by the client in the headers of the incoming message. It provides the information required to associate the incoming message with the correct map in the WebSphere Transformation Extender.

Using the WebSphere Transformation Extender data binding descriptor is one of two ways that you can set the invocation of the map. The alternative is to configure the properties of the WebSphere Transformation Extender data binding in WebSphere Integration Developer. However, you should use the WebSphere Transformation Extender data binding descriptor when you want to use a different map for each invocation of the operation.

For example, consider an export with an operation `Create(Customer)`. If you want every invocation of this operation to use the same WebSphere Transformation Extender map for each customer, then configuring the properties of the WebSphere Transformation Extender data binding in WebSphere Integration Developer is an appropriate choice.

However, you might want to invoke a different map depending on the customer. In this case the operation is invoked with customer and uses a different map for every invocation. Such a scenario cannot be configured in the properties of the data binding and you must instead use the data binding descriptor.

Syntax

The WebSphere Transformation Extender data binding descriptor has the following syntax

databinding://*domain***/***property***?***queryParameters*
>   *domain* is **WTX** for this data binding.

>   *property* has the value **map** in this case.

>   *queryParameters* is either name=*mapname* where *mapname* is the name of the map required, or businessObject=*Customer*&contentType=*format*, where *Customer* is the name of the Business Object and *format* is the format of the data stream (e.g., COBOL, EDI, etc.).

Sample

Examples of a WebSphere Transformation Extender data binding descriptor are shown below:

```
databinding://WTX/map?name=WTX/CustomerToCOBOL
```

Note: A directory called WTX must be in the top level of the module and must contain any maps that are required by this binding.

```
databinding://WTX/map?businessObject=Customer&contentType=COBOL
```

```
databinding://WTX/map?contentType=COBOL
```

Setting the binding descriptor for an EIS binding
The data binding descriptor cannot be set for EIS import or export binding.

About this task

If you are using EIS imports and exports, you must configure the interaction between your binding and WebSphere Transformation Extender using the configuration properties within the EIS binding definition. You cannot configure the connection at the instance level.

Setting the binding descriptor for a JMS Import binding
A JMS Import binding can be passed the WebSphere Transformation Extender data binding descriptor in either the JMSType or JMSProperties part of the message.

Before you begin

Use the WebSphere Transformation Extender JMS data binding for JMS, Generic JMS and MQ-JMS Imports.

You can only set the WebSphere Transformation Extender data binding descriptor on the response message for a JMS Import.

About this task

Set the data binding descriptor on the response message for a JMS Import when you want to configure the messages individually.

If you do not need to configure messages individually it is recommended that you use the custom properties within the binding itself to make the association with WebSphere Transformation Extender.

Procedure

1. The request message cannot be configured at the instance level. You must use the custom properties of the binding to make the association with the specific WebSphere Transformation Extender maps that you want to use.

2. Configure the response message. There are two ways to configure the response:

Option	Example
Set the JMSType in the incoming message	`jmsMessage.setJMSType("databinding://WTX/map?businessObject=Customer&contentType=COBOL");` Note that line breaks have been added to this example.
Set the JMS property called DataBindingDescriptor	`jmsMessage. setStringProperty ("DataBindingDescriptor","databinding://WTX/map?businessObject=Customer&contentType=COBOL");` Note that line breaks have been added to this example.

These properties are set when the client sends a message to WebSphere Process Server.

Setting the binding descriptor for a JMS Export binding

A JMS Export binding can be passed the WebSphere Transformation Extender data binding descriptor in either the JMSType or JMSProperties part of the message.

Before you begin

Use the WebSphere Transformation Extender JMS data binding for JMS, Generic JMS and MQ-JMS Exports.

You can only set the WebSphere Transformation Extender data binding descriptor on the response message for a JMS Export.

About this task

Set the data binding descriptor on the response message for a JMS Export when you want to configure the messages individually.

If you do not need to configure messages individually, it is recommended that you use the custom properties within the binding itself to make the association with WebSphere Transformation Extender.

Procedure

1. Configure the request message. There are two ways to configure the request message:

Option	Example
Set the JMSType in the incoming message	`jmsMessage.setJMSType("databinding://WTX/map?businessObject=Customer&contentType=COBOL");` Note that line breaks have been added to this example.
Set a JMS property called DataBindingDescriptor	`jmsMessage. setStringProperty ("DataBindingDescriptor", "databinding://WTX/map?businessObject=Customer&contentType=COBOL");` Note that line breaks have been added to this example.

2. The response message cannot be configured at the instance level. In order to use a WebSphere Transformation Extender transformation on a JMS Export response message you must set the configuration properties of your export.

Setting the binding descriptor for a native MQ Import

If you want to configure each message to interact in a specific way with WebSphere Transformation Extender you must configure the message to use the WebSphere Transformation Extender MQ data binding via the WebSphere Transformation Extender data binding descriptor.

Before you begin

Use the WebSphere Transformation Extender MQ data binding for native MQ Imports.

You can only set the WebSphere Transformation Extender data binding descriptor on the response message for a native MQ Import.

About this task

Set the data binding descriptor on the response message for a native MQ Import when you want to configure the messages individually.

If you do not need to configure messages individually it is recommended that you use the custom properties within the binding itself to make the association with WebSphere Transformation Extender.

Procedure

1. The request message cannot be configured at the instance level. You must use the custom properties of the binding to make the association with the specific WebSphere Transformation Extender maps that you want to use.
2. Configure the response message. Set the DataBindingDescriptor in the MQRFH2 header of the incoming message

What to do next

These properties are set when the MQ client sends a message to WebSphere Process Server.

Setting the binding descriptor for a native MQ Export

If you want to configure each message to interact in a specific way with WebSphere Transformation Extender you must configure the message to use the WebSphere Transformation Extender MQ data binding via the WebSphere Transformation Extender data binding descriptor.

Before you begin

Use the WebSphere Transformation Extender MQ data binding for native MQ Exports.

You can only set the WebSphere Transformation Extender data binding descriptor on the request message for a native MQ Export.

About this task

Set the data binding descriptor on the request message for a native MQ Export when you want to configure the messages individually.

If you do not need to configure messages individually it is recommended that you use the custom properties within the binding itself to make the association with WebSphere Transformation Extender.

Procedure

1. Configure the request message. Set the DataBindingDescriptor in the MQRFH2 header of the incoming message
2. The request message cannot be configured at the instance level. You must use the custom properties of the binding to make the association with the specific WebSphere Transformation Extender maps that you want to use.

What to do next

These properties are set when the MQ client sends a message to WebSphere Process Server.

Setting the binding descriptor for an HTTP Import

If you want to configure each HTTP message that is sent to the same operation on the import to use a different WebSphere Transformation Extender map, you must configure the message to use the WebSphere Transformation Extender data binding descriptor.

Before you begin

Use the WebSphere Transformation Extender HTTP data binding for HTTP Imports.

You can set the WebSphere Transformation Extender data binding descriptor only on the response message for an HTTP Import.

About this task

Set the data binding descriptor on the response message for an HTTP Import when you want to configure the messages individually.

If you do not need to configure messages individually it is recommended that you use the custom properties within the binding itself to make the association with WebSphere Transformation Extender.

Procedure

1. The request message cannot be configured at the instance level. You must use the custom properties of the binding to make the association with the specific WebSphere Transformation Extender maps that you want to use.
2. Configure the response message.

   Create a custom HTTP header called DataBindingDescriptor in the HTTP Import. For example:

   databinding://WTX/map?businessObject=Customer&contentType=EDI

Setting the binding descriptor for an HTTP Export

If you want to configure each HTTP message that is published to the same operation on the export to use a different WebSphere Transformation Extender map, you must configure the message to use the WebSphere Transformation Extender data binding descriptor.

Before you begin

Use the WebSphere Transformation Extender data binding for HTTP Exports.

You can set the WebSphere Transformation Extender data binding descriptor only in the request message for an HTTP Export.

The HTTP client sending requests to the HTTP export sets the data binding descriptor in the request message header or on the URL.

About this task

Set the data binding descriptor on the request message for an HTTP Export when you want to configure the messages individually.

If you do not need to configure messages individually it is recommended that you use the custom properties within the binding itself to make the association with WebSphere Transformation Extender.

Procedure

1. Configure the request message. There are two ways to set the data binding descriptor in the request message:

Option	Example
Create a custom HTTP Header called DataBindingDescriptor in the HTTP Export.	databinding://WTX/ map?businessObject=Customer& contentType =EDI
Add additional query parameters on the URL.	http://www.ibm.com/Export1/ map?businessObject=Customer &contentType=EDI

In general it is impractical to create the custom headers; setting query parameters on the URL is the recommended method.

2. The response message cannot be configured at the instance level. You must use the custom properties of the binding to make the association with the specific WebSphere Transformation Extender maps that you want to use.

Chapter 9. Adapters

WebSphere Process Server supports two types of adapters: WebSphere Adapters and WebSphere Business Integration Adapters. Adapters enable business applications to act as services by connecting them to diverse enterprise information systems (EISs), such as databases, enterprise resource planning systems, file systems, and e-mail systems.

With the help of an adapter, the application and EIS can "talk to each other," or, in other words, send and retrieve information in a consistent way. To allow your applications to operate as services, the adapter connects them to WebSphere Process Server, which powers your Service Oriented Architecture (SOA). With an adapter, you no longer need to provide proprietary connection utilities (or write custom connection utilities) for each EIS or application server.

Differences between WebSphere Adapters and WebSphere Business Integration Adapters

Both WebSphere Adapters and WebSphere Business Integration Adapters mediate communication between components and enterprise information systems. The two types of adapter differ in several respects including: their integration, their JCA-compliance, their data models, and the management of their connectivity.

There are several differences between WebSphere Adapters and WebSphere Business Integration Adapters. These distinctions are most important during development of applications. When deploying applications to a running server, the nature of the adapters used affects some of the steps which need to be followed.

Adapters provide communication mechanisms between enterprise information systems (EISs) and WebSphere applications. To illustrate the operation of the adapters, figures 1 and 2 provide details of the communication between the server and the EIS for the two types of adapter.
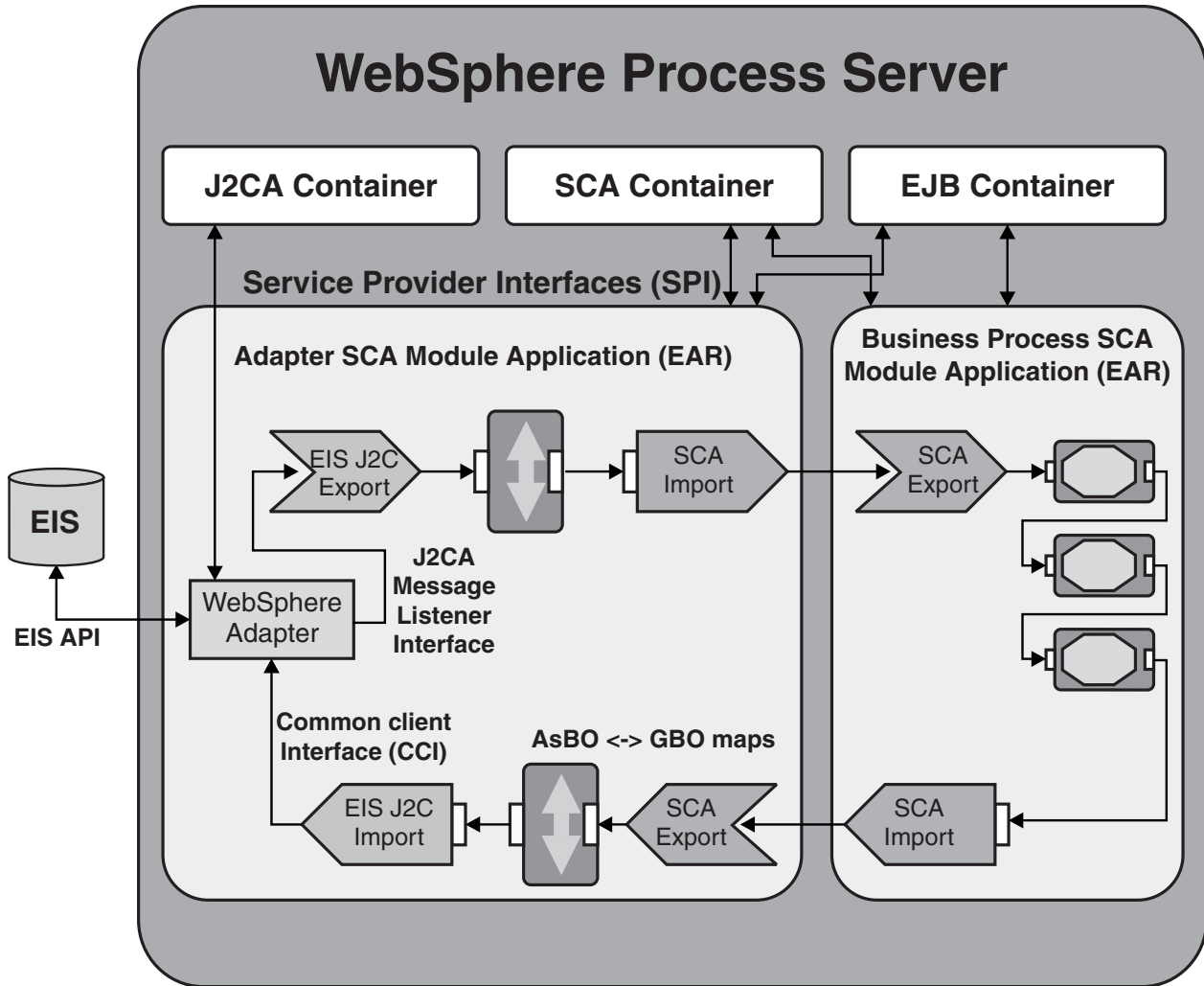
Figure 24. Detailed schematic of a WebSphere Adapter.

Figure 1 depicts a WebSphere Adapter managing the connectivity between a J2EE component supported by the server and the EIS. The WebSphere Adapter resides inside the server.
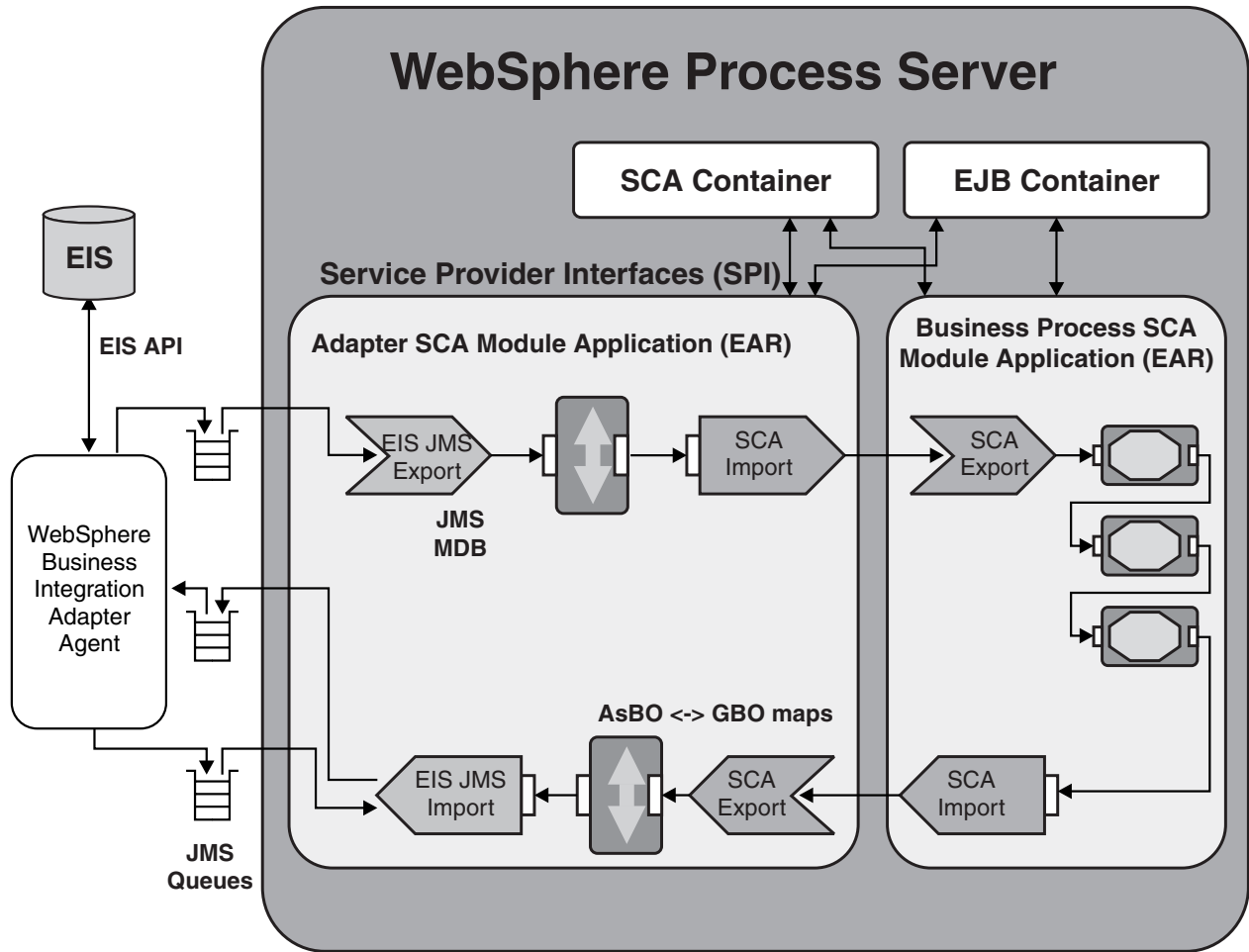
Figure 25. Detailed schematic of a WebSphere Business Integration Adapter.

Figure 2 shows a WebSphere Business Integration Adapter mediating communication between the WebSphere Integration Broker and the EIS. The integration broker communicates with the WebSphere Business Integration Adapter through the use of a Java Message Service (JMS) transport layer.

Table 1 shows the differences between the two types of adapter.

Table 32. Differences between WebSphere Adapters and WebSphere Business Integration Adapters

Feature	WebSphere Adapters	WebSphere Business Integration Adapters
JCA Compliance	Fully JCA compliant (version 1.5).	Not JCA-compliant.
Connectivity Manager	Rely on standard JCA contracts to manage life cycle tasks such as starting and stopping.	Rely on WebSphere Adapter Framework to manage connectivity.
Event Notification	Use an EventStore subclass to retrieve events from an EIS.	Manage event notification using a pollFor Events method.

Table 32. Differences between WebSphere Adapters and WebSphere Business Integration Adapters (continued)

Feature	WebSphere Adapters	WebSphere Business Integration Adapters
Request Processing	Clients directly invoke one of several interaction contracts to query or modify data in the EIS.	Rely on an integration server and the WebSphere Adapter Framework to initiate and help process requests.
Data Models	Use an Enterprise Metadata Discovery (EMD) utility to parse an EIS and develop Service Data Objects (SDOs) and other useful artifacts. The EMD is part of the WebSphere Adapter implementation.	Use a separate Object Discovery Agent (ODA) to introspect an EIS and generate business object definition schemas.
Integration	Run on the server.	Reside outside the server. The server or integration broker communicates with the adapter via a Java Message Service (JMS) transport layer.

WebSphere Adapters are the recommended product.

WebSphere Adapters

WebSphere Adapters, also known as resource adapters, enable managed, bidirectional connectivity between enterprise information systems (EISs) and J2EE components supported by the server.

WebSphere Adapters, which are preferred over WebSphere Business Integration Adapters, are covered elsewhere in this information library.

Where to find more information

To learn about configuring and using WebSphere Adapters, see Configuring and using adapters in the WebSphere Integration Developer. information center. In the adapter guide for your adapter, expand the navigation and click on **Administering the adapter module**.

For general information about adapters, see Accessing external services with adapters in the WebSphere Integration Developer information center.

WebSphere Business Integration Adapters

WebSphere Business Integration Adapters consist of a collection of software, Application Programming Interfaces (APIs), and tools to enable applications to exchange business data through an integration broker.

Each business application requires its own application-specific adapter to participate in the business integration process. You can install, configure, and test the adapter using current WebSphere Business Integration Adapter Framework and Development Kit System Manager tools. You can use WebSphere Integration Developer to import existing business objects and connector configuration files, to generate artifacts, and to assemble the solution for WebSphere Process Server.

Operational commands for the WebSphere Business Integration Adapters are part of the WebSphere Process Server administrative console.

The tasks in this section focus on WebSphere Business Integration Adapters in specific relation to WebSphere Process Server. These tasks are to perform installation and configuration procedures for the adapter, set up administration of the adapter, and use the administrative console to manage the adapter.

Where to find more information

For more information about working with these adapters, see the WebSphere Business Integration Adapter information center.

Installing and configuring the adapter

You must perform installation and configuration procedures in order for the WebSphere Business Integration Adapter to work with WebSphere Process Server.

Procedure

1. Install the adapter.
    a. Follow the procedures outlined at Installing WebSphere Business Integration Adapters products, which describe how to install WebSphere Business Integration Adapters.
    b. Follow any additional required procedures that are specific to your particular adapter by going to the WebSphere Business Integration Adapters documentation and expanding the navigation under **Adapters** to reveal your specific adapter, where you will find any additional installation tasks.
2. Configure your adapter by going to the WebSphere Business Integration Adapters documentation, expanding the navigation under **Adapters** to reveal your specific adapter, and following that adapter's configuration instructions. The configuration procedure will generate the required artifacts.
3. Install the application EAR file by following the instructions for *Installing a module on a production server* in the Developing and Deploying Modules PDF..

Setting up administration of a WebSphere Business Integration Adapter

You must perform several administrative functions before you can manage a WebSphere Business Integration Adapter.

Before you begin
- You must be familiar with the procedures outlined in Installing WebSphere Business Integration Adapters products.
- You must have installed the application EAR file to create the artifacts required for the WebSphere Business Integration Adapter before you perform this task.

About this task

In order to have administrative control over a WebSphere Business Integration Adapter, perform the following administrative functions.

Procedure

1. Create a Queue Connection Factory.

    From the top level of the administrative console follow these steps:
    a. Expand **Resources**.
    b. Expand **JMS**.
    c. Select **Queue connection factories**.

d. Click on **New** to create a new JMS queue connection factory. Select the
      scope level that matches the Administration Input/Output Queues' scope
      level.
   e. Choose the JMS resource provider. Select **Default messaging provider**.
      Click **OK**.
   f. Accept all the default values with the following exceptions:
      - Name: QueueCF
      - JNDI Name: jms/QueueCF
      - Bus. Name: *Your Bus name*
   g. Complete the creation of your new JMS queue connection factory.
      Click **OK**.
      A message window appears at the top of the JMS queue connection factory
      panel.
   h. Apply the changes that you have made at the local configuration level to
      the master configuration.
      Click **Save** in the message window.
2. Create a WebSphere Business Integration Adapter resource
   From the top level of the administrative console follow these steps:
   a. Expand **Resources**.
   b. Open the WebSphere Business Integration Adapters panel.
      Select **WebSphere Business Integration Adapters**.
   c. Create a new WebSphere Business Integration Adapter.
      Click **New**.
   d. Accept all the default values with the following exceptions:
      - Name: EISConnector
      - Queue Connection Factory JNDI Name: jms/QueueCF
      - Administration Input Queue JNDI Name: *connectorName/AdminInQueue*
      - Administration Output Queue JNDI Name: *connectorName/*
        AdminOutQueue
   e. Complete the creation of the WebSphere Business Integration Adapter.
      Click **OK**.
      A message window appears at the top of the WebSphere Business
      Integration Adapters panel.
   f. Apply the changes that you have made at the local configuration level to the
      master configuration.
      Click **save** in the message window.
3. Enable the WebSphere Business Integration Adapter Service.
   From the top level of the administrative console follow these steps:
   a. Expand **Servers**.
   b. Select **Application Servers**.
   c. From the list of servers select a server where the WebSphere Business
      Integration Adapter Service is to be enabled.
      Click on the name of the server that hosts the resources of interest.
   d. Select **WebSphere Business Integration Adapter Service**.
      Under the Business Integration subheading on the Configuration tab select
      WebSphere Business Integration Adapter Service.
   e. Ensure that the **Enable Service at startup** check box is selected.

f. Click **OK**.

   A message window appears at the top of the WebSphere Business Integration Adapters panel.

g. Repeat steps 3c on page 264 to 3f for each server on which the WebSphere Business Integration Adapter Service is to be enabled.

h. Apply the changes that you have made at the local configuration level to the master configuration.

   Click **save** in the message window.

Note: When you enable or disable a WebSphere Business Integration Adapter service, you must restart the server in order for the changes to take effect.

Managing the WebSphere Business Integration Adapter

You can manage a running WebSphere Business Integration Adapter from the administrative console.

Before you begin

The WebSphere Business Integration Adapter must be running in order to be managed.

About this task

Use the following procedures to manage your resources and to perform various administrative actions on them.

Procedure

1. Select the resource or resources to manage.

   From the top level of the administrative console follow these steps:

   a. Expand **Servers**.

   b. Select **Application Servers**.

   c. From the list of servers select the server where the resources you intend to manage reside.

      Click on the name of the server that hosts the resources of interest.

   d. Select **WebSphere Business Integration Adapter Service**.

      Under the Business Integration subheading on the Configuration tab select **WebSphere Business Integration Adapter Service**.

   e. Select **Manage the WebSphere Business Integration Adapter resources**.

   f. From the list of resources choose those that you want to manage.

      Select the check boxes associated with the resources you intend to manage.

2. Manage the selected resources.

   Click one of the command buttons to act upon the selected resources.

Command	Description
Deactivate	Changes the status of the selected resources from active to paused or inactive.
Activate	Changes the status of the selected resources from inactive to active.
Suspend	Changes the status of the selected resources from active to paused.

Command	Description
Resume	Changes the status of the selected resources from paused to active.
Shutdown	Changes the status of the selected resources from active to unavailable.

Chapter 10. Administering applications and application services

Applications for WebSphere Process Server involve similar administration tasks and interfaces as J2EE applications for WebSphere Application Server, with some additional tasks specifically relating to service applications, WebSphere MQ destinations, and other resources. Administering application services includes business processes and tasks, business rules, and schedules.

Administering service applications

You can manage service applications from the WebSphere Process Server administrative console. Service applications provide services, and have an associated Service Component Architecture (SCA) module.

About this task

The kind of SCA modules that WebSphere Process Server supports are called mediation modules. Mediation modules provide a simple way to change the format, content or target of service requests and responses.

Getting started with service applications

After deploying service applications you can view and manage components of your service applications.

About this task

You can view and manage the applications and the associated Service Component Architecture (SCA) modules. The kind of SCA modules that WebSphere Process Server supports are called mediation modules.

You can list all the SCA modules that you have deployed, and display details of how the SCA modules connect to service requesters and service providers.

Application integration environment

A service application has an associated Service Component Architecture (SCA) module. You deploy service applications, to WebSphere Process Server, within Enterprise ARchive (EAR) files.

Deploying a service application

The process of deploying an EAR file containing a service application is the same as the process of deploying any EAR file. You can modify values for mediation parameters at deployment time. After you have deployed an EAR file containing an SCA module, you can view details about the service application and its associated SCA module. You can see how an SCA module is connected to service requesters and service providers. SCA modules are connected to service requesters through exports, and to service providers through imports.

Viewing SCA module details

The SCA module details that you can view depend upon the SCA module. They include the following attributes.

- SCA module name
- SCA module description
- Associated application name
- SCA module imports:
  - Import interfaces are abstract definitions that describe how an SCA module accesses a service.
  - Import bindings are concrete definitions that specify the physical mechanism by which an SCA module accesses a service. For example, using SOAP/HTTP.
- SCA module exports:
  - Export interfaces are abstract definitions that describe how service requesters access an SCA module.
  - Export bindings are concrete definitions that specify the physical mechanism by which a service requester accesses an SCA module, and indirectly, a service.
- SCA module properties

Learning about service applications

Service applications provide services, and have an associated Service Component Architecture (SCA) module. SCA modules encapsulate services, so you can make changes to services without impacting users of the service. The kind of SCA modules that WebSphere Process Server supports are called mediation modules.

About this task

Mediation modules:

Mediation modules are Service Component Architecture (SCA) modules that can change the format, content, or target of service requests.

Mediation modules operate on messages that are in-flight between service requesters and service providers. You are able to route messages to different service providers and to amend message content or form. Mediation modules can provide functions such as message logging, and error processing that is tailored to your requirements.

You can change certain aspects of mediation modules dynamically, from the WebSphere Process Server administrative console, without having to redeploy the module.

Components of mediation modules

Mediation modules contain the following items:

- Imports: which define interactions between SCA modules and service providers. They allow SCA modules to call external services as if they were local. You can view mediation module imports from WebSphere Process Server and modify the binding.

- Exports: which define interactions between SCA modules and service requesters. They allow an SCA module to offer a service and define the external interfaces (access points) of an SCA module. You can view mediation module exports from WebSphere Process Server.
- SCA components: which are building blocks for SCA modules such as mediation modules. You can create and customize SCA modules and components graphically, using WebSphere Integration Developer. After you deploy a mediation module you can customize certain aspects of it from the WebSphere Process Server administrative console, without having to redeploy the module.

  Usually, mediation modules contain a specific type of SCA component called a *mediation flow component*. Mediation flow components define mediation flows. A mediation module can contain only one mediation flow component.

  A mediation flow component can contain none, one, or a number of mediation primitives. WebSphere Process Server supports a supplied set of mediation primitives that provide functionality for message routing and transformation. If you need additional mediation primitive flexibility, you can use the Custom Mediation primitive to call custom logic.

  The purpose of a mediation module that does not contain a mediation flow component is to transform service requests from one protocol to another. For example, a service request might be made using SOAP/JMS but might need transforming to SOAP/HTTP before sending on.

  Note: You can view and make certain changes to mediation modules from WebSphere Process Server. However, you cannot view or change the SCA components inside a WebSphere Process Server module. Use WebSphere Integration Developer to customize SCA components.
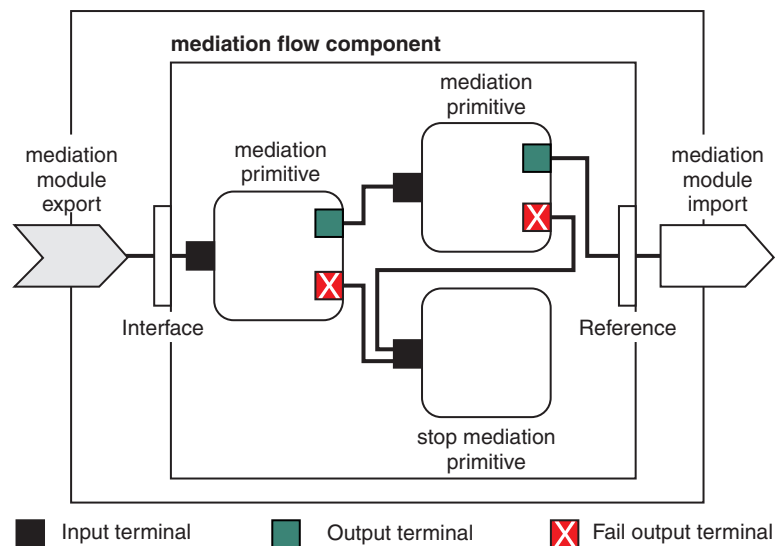
Figure 26. Simplified example of a mediation module. The mediation module contains one mediation flow component, which contains mediation primitives.

- Properties

  Mediation primitives have properties, some of which can be displayed in the administrative console as additional properties of an SCA module.

  For mediation primitive properties to be visible from the WebSphere Process Server administrative console, the integration developer must flag the properties as promoted. Certain properties lend themselves to being administratively

configured and WebSphere Integration Developer describes these as promotable properties, because they can be promoted from the integration cycle to the administrative cycle. Other properties are not suitable for administrative configuration, because modifying them can affect the mediation flow in such a way that the mediation module needs to be redeployed. WebSphere Integration Developer lists the properties that you can choose to promote under the promoted properties of a mediation primitive.

You can use the WebSphere Process Server administrative console to change the value of promoted properties without having to redeploy a mediation module, or restart the server or module.

Newly called mediation flows use property changes immediately, unless the changes occur in a deployment manager cell. If changes occur in a deployment manager cell they take effect on each node as that node is synchronized. Mediation flows that are in-flight continue to use previous values.

Note: If you want to change the property names and types of mediation primitives, and not the property values, you should use WebSphere Integration Developer.

Deploying mediation modules

Mediation modules are created using WebSphere Integration Developer, and are generally deployed to WebSphere Process Server inside an enterprise archive (EAR) file.

You can change the value of promoted properties at deployment time.

You can export a mediation module from WebSphere Integration Developer, and cause WebSphere Integration Developer to package the mediation module inside a Java archive (JAR) file, and the JAR file inside an EAR file. You can then deploy the EAR file, by installing a new application from the administrative console.

Mediation modules can be thought of as one entity. However, SCA modules are defined by a number of XML files stored in a JAR file.

Example of EAR file, containing a mediation module
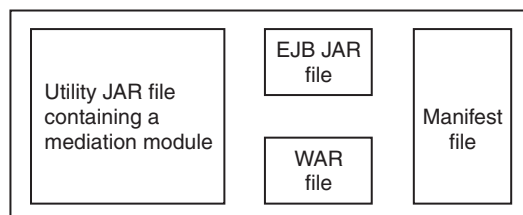
Figure 27. Simplified example of an EAR file containing a mediation module. The EAR file contains JARs. The utility JAR file contains a mediation module.

Related information

Working with mediation modules
You can display the mediation modules deployed to WebSphere Process Server. You can start or stop mediation modules, and view details of a mediation module or its application.

"Displaying mediation module properties" on page 280
You can display the properties of mediation modules that have been deployed to WebSphere Process Server.

"Changing mediation module properties" on page 281
You can change the value of some mediation module properties.

⮕ Promotable properties

Imports and import bindings:

Imports define interactions between Service Component Architecture (SCA) modules and service providers. SCA modules use imports to allow components to access external services (services that are outside the SCA module) using a local representation. Import bindings define the specific way that an external service is accessed.

If SCA modules do not need to access external services they are not required to have imports. Mediation modules usually have one or more imports that are used to pass messages or requests on to their intended targets.

Interfaces and bindings

An SCA module import needs at least one interface, and an SCA module import has a single binding.
- Import interfaces are abstract definitions that define a set of operations using Web Services Description Language (WSDL), an XML language for describing Web services. An SCA module can have many import interfaces.
- Import bindings are concrete definitions that specify the physical mechanism that SCA modules use to access an external service.

Supported import bindings

enterprise service bus supports the following import bindings:
- Web Service bindings allow components to invoke Web services. The supported protocols are SOAP/HTTP and SOAP/JMS. A Web services binding that uses SOAP/JMS, supports JMS using the default WebSphere Application Server default messaging provider in a point-to-point configuration. The SOAP/JMS binding does not support: generic JMS, MQ JMS, or the JMS broadcast mode.
- SCA Bindings connect SCA modules to other SCA modules. SCA bindings are also referred to as default bindings.
- Java Message Service (JMS) 1.1 Bindings allow interoperability with the WebSphere Application Server default messaging provider. JMS can exploit various transport types, including TCP/IP and HTTP or HTTPS. The JMS Message class and its five subtypes (Text, Bytes, Object, Stream, and Map) are automatically supported.
- WebSphere MQ JMS bindings allow interoperability with WebSphere MQ based JMS providers. The JMS Message class and its five subtypes (Text, Bytes, Object, Stream, and Map) are automatically supported. If you want to use WebSphere MQ as a JMS provider you might have WebSphere MQ JMS bindings.
- WebSphere MQ Bindings allow interoperability with WebSphere MQ. You can use WebSphere MQ bindings only with remote queue managers via a WebSphere MQ client connection; you cannot use them with local queue managers. You might have WebSphere MQ bindings if you want to communicate with native WebSphere MQ applications.

- Generic JMS bindings allow interoperability with third-party JMS providers that integrate with the WebSphere Application Server using the JMS Application Server Facility (ASF).
- WebSphere Adapters bindings enable interaction with Enterprise Information Systems (EIS).
- HTTP bindings allow you to access applications using the HTTP protocol.

  Related information

  Working with imports
  You can list the imports of mediation modules that have been deployed to WebSphere Process Server. You can also display import interfaces and change the details of import bindings.

Exports and export bindings:

Exports define interactions between Service Component Architecture (SCA) modules and service requesters. SCA modules use exports to offer services to others. Export bindings define the specific way that an SCA module is accessed by service requesters.

Interfaces and bindings

An SCA module export needs at least one interface.
- Export interfaces are abstract definitions that define a set of operations using Web Services Description Language (WSDL), an XML language for describing Web services. An SCA module can have many export interfaces.
- Export bindings are concrete definitions that specify the physical mechanism that service requesters use to access a service. Usually, an SCA module export has one binding specified. An export with no binding specified is interpreted by the run time as an export with an SCA binding.

Supported export bindings

enterprise service bus supports the following export bindings:
- Web Service bindings allow exports to be invoked as Web services. The supported protocols are SOAP/HTTP and SOAP/JMS. A Web services binding that uses SOAP/JMS, supports JMS using the default WebSphere Application Server default messaging provider in a point-to-point configuration. The SOAP/JMS binding does not support: generic JMS, MQ JMS, or the JMS broadcast mode.
- SCA bindings connect SCA modules to other SCA modules. SCA bindings are also referred to as default bindings.
- Java Message Service (JMS) 1.1 bindings allow interoperability with the WebSphere Application Server default messaging provider. JMS can exploit various transport types, including TCP/IP and HTTP(S). The JMS Message class and its five subtypes (Text, Bytes, Object, Stream, and Map) are automatically supported.
- WebSphere MQ JMS bindings allow interoperability with WebSphere MQ based JMS providers. The JMS Message class and its five subtypes (Text, Bytes, Object, Stream, and Map) are automatically supported. If you want to use WebSphere MQ as a JMS provider you might have WebSphere MQ JMS bindings.
- WebSphere MQ bindings allow interoperability with WebSphere MQ. A remote (or client) connection is the type of connection needed to connect to an MQ queue manager on a remote machine. A local (or bindings) connection is a direct

connection to WebSphere MQ. This can only be used for a connection to a MQ queue manager on the same machine. WebSphere MQ will allow both types of connection, But MQ bindings only support the "remote" (or "client") connection.

- Generic JMS bindings allow interoperability with third-party JMS providers that integrate with the WebSphere Application Server using the JMS Application Server Facility (ASF).
- WebSphere Adapters bindings enable interaction with Enterprise Information Systems (EIS).
- HTTP bindings allow exports to be accessed using the HTTP protocol.

Related information

Working with exports
You can list the exports of mediation modules that have been deployed to WebSphere Process Server. You can also display export interfaces and export bindings.

Service application features of the administrative interfaces

WebSphere Process Server allows you to view and change aspects of service applications using the administrative console.

Service applications provide services, and have an associated Service Component Architecture (SCA) module. The type of SCA modules that are supported by enterprise service bus are mediation modules.

Viewable SCA module details

After you have deployed an EAR (Enterprise ARchive) file containing an SCA module, you can view SCA module details. You can list all your SCA modules, and their associated applications, and you can view details about a particular SCA module.

The SCA module details you can view include some of the following:
- SCA module name.
- Associated application.
- SCA module imports:
  - Interfaces.
  - Bindings.
- SCA module exports:
  - Interfaces.
  - Bindings.
- SCA module properties.

Modifiable SCA module details

After you have deployed an EAR file containing an SCA module you can change the following SCA module details using the administrative console, without having to redeploy the EAR file.
- Import bindings of type SCA:
  - Changing import bindings lets you change service interactions.
  - SCA bindings connect SCA modules to other SCA modules. One SCA module can interact with a second SCA module, and can be changed to interact with another SCA module.
  - Web service bindings connect SCA modules to external services using SOAP.

- Import bindings of type Web service (WS):
  - Changing import bindings lets you change service interactions.
  - WS import bindings allow SCA modules to access web services. A WS import binding calls a service located at a specified endpoint. You can change the end point such that the binding calls the service at an alternative end point, or even an entirely different service with compatible interfaces.
- Export and import bindings of types JMS, WebSphere MQ JMS, generic JMS, WebSphere MQ, and HTTP have attributes that you can modify.
- Mediation module properties:
  - Mediation module properties belong to the mediation primitives with which they are associated. However, the WebSphere Process Server administrative console displays some of them as Additional Properties of an SCA module. The integration developer must flag a mediation primitive property as Promoted in order for it to be visible from WebSphere Process Server.
  - Changing mediation module properties lets you change the behavior of your mediations. The mediation changes that you can make depend upon the properties that have been promoted.

Note: An export with no binding specified is interpreted by the runtime as an export with an SCA binding.
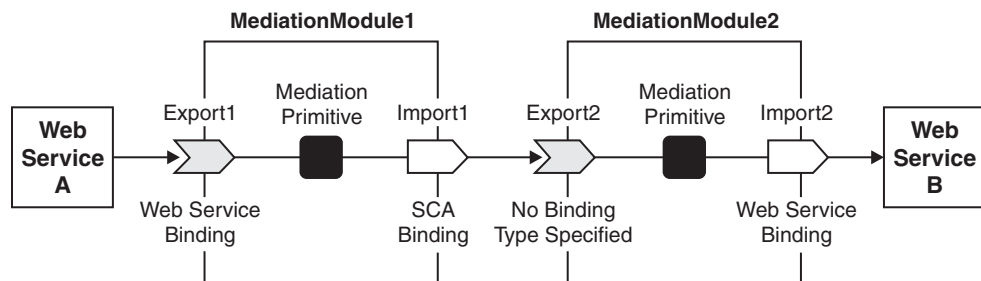
Figure 28. Example showing one mediation module interacting with another mediation module. Mediation Module1 connects to Mediation Module2

Installing a module on a production server

This topic describes the steps involved in taking an application from a test server and deploying it into a production environment.

Before you begin

Before deploying a service application to a production server, assemble and test the application on a test server. After testing, export the relevant files as described in *Preparing to deploy to a server* in the Developing and Deploying Modules PDF and bring the files to the production system to deploy. See the information centers for WebSphere Integration Developer and WebSphere Application Server Network Deployment for more information.

Procedure

1. Copy the module and other files onto the production server.

   The modules and resources (EAR, JAR, RAR, and WAR files) needed by the application are moved to your production environment.
2. Run the serviceDeploy command to create an installable EAR file.

This step defines the module to the server in preparation for installing the application into production.

   a. Locate the JAR file that contains the module to deploy.

   b. Issue the command using the JAR file from the previous step as input.

3. Install the EAR file from step 2 on page 274. How you install the applications depends on whether you are installing the application on a stand alone server or a server in a cell.

   Note: You can either use the administrative console or a script to install the application. See the WebSphere Application Server information center for additional information.

4. Save the configuration. The module is now installed as an application.

5. Start the application.

Results

The application is now active and work should flow through the module.

What to do next

Monitor the application to make sure the server is processing requests correctly.

Related concepts

WebSphere JMS bindings

WebSphere MQ bindings

Related tasks

Administering the Common Event Infrastructure

These topics describe several administrative tasks you can perform to control the operation of the Common Event Infrastructure components at run time.

Related information

serviceDeploy command

Use the serviceDeploy command to package Service Component Architecture (SCA) compliant modules as Java applications that can be installed on a server. The command is useful when performing batch installs through wsadmin.

Promotable properties

Managing resources for mediation modules

Mediation modules uses resources provided by the service integration technologies of WebSphere Application Server. Mediation modules can also make use of a range of resources, including those provided by the Java Message Service (JMS) and common event infrastructure. To administer the resources for mediation modules, you can use the WebSphere administrative console, commands, and scripting tools.

"Administering enterprise applications" on page 312
Use the console's Enterprise Application page (viewed by clicking **Applications > Enterprise Applications**) to view and administer enterprise applications installed on the server.

Installing a mediation module EAR file with the console

Before you can start running a mediation module, you must deploy it to a server or cluster. Deployment involves creating an installable EAR file and installing the EAR file onto the server or cluster.

Before you begin

If you have exported your mediation module to a JAR file, use the serviceDeploy command to create an installable EAR file from the JAR file. For more information, see "Installing a module on a production server" on page 274.

About this task

You must install the EAR file onto a server or cluster before you can start running the mediation module.

Instead of using the administrative console, you can use other methods to install the EAR file, such as the AdminApp.install or AdminApp.installinteractive command with the wsadmin tool.

Important: If, after you start performing the steps, you decide not to install the application you must click **Cancel**: do not simply move to another administrative console page.

Procedure

1. Use the administrative console to install the EAR file: click **Applications** → **Install New Application** in the console navigation pane. The first of two **Preparing for application installation** pages is displayed.
2. On the first **Preparing for application installation** page:
   a. Specify the full path name of the EAR file. For more information, see Installing applications with the console.
   b. Select whether to use default values or specify some of the values yourself:
      Prompt me only when additional information is required
         Displays only the module mapping step and other steps where you must specify information.
      Show me all installation options and parameters
         Displays all installation steps. To use **Generate default bindings**, which supplies default values for incomplete bindings, select this option.
   c. Click **Next**.
3. Installing a mediation module EAR file is similar to installing any other enterprise application EAR file into WebSphere Application Server. For detailed information about filling in the second **Preparing for application installation** page and specifying the options in the remaining wizard steps, see Installing applications with the console.
4. For a mediation module EAR file, there is one additional optional step you can perform: on the **Edit module properties** panel, you can edit the values of the properties in the module.

Results

You can now start the mediation module.

Deploying applications using Apache Ant tasks

This topic describes how to use Apache™ Ant tasks to automate the deployment of applications to WebSphere Process Server. By using Apache Ant tasks, you can define the deployment of multiple applications and have them run unattended on a server.

Before you begin

This task assumes the following:
- The applications being deployed have already been developed and tested.
- The applications are to be installed on the same server or servers.
- You have some knowledge of Apache Ant tasks.
- You understand the deployment process.

Information about developing and testing applications is located in the WebSphere Integration Developer information center.

The reference portion of the information center for WebSphere Application Server Network Deployment contains a section on application programming interfaces. Apache Ant tasks are described in the package com.ibm.websphere.ant.tasks. For the purpose of this topic, the tasks of interest are ServiceDeploy and InstallApplication.

About this task

If you need to install multiple applications concurrently, develop an Apache Ant task before deployment. The Apache Ant task can then deploy and install the applications on the servers without your involvement in the process.

Procedure

1. Identify the applications to deploy.
2. Create a JAR file for each application.
3. Copy the JAR files to the target servers.
4. Create an Apache Ant task to run the ServiceDeploy command to create the EAR file for each server.
5. Create an Apache Ant task to run the InstallApplication command for each EAR file from step 4 on the applicable servers.
6. Run the ServiceDeploy Apache Ant task to create the EAR file for the applications.
7. Run the InstallApplication Apache Ant task to install the EAR files from step 6.

Results

The applications are correctly deployed on the target servers.

Example of deploying an application unattended

This example shows an Apache Ant task contained in a file myBuildScript.xml.

```
<?xml version="1.0">

<project name="OwnTaskExample" default="main" basedir=".">
 <taskdef name="servicedeploy"
   classname="com.ibm.websphere.ant.tasks.ServiceDeployTask" />
 <target name="main" depends="main2">
  <servicedeploy scaModule="c:/synctest/SyncTargetJAR"
```

```
    ignoreErrors="true"
    outputApplication="c:/synctest/SyncTargetEAREAR"
    workingDirectory="c:/synctest"
    noJ2eeDeploy="true"
    cleanStagingModules="true"/>
 </target>
</project>
```

This statement shows how to invoke the Apache Ant task.

```
${WAS}/bin/ws_ant -f myBuildScript.xml
```

Tip: Multiple applications can be deployed unattended by adding additional project statements into the file.

What to do next

Use the administrative console to verify that the newly installed applications are started and processing the workflow correctly.

Administering mediation modules

You can list the mediation modules that have been deployed to WebSphere Process Server, view information associated with individual mediation modules and make changes to some import bindings.

About this task

After deploying service applications, you can view and manage the associated Service Component Architecture (SCA) modules. Mediation modules are types of SCA modules.

Working with mediation modules

You can display the mediation modules deployed to WebSphere Process Server. You can start or stop mediation modules, and view details of a mediation module or its application.

About this task

To display the deployed mediation modules use the administrative console.

Note: Mediation modules are types of Service Component Architecture (SCA) modules.

To list the SCA modules, expand **Applications** ▸ **SCA Modules** in the navigation pane,

Results

The content pane displays the mediation modules that have been deployed to WebSphere Process Server. You can also see the applications that the mediation modules are associated with, and whether the applications are running.

> **Related information**
> Mediation modules
> Mediation modules are Service Component Architecture (SCA) modules that can change the format, content, or target of service requests.
>
> ⏏ Promotable properties

Displaying details of a mediation module:

You can display information on mediation modules that have been deployed to WebSphere Process Server.

About this task

To display details about the deployed mediation modules use the administrative console to complete the following steps.

Procedure

1. In the navigation pane, expand **Applications** → **SCA Modules**, to display the SCA modules.
2. In the content pane, click the SCA module to choose an SCA module.

Results

The content pane displays the SCA module name and description; the name of the associated enterprise application; expandable lists of imports and exports; and a module properties link.

Displaying details of the application for a mediation module:

You can display details about the application used to deploy a mediation module to WebSphere Process Server.

About this task

The application used to deploy a mediation module defines a range of configuration properties that affect the use of the mediation module and associated components. When you installed the application, you specified most, if not all, of its property values.

After installing an application, you might want to review the properties and, if needed, change some of the values.

To display details about the application used to deploy a mediation module, use the administrative console to complete the following steps.

Procedure

1. Expand **Applications** → **SCA Modules** in the navigation pane to display the SCA modules.
2. In the column labelled **Application**, click the application name, to choose an SCA module.

Results

The content pane displays the application details page that provides the application's configuration properties and, if appropriate, local topology.

From this page, you can review and, if needed, change the configuration properties for the application and link to additional console pages, as described in Configuring an application.

Starting and stopping mediation modules:

You can start a mediation module that has a status of Stopped or stop one that is running a status of Started. To change the status of a mediation module, start or stop the application used to deploy the module.

Before you begin

Before you can start or stop the application for a mediation module, you must have deployed the mediation module into WebSphere Process Server, as described in Installing a module on a production server.

About this task

To use the services of a mediation module and associated components, start the associated application. By default, the application starts automatically when the server starts.

You can manually start and stop applications using the following:
* Administrative console
* wsadmin startApplication and stopApplication commands
* Java programs that use ApplicationManager or AppManagement MBeans

To start or stop a mediation module, using the administrative console you complete the following steps.

Procedure

1. Expand **Applications** → **SCA Modules**, in the navigation pane, to list the SCA modules.
2. Select the check box for the SCA module that you want to start or stop.
3. Click the Start or Stop button

Option	Description
Start	Runs the application and changes the state of the application to *Started*. The status is changed to *partially started* if not all servers on which the application is deployed are running.
Stop	Stops the processing of the application and changes the state of the application to *Stopped*.

4. Click **Stop** or select the application you want to restart, then click **Start** to restart a running application.

Results

The status of the application changes and a message stating that the application started or stopped displays at the top the page.

You can change whether or not an application starts automatically when the server on which it resides starts. For more information about starting and stopping WebSphere applications, see Starting and stopping applications.

Displaying mediation module properties:

You can display the properties of mediation modules that have been deployed to WebSphere Process Server.

About this task

You might want to check that property values are what you expect before running a service application.

To display the properties of deployed mediation modules, use the administrative console to complete the following steps.

Procedure
1. In the navigation pane, expand **Applications** → **SCA Modules** to display the SCA modules
2. Click the required SCA module, in the content pane, to choose a SCA module.
3. Click **Module Properties**, under Additional Properties, in the content pane, to list the SCA module properties.

Results

The content pane displays the updatable properties for the SCA module in a table that shows property names, types and values. Only property values are updatable from the administrative console: to change property names and types you use WebSphere Integration Developer. A message is displayed if there are no properties that you can update.

Related information

Mediation modules
Mediation modules are Service Component Architecture (SCA) modules that can change the format, content, or target of service requests.

➦ Promotable properties

Changing mediation module properties:

You can change the value of some mediation module properties.

About this task

You might want to change property values if the runtime environment changes.

To change the values of mediation module properties, use the administrative console to complete the following steps.

Procedure
1. Expand **Applications** → **SCA Modules** in the navigation pane to list the SCA modules.
2. Click a SCA module in the content pane to choose a SCA module.
3. Under Additional Properties, select **Module Properties**, in the content pane, to display the SCA module properties. This displays the module properties that you can update. Property names, types and values are displayed. You can only update property values from the administrative console, to change property names and types use WebSphere Integration Developer.
4. Click a property value from the Properties table to choose a property value.
5. Enter a value that conforms to the property type to change a property value.
6. Click **OK** to save your changes. Then save your changes to the master configuration.

Results

Changes a property value. Newly called mediation flows use property changes immediately, unless the changes occur in a deployment manager cell. If changes occur in a deployment manager cell they take affect on each node in the cell after that node has been synchronized. Mediation flows that are in-flight at the time of the property value change continue to use previous values.

Note: If you want to change property names and types, rather than property values, use WebSphere Integration Developer.

> **Related information**
>
> Mediation modules
> Mediation modules are Service Component Architecture (SCA) modules that can change the format, content, or target of service requests.
>
> ⏏ Promotable properties

Working with imports

You can list the imports of mediation modules that have been deployed to WebSphere Process Server. You can also display import interfaces and change the details of import bindings.

About this task

To list the imports of mediation modules that you have deployed, use the administrative console to complete the following steps.

Procedure

1. Expand **Applications** ↣ **SCA Modules**, in the navigation pane to list the SCA modules.
2. Select the SCA module that you are interested in.
3. Expand **Imports**, in the content pane, under Module components to list the SCA module imports.

Results

The content pane displays a list of imports. If there are no imports an information message is displayed.

> **Related concepts**
>
> "HTTP bindings" on page 192
> The HTTP binding is designed to provide Service Component Architecture (SCA) connectivity to HTTP. This allows existing or newly-developed HTTP applications to participate in Service Oriented Architecture (SOA) environments.
>
> **Related information**
>
> Imports and import bindings
> Imports define interactions between Service Component Architecture (SCA) modules and service providers. SCA modules use imports to allow components to access external services (services that are outside the SCA module) using a local representation. Import bindings define the specific way that an external service is accessed.

Displaying an import interface:

You can display the import interfaces of mediation modules that have been deployed to WebSphere Process Server.

About this task

To display the import interfaces of mediation modules that you have deployed, use the administrative console to complete the following steps.

Procedure
1. Expand **Applications** → **SCA Modules**, in the navigation pane to list the SCA modules.
2. Select a SCA module.
3. Expand **Imports**, under Module components, in the content pane, to list the SCA module imports.
4. Expand the import you require, under Module components, in the content pane, to display import details.
5. Expand **Interfaces** to display import interfaces.
6. Select an interface.

Results

The content pane displays the WSDL (Web Services Description Language) interface.

Displaying an import binding:

You can display Web service, SCA, JMS or Adapter types of import bindings, after you deploy mediation modules to WebSphere Process Server.

About this task

To display the import bindings of mediation modules, use the administrative console to complete the following steps.

Procedure
1. Expand **Applications** → **SCA Modules**, in the navigation plane, to list the SCA modules.
2. Select an SCA module.
3. Expand **Imports**, under Module components, in the content pane, to list the SCA module imports.
4. Expand the import you require, under Module components, in the content pane, to display import details.
5. Expand **Binding** to display import bindings.
6. Select a binding.

Results

The content pane displays the import binding details are displayed.

If you select a different target module the list of target exports and export interfaces changes.

    Related concepts

    Bindings

    Bindings configure SCA imports and exports to allow interaction via a variety of protocols and transports. SCA applications connect to external clients and

services via the transports and protocols of the chosen bindings, which specify the message format and protocol details for a particular interface.

Changing an SCA import binding:

Using the administration console, you can change the details of SCA import bindings. For information about changing the details of other types of binding, see the section on Bindings.

About this task

To change SCA import bindings of mediation modules, use the administrative console to complete the following steps.

Procedure

1. Expand **Applications** › **SCA Modules**, in the navigation pane, to list the SCA modules.
2. Choose an SCA module.
3. Expand **Imports**, in the content pane, under Module components, to list the SCA module imports.
4. Expand the import you require, in the content pane, under Module components, to display the import details.
5. Expand **Binding** to display import bindings.
6. Choose an SCA import binding. SCA import bindings are indicated using the identifier [SCA].
7. Choose a new target SCA module. Select a module from the **Target** drop-down menu. Selecting a different SCA module changes the exports and export interfaces that are displayed.
8. Select an export from the **Export** drop-down menu.
9. Save your changes to the master configuration.

Results

The SCA import binding is changed for the selected SCA module import.

WebSphere Process Server issues a warning for each import interface that is not satisfied by an export interface. WebSphere Process Server compares the WSDL (Web Services Description Language) port type names of import and export. If the port type names are not the same a warning is issued, but you are allowed to ignore the warning. However, if the port type names match WebSphere Process Server assumes that the operations provided are equivalent and no warning is issued.

    Related concepts

    ↪ Bindings

    Bindings configure SCA imports and exports to allow interaction via a variety of protocols and transports. SCA applications connect to external clients and services via the transports and protocols of the chosen bindings, which specify the message format and protocol details for a particular interface.

Changing a Web service import binding:

Using the administrative console, you can change the endpoint URL of a Web service import binding. For information about changing the details of other types of binding, see the section on Bindings.

About this task

To change web service import bindings of mediation modules, use the administrative console to complete the following steps.

Procedure
1. Expand **Applications** → **SCA Modules**, in the navigation pane, to list the SCA modules.
2. Choose an SCA module.
3. Expand **Imports**, in the content pane, under, Module components, to list the SCA module imports.
4. Expand the import you require, in the content pane, under Module components, to display import details.
5. Expand **Binding** to display import bindings.
6. Choose a web service import binding. Web service import bindings are indicated using the identifier [Web service].
7. Change the endpoint URL. Ensure the endpoint is a well-formed URL.
8. Save your changes to the master configuration.

Results

The web service import binding is changed for the selected SCA module import.

The changes take effect after you update the master configuration and restart the SCA module.

> **Related concepts**
>
> Bindings
>
> Bindings configure SCA imports and exports to allow interaction via a variety of protocols and transports. SCA applications connect to external clients and services via the transports and protocols of the chosen bindings, which specify the message format and protocol details for a particular interface.

Working with exports
You can list the exports of mediation modules that have been deployed to WebSphere Process Server. You can also display export interfaces and export bindings.

About this task

To list the exports of mediation modules that you have deployed, use the administrative console to complete the following steps.

Procedure
1. Expand **Applications** → **SCA Modules**, in the navigation pane, to list the SCA modules.
2. Select the particular SCA module that you are interested in.
3. Expand **Exports**, in the content pane, under, Module components, to list the SCA module exports.

Results

The content pane displays a list of exports is displayed. If there are no exports then an information message is displayed.

Related concepts

"HTTP bindings" on page 192
The HTTP binding is designed to provide Service Component Architecture (SCA) connectivity to HTTP. This allows existing or newly-developed HTTP applications to participate in Service Oriented Architecture (SOA) environments.

Related information

Exports and export bindings
Exports define interactions between Service Component Architecture (SCA) modules and service requesters. SCA modules use exports to offer services to others. Export bindings define the specific way that an SCA module is accessed by service requesters.

Displaying an export interface:

You can display the export interfaces of mediation modules that have been deployed to WebSphere Process Server.

About this task

To display the export interfaces of mediation modules that you have deployed, use the administrative console to complete the following steps.

Procedure

1. Expand **Applications** → **SCA Modules**, in the navigation pane, to list the SCA modules.
2. Select an SCA module.
3. Expand **Exports**, in the content pane, under Module components, to list the SCA module, exports.
4. Expand the export you require, in the content pane, under Module components, to display export details.
5. Expand **Interfaces** to display export interfaces.
6. Select an interface.

Results

The content pane displays the WSDL (Web Services Description Language) interface.

Displaying an export binding:

You can display some types of export bindings, after you deploy mediation modules to WebSphere Process Server.

About this task

To display the export bindings of mediation modules, use the administrative console to complete the following steps.

Procedure

1. Expand **Applications** → **SCA Modules**, iIn the navigation pane, to list the SCA modules.
2. Select an SCA module.
3. Expand **Exports**, in the content pane, under Module components, to list the SCA module exports.
4. expand the export you require, in the content pane, under Module components, to display export details.
5. Expand **Binding** to display export bindings.
6. Select a binding.

Administering access to WSRR

You can create, configure, and view all your WebSphere Service Registry and Repository (WSRR) access definitions using the administrative console.

Before you begin

Install and configure WSRR as described in the WSRR Information Center at http://publib.boulder.ibm.com/infocenter/sr/v6r1/topic/com.ibm.sr.doc/cwsr_overview_overview.html

About this task

WebSphere Process Server supports the use of the WSRR product, which allows the storage and retrieval of services. WSRR is installed as an enterprise application and provides a Web service interface that allows you to connect Endpoint Lookup mediation primitives to a registry instance (as part of a mediation flow), to look up services to call. A WSRR definition and its connection properties is the mechanism used to connect to a registry instance and look up a Web service to call. To display a list of all your WSRR definitions, use the administrative console to complete the following steps.

Procedure

In the navigation pane, expand **Service integration** → **WSRR definitions**.

Results

The content pane displays a list of all the WSRR definitions. Each definition holds a set of properties that identify a WSRR instance and specify how you can access it.

What to do next

Click on a WSRR definition to display the general properties for that definition. You can access the detailed connection properties from the content pane.

Related information

WebSphere Service Registry and Repository

The WebSphere Service Registry and Repository (WSRR) product allows you to store, access, and manage information about services. You can use this information to select, invoke, and reuse services.

WSRR administration commands

You can use the WebSphere Services Registry and Repository (WSRR) administration commands to configure WSRR definitions, instead of using the administration panels.

Creating a new WSRR definition

You can create a new WSRR definition using the administrative console.

About this task

Each definition holds a set of properties that identify a WSRR instance and specify how you can access it. To create a new WSRR definition, use the administrative console to perform the following steps.

Procedure

1. Expand **Service integration** → **WSRR definitions**, in the navigation pane. The WSRR definitions page is displayed in the content pane, which shows a list of all the WSRR definitions.
2. Click the New button. The WSRR definitions configuration page displays.
3. Complete the following property fields:

   WSRR definition name
   > This is required and must be unique within the cell. It is the administrative name for this WSRR definition.

   Description
   > You have the option to enter a description for the definition.

   Default WSRR definition
   > This indicates whether this definition is the default. If this is the first definition you have created, it will automatically be set as the default. You cannot update the default from this page.

   Timeout of cache
   > The time (in seconds) after which the queried results in the WSRR cache expire and will be refreshed. This can be changed from the default value. If you specify a value of 0 (zero), then queried results are never cached.

   Connection type
   > Your only choice is **Web service**.

4. Click **Apply** to save these properties.
5. Click **Connection properties**, under Additional properties in the content pane to display. The Connection properties configuration page displays.
6. Complete the following property fields:

   Connection type
   > This is set when the registry definition is created, and cannot be changed.

   Registry URL
   > The URL to be used to connect to the WSRR instance. The default value is: http://localhost:9080/WSRRCoreSDO/services/WSRRCoreSDOPort

   Authentication alias
   > The alias to be used to authenticate with the WSRR instance.

   SSL Configuration
   > The Secure Sockets Layer (SSL) configuration to be used to communicate with a secured WSRR instance.

7. Click **Apply** to save these properties.
8. Click on **Save** to apply your changes to the master configuration.
9. Click **Save**.

Results

A new WSRR definition is created.

> **Related tasks**
>
> "Setting a default WSRR definition"
> You can set a WSRR definition to be the default definition from the
> administrative console.

Deleting a WSRR definition

You can delete a WSRR definition from the administrative console.

About this task

To delete a WSRR definition, use the administrative console to perform the
following steps.

Procedure

1. Expand **Service integration** → **WSRR definitions** in the navigation pane. The
   WSRR definitions page displays in the content pane, showing a list of all the
   WSRR definitions.
2. Select the check box next to the WSRR definition you want to delete. Click
   Delete. The definition disappears from the collection panel.
3. Click on **Save** to apply your changes to the master configuration.

Results

The WSRR definition is deleted.

Note: You cannot delete the default WSRR definition unless it is the only one. If
you want to delete the current default definition, you must first set a different
definition to be the default.

> **Related tasks**
>
> "Setting a default WSRR definition"
> You can set a WSRR definition to be the default definition from the
> administrative console.

Setting a default WSRR definition

You can set a WSRR definition to be the default definition from the administrative
console.

About this task

To set a WSRR definition to be the default, use the administrative console to
perform the following steps.

Procedure

1. Expand **Service integration** → **WSRR definitions** in the navigation pane. The
   WSRR definitions page displays in the content pane, showing a list of all the
   WSRR definitions.

2. Select the check box next to the WSRR definition you want to set as the default. Click **Set as default**. A **Yes** displays in the Default column for the selected WSRR definition. If another WSRR definition was previously set to be the default, a **No** displays in the Default column for that WSRR definition.

3. Click on **Save** to apply your changes to the master configuration.

4. Click **Save**.

Results

The selected WSRR definition is set as the default definition.

Note: The Default column is sortable, so that the default definition always appears at the top, or the bottom, of the list.

> **Related tasks**
>
> "Deleting a WSRR definition" on page 289
> You can delete a WSRR definition from the administrative console.
>
> "Creating a new WSRR definition" on page 288
> You can create a new WSRR definition using the administrative console.

Administering the throughput of SCA requests

For each SCA module deployed on WebSphere Process Server, requests being processed are held on queue points and in the data store for messaging engines. You can display the data for SCA requests, and take any appropriate action to manage the throughput of SCA requests.

About this task

When an SCA module is running in enterprise service bus, requests normally flow through the enterprise service bus without needing to be managed. Sometimes, you might want to check the throughput of a request, check the contents of a request, or if some problem has occurred, delete a request. You might also want to take other actions such as to monitor the overall throughput of requests, or change the reliability setting for requests.

Requests are handled as messages by the service integration technologies of the underlying WebSphere Application Server. For this reason, actions to manage requests are managed by using the WebSphere Application Server tasks to act on service integration messages.

This topic provides an overview of the main tasks that you might consider using, and links into the WebSphere Application Server tasks for more detailed information.

- Listing messages on a message point

  SCA requests that are being processed are held on queue points of the SCA.SYSTEM.bus. You can list the SCA requests either through a queue destination for a component of the SCA module, or through the messaging engine that hosts a queue point; for example: **Service integration** → **Buses** → **SCA.SYSTEM.localhostNode01Cell.Bus** → **Destinations** → **StockQuoteService_Export** → **Queue points** → **StockQuoteService_Export@localhostNode01.server1-SCA.SYSTEM.localhostNode01Cell.Bus** → **Runtime** → **Messages**

- Resolving locked messages on a message point

If a problem occurs, an SCA request might remain locked on the queue point where it is being processed. You can display the message **State** property that indicates whether or not the request is locked, and if appropriate take further action to resolve the problem.

- Deleting messages on a message point

  Under exceptional circumstances, you might need to delete one or more messages that exist on a message point for a selected bus destination or messaging engine. You should not normally need to delete messages on a message point. This task is intended as part of a troubleshooting procedure.

- Viewing data in the data store for a messaging engine.

  A messaging engine maintains requests as volatile (nonpersistent) and durable (persistent) data in its data store. You can use the database tools for the data store to view request data in the data store for a messaging engine. For example, if the messaging engine uses the default Cloudscape™ database, you can use the CloudView tool to view request data.

- Changing message reliability for a destination

  Request messages have a quality of service attribute that specifies the reliability of message delivery. You can select a reliability to suit your requirements for assured delivery, and system performance. The administrator can specify the reliability setting on bus destinations, or the reliability can be specified by individual producers (typically under application control through an API call).

Viewing data in a data store

A messaging engine persists both volatile and durable data in its data store, including messages, transaction states, and communication channel states. You can use the database tools to view data in the data store for a messaging engine.

Before you begin

Before you can use the CloudView tool to view data in a Cloudscape data store for a messaging engine, you must have stopped the messaging engine.

About this task

Volatile data is lost when a messaging engine stops, in either a controlled or an uncontrolled manner. Durable data is available after the server restarts.

In some cases, you might want to view the data in a data store; for example, to examine the messages being processed by the messaging engine.

You can use the database tools for the data store to view data in the data store for a messaging engine. For example, if the messaging engine uses the default Cloudscape database, you can use the CloudView tool to view request messages.

Procedure

1. Start the CloudView tool. For example, on Windows complete the following sub-steps:
   a. Open a command window
   b. Change directory to *install_root*/cloudscape/bin/embedded
   c. Type `cview`
2. Open the data store for the messaging engine. Use the CloudView tool to complete the following sub-steps:
   a. Click **File** → **Open**

b. Browse to, then select, the database file.

   For a messaging engine, the database is stored in the directory *install_root*/profiles/*profile_name*/databases/com.ibm.ws.sib and has the name of the messaging engine; for example, for the default standalone profile on Windows, the database file for the messaging engine localhostNode01.server1-SCA.SYSTEM.localhostNode01Cell.Bus (for server1 on the SCA.SYSTEM bus) is:

install_root/profiles/default/databases/com.ibm.ws.sib/localhostNode01.server1-SCA.SYSTEM.localhostNode01Cell.Bus

   c. Click **Open**

3. Use the CloudView controls to view data.

   a. Expand Tables in the navigation pane.

   b. Click a table name.

   c. In the content pane, click the Data tab.

4. Optional: Display more help about using the CloudView tool.

   To view help about using CloudView, click the Help button or the menubar option **Help → Cview help...**.

   Related information

   ⇨ About Cloudscape v10.1.x

Changing message reliability for a bus destination

Messages have a quality of service attribute that specifies the reliability of message delivery. You can select a reliability to suit your requirements for assured delivery, and system performance.

About this task

The administrator can specify the reliability setting on bus destinations, or the reliability can be specified by individual producers (typically under application control through an API call). The administrator can specify whether the default reliability for the destination can be overridden by a producer, and the maximum reliability that attached producers can request.

To browse or change the message reliability setting of a destination, use the administrative console to complete the following steps:

Procedure

1. In the navigation pane, click **Service integration → Buses**.

2. In the content pane, click the name of the bus on which the destination exists.

3. Click **Destinations**

4. Click the destination name. This displays the details page for the destination.

5. Review the reliability properties. The following properties control the message reliability for the destination:

   Default reliability
   > The reliability assigned to a message produced to this destination when an explicit reliability has not been set by the producer.

   Maximum reliability
   > The maximum reliability of messages accepted by this destination.

   These properties can have values from the following list:

   Best effort nonpersistent
   > Messages are discarded when a messaging engine stops or fails.

Messages may also be discarded if a connection used to send them becomes unavailable and as a result of constrained system resources.

Express nonpersistent
Messages are discarded when a messaging engine stops or fails. Messages may also be discarded if a connection used to send them becomes unavailable.

Reliable nonpersistent
Messages are discarded when a messaging engine stops or fails.

Reliable persistent
Messages may be discarded when a messaging engine fails.

Assured persistent
Messages are not discarded.

For more information about using these properties to control message reliability, see Message reliability levels.

6. Review whether producers can override the default reliability setting.

Enable producers to override default reliability
Select this option to enable producers to override the default reliability that is set on the destination.

7. Optional: Change the destination properties to suit your needs.

You can further refine the configuration of a destination by setting other properties to suit your needs, as described in Configuring bus destinations.

8. Click **OK**.

9. Save your changes to the master configuration.

Doing more with service applications

You can use the WebSphere administrative console not only to manage mediation modules themselves, but also the resources used by mediation modules and the applications that contain mediation modules. You also can use commands to do these tasks.

About this task

The more routine tasks for managing mediation modules are described in "Administering mediation modules" on page 278.

This set of topics provides links into the WebSphere Application Server topics for information about tasks involving applications used to deploy mediation modules.

For information about doing more with mediation modules, see the following sub-topics:

Managing resources for mediation modules

Mediation modules uses resources provided by the service integration technologies of WebSphere Application Server. Mediation modules can also make use of a range of resources, including those provided by the Java Message Service (JMS) and common event infrastructure. To administer the resources for mediation modules, you can use the WebSphere administrative console, commands, and scripting tools.

For more information about managing resources for mediation modules, see the related topics.

Service integration technologies

Service integration resources, such as bus destinations, enable a mediation module to use service integration technologies. Queue destinations are used by the SCA runtime exploited by the mediation module as a robust infrastructure to support asynchronous interactions between components and modules. When you install a mediation module into WebSphere Process Server, the destinations used by a module are defined on a member of the SCA.SYSTEM.bus. These bus destinations are used to hold messages that are being processed for components of the mediation module that use asynchronous interactions:

Queue sca/*module_name*
>   This is the destination used to buffer asynchronous requests sent to module *module_name*

Queue sca/*module_name***/export/***export_name*
>   This is the destination used to buffer asynchronous requests routed to module export *export_name*.

Queue sca/*module_name***/exportlink/***export_name*
>   This is the destination used by the export to send asynchronous requests to the module. Requests are routed to the component target linked to the export.

Queue sca/*module_name***/component/***component_name*
>   This is the destination used to buffer asynchronous requests sent to component *component_name*

Queue sca/*module_name***/component/***component_name***/source/***source_name*
>   This is the destination used to buffer asynchronous requests routed to the component source import *source_name*.

Queue sca/*module_name***/component/***component_name***/target/***target_name*
>   This is the destination used to buffer asynchronous requests routed to the component target export *target_name*.

Queue sca/*module_name***/import/***import_name*
>   This is the destination used to buffer asynchronous requests sent to import *import_name*.

Queue sca/*module_name***/importlink/***import_name*
>   This is the destination used by the import to send asynchronous requests out of the module. Requests are routed to the module export linked to the import.

Queue sca/*module_name***/import/sca/dynamic/import/scaimport [for SCA binding]**

Queue sca/*module_name***/import/sca/dynamic/import/wsimport [for Web service binding]**

Queue sca/contextStore/*module_name*
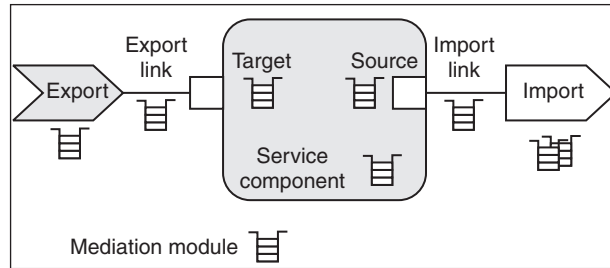
For each of the destinations, a queue point is also created, and defined on the messaging engine of the relevant bus member.

You can deploy and use mediation modules without needing to manage these resources. However, you might want to adjust the configuration of the resources (for example, to modify the maximum messaging quality of service used) or to use them in locating messages for troubleshooting.

Java Message Service (JMS)

JMS resources enable a mediation module to use asynchronous messaging as a method of communication based on the Java Message Service (JMS) programming interface. The JMS support used depends on the JMS binding of the module. For example, a module with a JMS binding uses a JMS connection factory configured on the default messaging provider provided by the underlying WebSphere Application Server, while a module with a WebSphere MQ JMS binding uses a JMS connection factory configured on WebSphere MQ as the JMS provider. To manage use of the Java Message Service, you can administer the following resources:

JMS connection factory
: A JMS connection factory is used to create connections to the associated JMS provider of JMS destinations, for both point-to-point and publish/subscribe messaging. Use connection factory administrative objects to manage JMS connection factories for the provider.

JMS queue
: A JMS queue is used as a destination for point-to-point messaging. Use JMS queue destination administrative objects to manage JMS queues for the provider.

JMS topic
: A JMS topic is used as a destination for publish/subscribe messaging. Use topic destination administrative objects to manage JMS topics for the provider.

JMS activation specification
: A JMS activation specification is associated with one or more message-driven beans and provides the configuration necessary for them to receive messages.

JMS listener port
: A JMS listener port defines the association between a connection factory, a destination, and a message-driven bean. This enables deployed message-driven beans associated with the port to retrieve messages from the destination.

Common Event Infrastructure (CEI)

CEI resources enable a mediation module to use standard formats and mechanisms for managing event data. To manage use of the common event infrastructure, you can administer the following resources:

Data Store Profile
>Defines properties used by the default data store. The default data store is the data store supplied by the Common Event Infrastructure.

Emitter Factory Profile
>This profile defines the options for an event emitter.

Event Bus Transmission Profile
>This profile defines the EJB entry into the event bus.

Event Group Profile
>This profile defines a list of events which are determined through selector expressions. JMS queues and a JMS topic can be associated with each event group. If the event server distribution service is enabled and an event matches an event group the event is distributed to any topic or queues configured for that particular event group.

Event Server Profile
>This profile defines the properties for the event server.

Filter Factory Profile
>This profile defines the properties of a filter. The filter uses the filter configuration string to determine whether an event will be passed to the bus.

JMS Transmission Profile
>The database schema that contains the event tables.

Related concepts

WebSphere JMS bindings

WebSphere MQ bindings

Related tasks

Administering the Common Event Infrastructure

These topics describe several administrative tasks you can perform to control the operation of the Common Event Infrastructure components at run time.

Related information

Installing a module on a production server

This topic describes the steps involved in taking an application from a test server and deploying it into a production environment.

serviceDeploy command

Use the serviceDeploy command to package Service Component Architecture (SCA) compliant modules as Java applications that can be installed on a server. The command is useful when performing batch installs through wsadmin.

Promotable properties

"Administering enterprise applications" on page 312
Use the console's Enterprise Application page (viewed by clicking **Applications > Enterprise Applications**) to view and administer enterprise applications installed on the server.

Using commands to manage service applications

You can manage service applications using commands. The commands can be used within scripts.

Before you begin

Use the wsadmin tool to run service application commands.

About this task

You can use the wsadmin tool in different ways. You can use the tool interactively, as an individual command, or with a script. Running multiple commands in a script is useful if you are administering multiple machines.

WebSphere Process Server includes commands that display SCA modules and their imports and exports, and change the details of import and export bindings

Note: Jacl syntax for wsadmin scripts is deprecated and support will be removed in a future release. Jython is the strategic syntax for wsadmin scripts.

Procedure

1. List the SCA administration commands. `$AdminTask help SCAAdminCommands`
2. Display detailed help about a given command. `$AdminTask help command_name`

`$AdminTask help listSCAModules`

    Related information

      Jacl

      Jython

Administering mediation modules using commands:

You can list the mediation modules that have been deployed to WebSphere Process Server from the command line. You can also view information associated with individual mediation modules and make changes to some import bindings.

Before you begin

Use the wsadmin tool to run WebSphere Process Server commands.

About this task

You can run commands individually or in a script. Running multiple commands in a script is useful if you are administering multiple hosts, or producing regular reports.

Listing mediation modules using commands:

You can use a command to list the mediation modules that have been deployed to WebSphere Process Server.

Before you begin

Use the wsadmin tool to run the following command.

About this task

Use the wsadmin tool to list the deployed mediation modules by completing the following step.

As Mediation modules are types of Service Component Architecture (SCA) modules you list the SCA modules, to display the deployedmediation modules.

Procedure

List the deployed SCA modules. `$AdminTask listSCAModules`

Results

Lists the SCA modules that have been deployed to WebSphere Process Server, and the applications they are associated with. The output is returned in the format: *module name:application name*. This makes it easier for scripts to parse the output and extract names, for use in subsequent commands.

Displaying details of a mediation module using commands:

You can use a command to display attributes of mediation modules.

Before you begin

Use the wsadmin tool to run the following commands.

About this task

Use the wsadmin tool to display the description of a mediation module, by completing the following steps.

To show the description of a particular mediation module, you need to know the mediation module name.

The listSCAModules command lists SCA modules that have been deployed to WebSphere Process Server.

Procedure
1. List the deployed SCA modules. `$AdminTask listSCAModules`
2. Display the details of a particular SCA module. `$AdminTask showSCAModule {`**`-moduleName`** *moduleName*`}`

   Note: In addition to specifying the *moduleName,* you have the option of specifying the *applicationName*. Providing an *applicationName* improves performance.

Results

Displays the name and description of a particular SCA module.

`$AdminTask showSCAModule {`**`-moduleName`** *myModule* **`-applicationName`** *myApplication*`}`

Displaying the properties of a mediation module using commands:

You can use a command to show the properties for a specified mediation module.

Before you begin

Use the wsadmin tool to run the following commands.

About this task

Use the wsadmin tool to display the properties of a mediation module, by completing the following steps.

To show the properties of a particular mediation module, you need to know the mediation module name.

The listSCAModules command lists SCA modules that have been deployed to WebSphere Process Server.

Procedure

1. List the deployed SCA modules. `$AdminTask listSCAModules`
2. Display the properties of a particular SCA module. `$AdminTask showSCAModuleProperties {`**`-moduleName`** *`moduleName`*`}`

   Note: In addition to specifying the *moduleName*, you have the option of specifying the *applicationName*. Providing an *applicationName* improves performance.

Results

Displays the properties of a particular SCA module.

`$AdminTask showSCAModuleProperties {`**`-moduleName`** *`myModule`* **`-applicationName`** *`myApplication`*`}`

Changing a mediation module property using commands:

You can use a command to change a property value for a specified mediation module.

Before you begin

Use the wsadmin tool to run the following commands.

About this task

The listSCAModules command lists all SCA modules that have been deployed to WebSphere Process Server.

Procedure

1. List the deployed SCA modules. `$AdminTask listSCAModules`
2. List the properties for a particular SCA module. `$AdminTask showSCAModuleProperties {`**`-moduleName`** *`moduleName`*`}`

   Note: In addition to specifying the *moduleName*, you have the option of specifying the *applicationName*. Providing an *applicationName* improves performance.

3. Modify a module property for the SCA module. `$AdminTask modifySCAModuleProperty` **-moduleName** *moduleName* **-propertyName** *propertyName* **-newPropertyValue** *newpropertyValue*

   Note: You also have the option of specifying the *applicationName*. Providing an *applicationName* improves performance.

Results

Changes the property value for a particular SCA module property.

`$AdminTask modifySCAModuleProperty {`**-moduleName** *myModule* **-applicationName** *myApplication* **-propertyName** *myPropertyName* **-newPropertyValue** *myNewPropertyValue*`}`

Listing imports using commands:

You can use a command to list the imports of any mediation module that has been deployed to WebSphere Process Server.

Before you begin

Use the wsadmin tool to run the following commands.

About this task

Use the wsadmin tool to list the imports of a mediation module, by completing the following steps.

To list the imports of a mediation module you need to know the name of the mediation module.

The listSCAModules command lists all SCA modules that have been deployed to WebSphere Process Server, and the listSCAImports command lists the imports for a particular SCA module. It is possible for an SCA module not to have any imports.

Procedure
1. List the deployed SCA modules. `$AdminTask listSCAModules`
2. List the imports of a particular SCA module. `$AdminTask listSCAImports {`**-moduleName** *moduleName*`}`

   Note: In addition to specifying the *moduleName*, you have the option of specifying the *applicationName*. Providing an *applicationName* improves performance.

Results

Lists the imports for a particular SCA module.

`$AdminTask listSCAImports {`**-moduleName** *myModule* **-applicationName** *myApplication*`}`

Displaying details of an import using a command:

You can use a command to display import details of a mediation module that has been deployed to WebSphere Process Server.

Before you begin

Use the wsadmin tool to run the following commands.

About this task

Use the wsadmin tool to display the import details of mediation module by completing the following steps.

To show the details of a particular mediation module import, you need to know the mediation module name and the import name.

The listSCAModules command lists all SCA modules that have been deployed to WebSphere Process Server, and the listSCAImports command lists the imports for a particular SCA module. It is possible for an SCA module not to have any imports.

Procedure

1. List the deployed SCA modules. `$AdminTask listSCAModules`
2. List the imports for a particular SCA module. `$AdminTask listSCAImports` `{`**`-moduleName`** `moduleName}`

   Note: In addition to specifying the *moduleName*, you have the option of specifying the *applicationName*. Providing an *applicationName* improves performance.
3. Show the details of a particular SCA module import. `$AdminTask showSCAImport` `{`**`-moduleName`** `moduleName` **`-import`** `importName}`

   In addition to specifying the *moduleName* and *importName*, you have the option of specifying the *applicationName*.

   Note: Providing an *applicationName* improves performance.

Results

Displays the import details for a particular SCA module import.

`$AdminTask showSCAImport {`**`-moduleName`** `myModule` **`-applicationName`** `myApplication` **`-import`** `myImport}`

Displaying an import binding using a command:

You can use a command to display the import bindings of a mediation module that has been deployed to WebSphere Process Server.

Before you begin

Use the wsadmin tool to run the following commands.

About this task

Use the wsadmin tool to display the import bindings of a particular mediation module, by completing the following steps.

To show the import bindings of a particular mediation module import, you need to know the mediation module name and the import name.

The listSCAModules command lists all SCA modules that have been deployed to WebSphere Process Server, and the listSCAImports command lists the imports for a particular SCA module. It is possible for an SCA module not to have any imports.

Procedure

1. List the deployed SCA modules. `$AdminTask listSCAModules`
2. List the imports for a particular SCA module. `$AdminTask listSCAImports {-moduleName moduleName}`

   In addition to specifying the *moduleName*, you have the option of specifying the *applicationName*

   Note: Providing an *applicationName* improves performance.
3. Show the import binding for a particular import. `$AdminTask showSCAImportBinding {-moduleName moduleName -import importName}`

   In addition to specifying the *moduleName* and *importName*, you have the option of specifying the *applicationName*.

   Note: Providing an *applicationName* improves performance.

Results

Displays the import binding for a particular SCA module import.

`$AdminTask showSCAImportBinding {-moduleName myModule -applicationName myApplication -import myImport}`

Changing an SCA import binding using commands:

You can use a command to change the SCA import bindings of mediation modules that have been deployed to WebSphere Process Server.

Before you begin

Use the wsadmin tool to run the following commands.

About this task

An SCA binding connects one Service Component Architecture (SCA) module to another SCA module. Mediation modules are types of SCA module

You might change an SCA import binding if you wanted a particular mediation module to call a different mediation module. If you change an import binding you must ensure that the import and export match, that is, that the operations provided are equivalent. This might involve reviewing the WSDL.

To modify the binding of a particular mediation module import, you need to know the names of the source and target mediation modules, and the specific import and export.

The listSCAModules command lists all SCA modules that have been deployed to WebSphere Process Server. The listSCAImports command lists all imports for a particular SCA module and the listSCAExports command lists all exports for a particular SCA module.

Procedure

1. List the deployed SCA modules. $AdminTask listSCAModules
2. List the imports for a particular SCA module. $AdminTask listSCAImports {**-moduleName** *moduleName*}

   Note: In addition to specifying the *moduleName*, you have the option of specifying the *applicationName*. Providing an *applicationName* improves performance.

3. Modify an SCA module import binding, of type SCA. $AdminTask modifySCAImportSCABinding **-moduleName** *moduleName* **-import** *importName* **-targetModule** *targetModuleName* **-targetExport** *targetExportName*

   You also have the option of specifying the *applicationName* and *targetApplicationName*.

   Note: Providing an *applicationName* and a *targetApplicationName* improves performance.

Results

Changes the SCA import binding for a particular SCA module import.

WebSphere ESB issues a warning for each import interface that is not satisfied by an export interface. WebSphere Process Server compares the WSDL port type names of import and export, if they are not the same then a warning is issued. However, if the port type names do match, then WebSphere Process Server assumes that the operations provided are equivalent and no warning is issued.

$AdminTask modifySCAImportSCABinding {**-moduleName** *myModule* **-applicationName** *myApplication* **-import** *myImport* **-targetModule** *myTargetModule* **-targetApplicationName** *myTargetApplication* **-targetExport** *myTargetExport*}

Changing an import Web service binding using commands:

You can use a command to change the Web service import bindings of mediation modules that have been deployed to WebSphere Process Server.

Before you begin

Use the wsadmin tool to run the following commands.

About this task

You might change an import Web service binding if you wanted a particular mediation module to call a different Web service. If you change an import binding you must ensure that the import and export match, that is, that the operations provided are equivalent. This might involve reviewing the WSDL. Use the wsadmin tool to complete the following steps.

To modify the Web service binding of a particular mediation module import, you need to know the specific URL for the target endpoint.

The listSCAModules command lists all SCA modules that have been deployed to WebSphere Process Server. The listSCAImports command lists all imports for a particular SCA module and the listSCAExports command lists all exports for a particular SCA module.

Procedure

1. List the deployed SCA modules. $AdminTask listSCAModules
2. List the imports for a particular SCA module. $AdminTask listSCAImports {**-moduleName** *moduleName*}

   In addition to specifying the *moduleName*, you have the option of specifying the *applicationName*.

   Note: Providing an *applicationName* improves performance.
3. Modify an import binding, of type WS. $AdminTask modifySCAImportWSBinding **-moduleName** *moduleName* **-import** *importName* **-endpoint** *targetEndpointName*

Results

Changes the import Web service binding for a particular Web service import.

WebSphere ESB issues a warning for each import interface that is not satisfied by an export interface. WebSphere Process Server compares the WSDL port type names of import and export, if they are not the same then a warning is issued. However, if the port type names do match, then WebSphere Process Server assumes that the operations provided are equivalent and no warning is issued.

$AdminTask modifySCAImportWSBinding {**-moduleName** *myModule* **-applicationName** *myApplication* **-import** *myImport* **-endpoint** *http://myTargetEndpoint*}

Listing Exports using commands:

You can use a command to list the exports of any mediation module that has been deployed to WebSphere Process Server.

Before you begin

Use the wsadmin tool to run the following commands.

About this task

Use the wsadmin tool to list the exports of a mediation module by completing the following steps.

To list the exports of a particular mediation module, you need to know the mediation module name.

The listSCAModules command lists all SCA modules that have been deployed to WebSphere Process Server, and the listSCAExports command lists the exports for a particular SCA module. It is possible for an SCA module not to have any exports.

Procedure

1. List the deployed SCA modules. $AdminTask listSCAModules
2. List the exports of a particular SCA module. $AdminTask listSCAExports {**-moduleName** *moduleName*}

   In addition to specifying the *moduleName*, you have the option of specifying the *applicationName*.

   Note: Providing an *applicationName* improves performance.

Results

Lists the exports for a particular SCA module.

`$AdminTask listSCAExports {`**`-moduleName`** `myModule` **`-applicationName`**
`myApplication}`

Displaying details of an export using commands:

You can use a command to display export details of a mediation module that has
been deployed to WebSphere Process Server.

Before you begin

Use the wsadmin tool to run the following commands.

About this task

Use the wsadmin tool to display the export details of mediation module, by
completing the following steps.

To show the details of a particular mediation module export, you need to know
the mediation module name and the export name.

The listSCAModules command lists all SCA modules that have been deployed to
WebSphere Process Server, and the listSCAExports command lists the exports for a
particular SCA module. It is possible for an SCA module not to have any exports.

Procedure
1. List the deployed SCA modules. `$AdminTask listSCAModules`
2. List the exports for a particular SCA module. `$AdminTask listSCAExports`
   `{`**`-moduleName`** `moduleName}`

   Note: In addition to specifying the *moduleName*, you have the option of
   specifying the *applicationName*. Providing an *applicationName* improves
   performance.
3. Show the details of a particular SCA module export. `showSCAExport`
   `-moduleName` `moduleName` **`-export`** `exportName`

   In addition to specifying the *moduleName* and *exportName*, you have the option
   of specifying the *applicationName*.

   Note: Providing an *applicationName* improves performance.

Results

Displays the export details for a particular SCA module export.

`$AdminTask showSCAExport {`**`-moduleName`** `myModule` **`-applicationName`**
`myApplication` **`-export`** `myExport}`

Displaying an export binding using commands:

You can use a command to display the export bindings of a mediation module that
has been deployed to WebSphere Process Server.

Before you begin

Use the wsadmin tool to run the following commands.

About this task

To display the export bindings of a particular mediation module, use the wsadmin tool to complete the following steps.

In order to show the export bindings of a particular mediation module export, you need to know the mediation module name and the export name.

The listSCAModules command lists all SCA modules that have been deployed to WebSphere Process Server, and the listSCAExports command lists the exports for a particular SCA module.

Procedure

1. List the deployed SCA modules. `$AdminTask listSCAModules`
2. List the exports for a particular SCA module. `$AdminTask listSCAExports {`**`-moduleName`** *`moduleName`*`}`

    In addition to specifying the *moduleName*, you have the option of specifying the *applicationName*.

    Note: Providing an *applicationName* improves performance.
3. Show the export binding for a particular export. `$AdminTask showSCAExportBinding {`**`-moduleName`** *`moduleName`* **`-export`** *`exportName`*`}`

    Note: In addition to specifying the *moduleName* and *export*, you have the option of specifying the *applicationName*. Providing an *applicationName* improves performance.

Results

Shows the export binding for a particular SCA module export The information displayed depends upon the type of binding. If an export has no binding specified then the runtime assumes that the binding is of type SCA.

`$AdminTask showSCAExportBinding {`**`-moduleName`** *`myModule`* **`-applicationName`** *`myApplication`* **`-export`** *`myExport`*`}`

Changing WebSphere MQ JMS destinations of deployed SCA modules

As your business environment evolves, it is sometimes necessary to change which WebSphere MQ destinations a Service Component Architecture (SCA) module uses.

Before you begin

You must know which SCA modules use the WebSphere MQ JMS destinations you are changing. By default, these destinations are created when you install an SCA module into a server or cluster. The destinations are of the form:

- For a one-way export:
    - *modulename.exportname*_**MQEXPORT_CF**
    - *modulename.exportname*_**MQ_RECEIVE_D**
- In addition, for a two-way export:

- *modulename.exportname_***MQ_SEND_D**
- For a one-way import:
  - *modulename.importname_***MQIMPORT_CF**
  - *modulename.importname_***MQ_SEND_D**
- In addition, for a two-way import:
  - *modulename.importname_***MQ_RECEIVE_D**

This task assumes that you are using the administrative console to change the configuration.

Restrictions: When changing the destinations keep the following in mind:
- You must configure queue destinations with the target client set to **MQ**.
- You must use a TCP/IP client connection to connect to WebSphere MQ. See WebSphere MQ Intercommunication for details.
- You cannot use channel compression.
- You must set any necessary channel exits for correct data handling: click **WebSphere MQ messaging provider** → **WebSphere MQ connection factory objects** → **Custom Properties** and set the following custom properties:
  - **SENDEXIT** to the value com.ibm.ws.sca.internal.mq.exit.MQInternalSendExitImpl.
  - **RECEXIT** to the value com.ibm.ws.sca.internal.mq.exit.MQInternalReceiveExitImpl.
  - **SENDEXITINIT** to a text string of your choice. Setting this property helps you identify the exit in a trace.
  - **RECEXITINIT** to a text string of your choice. Setting this property helps you identify the exit in a trace.
- You cannot use the same connection factories for WebSphere MQ and Java Message Service (JMS) MQ bindings as the exits will inhibit the JMS MQ bindings.

About this task

After deploying SCA modules, you may need to change the WebSphere MQ JMS destinations to meet new business requirements.

Procedure
1. Stop all applications that use the destinations you are changing. Use the steps described in "Administering enterprise applications."
2. Allow the destination to complete processing the messages in progress.
3. Display the WebSphere MQ JMS destinations. Navigate to this page in the administration console using **Resources > JMS Providers > WebSphere MQ > WebSphere MQ queue destinations**
4. Select the scope of the destinations you are changing.

   If the SCA modules containing the destinations are installed on single servers, select **Server scope**.

   If the SCA modules containing the destinations are installed on clusters, select **Cluster scope**.
5. Select the destination to change from the list.
6. Change the fields on the next page to the new values.
7. Repeat steps 5 and 6 for each destination you are changing.

8. Save the configuration changes.

What to do next

Restart the applications you stopped in step 1 on page 307.

Service Component Architecture modules and WebSphere MQ

SCA modules and WebSphere MQ queues can be connected to provide services to one another.

Service Component Architecture (SCA) modules can communicate with WebSphere MQ applications much in the same way as they do other SCA modules. A module that wants to send requests to a WebSphere MQ application uses an import configured with the correct response and request queues associated with that application. Similarly, an SCA module can provide services to a WebSphere MQ application using an export configured with the appropriate application request and response queues. You define the connections between the SCA modules and the WebSphere MQ queues when you build your module.

From the WebSphere MQ queue manager perspective, the SCA module looks as if it were a normal MQ client. From the SCA module end, the WebSphere MQ queue looks like any other service. You can even further shield the SCA module from the WebSphere MQ queues by using a mediation module between the SCA module and the WebSphere MQ queue and let the mediation transform the original SCA request to the correct format for the target queue and handle the response when it becomes available.

Restrictions: When configuring WebSphere MQ for imports and exports, keep in mind the following:
- You must configure queue destinations with the target client set to **MQ**.
- You must use a TCP/IP client connection to connect to WebSphere MQ. See WebSphere MQ Intercommunication for details.
- You cannot use channel compression.
- You must set any necessary channel exits for correct data handling: click **WebSphere MQ messaging provider** ⟶ **WebSphere MQ connection factory objects** ⟶ **Custom Properties** and set the following custom properties:
    - **SENDEXIT** to the value com.ibm.ws.sca.internal.mq.exit.MQInternalSendExitImpl.
    - **RECEXIT** to the value com.ibm.ws.sca.internal.mq.exit.MQInternalReceiveExitImpl.
    - **SENDEXITINIT** to a text string of your choice. Setting this property helps you identify the exit in a trace.
    - **RECEXITINIT** to a text string of your choice. Setting this property helps you identify the exit in a trace.
- You cannot use the same connection factories for WebSphere MQ and Java Message Service (JMS) MQ bindings as the exits will inhibit the JMS MQ bindings.

Working with targets

Targets provide additional flexibility by allowing you to modify processing by changing the target configured for a reference.

A component can call a component in another module to minimize the time and cost of building an application. Targets provide additional flexibility: to allow your installed applications to benefit from advances in processing or other changes, you can use the administrative console to change the endpoint of a cross-module invocation, without rewriting or redeploying the application.

To take advantage of the flexibility provided, you must understand how the system names the targets. The link from the calling module must connect to the correct target.

Target names

Target names are derived from how the calling component invokes the target. The names have the following format:

Invocation type
> **Name format**

Synchronous
> A name that follows the Java Naming and Directory Interface (JNDI) format, for example:
>
> *folder*/export/*fullpath_to_target*/*target_component_name*

Asynchronous
> A name with the format
>
> *folder*/*calling_component_name*/
> *full_path_to_target_component*/*target_component_name*

Multiple destinations
> This name is the same as an asynchronous invocation but the target actually sends a message to multiple destination components.

> **Related tasks**
>
> "Changing targets"
> Changing the target of a reference provides applications with the flexibility of taking advantage of advances in components as they happen without recompiling and reinstalling the application.

Changing targets

Changing the target of a reference provides applications with the flexibility of taking advantage of advances in components as they happen without recompiling and reinstalling the application.

Before you begin

Before changing the target for a reference you must:
- Make sure the new target uses the same data object type
- Know whether the module is synchronously or asynchronously invoking the target
- Know whether the reference targets a single or multiple services

About this task

Change the target of an import from a module when another service with the same interface as the original target provides new or improved functionality that your module can use.

Procedure

1. Stop the module that contains the reference that you are changing.

   a. Using the administrative console, display the Service Component Architecture (SCA) modules.

      Navigate to this panel using **Applications > SCA Modules**

   b. Select your module and press **Stop**. The display updates to show the application as stopped.

2. Change the target destination of the reference.

   How you make the change depends on how the module invokes the target.

Type of invocation	How to change
Single target service	1. Using the administrative console, display the SCA Modules. Navigate to the panel using **Applications > SCA Modules**. 2. From the displayed list, select the module that contains the import that references the target to change. 3. Expand the list of imports by clicking the plus sign (+) next to **Imports**. 4. Select the import to change from the list. 5. In the **Target** area, select the **Module** from the list. 6. After the **Export** list refreshes, select the export for the new target. 7. Save the change by clicking **OK**.
Multiple target services	1. Display the buses on the system on which the module resides. Navigate to the panel using **Service Integration > Buses**. 2. Select the **SCA.System.*cellname*.Bus** 3. Display the destination targets for the bus by clicking **Destinations**. 4. Select the destination that represents the import that connects the calling module to the targets. This identifier will contain the word import. 5. Display the list of properties by clicking **Context properties**. 6. Select the property to change by clicking on the targets property in the list. 7. Change the **Context value** field to the new destination targets. 8. Return to the **Context properties** panel by clicking **OK**. 9. Save the change by clicking **OK**.

3. Save your changes. Click **Save** when prompted.

What to do next

Start the module and make sure the module receives the expected results.

Deleting J2C activation specifications

The system builds J2C application specifications when installing an application that contains services. There are occasions when you must delete these specifications before reinstalling the application.

Before you begin

If you are deleting the specification because of a failed application installation, make sure the module in the Java Naming and Directory Interface (JNDI) name matches the name of the module that failed to install. The second part of the JNDI name is the name of the module that implemented the destination. For example in sca/SimpleBOCrsmA/ActivationSpec, **SimpleBOCrsmA** is the module name.

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as administrator or configurator to perform this task.

About this task

Delete J2C activation specifications when you inadvertently saved a configuration after installing an application that contains services and do not require the specifications.

Procedure

1. Locate the activation specification to delete.

   The specifications are contained in the resource adapter panel. Navigate to this panel by clicking **Resources > Resource adapters**.

   a. Locate the **Platform Messaging Component SPI Resource Adapter.**

      To locate this adapter, you must be at the **node** scope for a standalone server or at the **server** scope in a deployment environment.

2. Display the J2C activation specifications associated with the Platform Messaging Component SPI Resource Adapter.

   Click on the resource adapter name and the next panel displays the associated specifications.

3. Delete all of the specifications with a **JNDI Name** that matches the module name that you are deleting.

   a. Click the check box next to the appropriate specifications.

   b. Click **Delete.**

Results

The system removes selected specifications from the display.

What to do next

Save the changes.

Deleting SIBus destinations

SIBus destinations are the connections that make services available to applications. There will be times that you will have to remove destinations.

Before you begin

If you are deleting the destination because of a failed application installation, make sure the module in the destination name matches the name of the module that failed to install. The second part of the destination is the name of the module that implemented the destination. For example in sca/SimpleBOCrsmA/component/ test/sca/cros/simple/cust/Customer, **SimpleBOCrsmA** is the module name.

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as administrator or configurator to perform this task.

About this task

Delete SIBus destinations when you inadvertently saved a configuration after installing an application that contains services or you no longer need the destinations.

Note: This task deletes the destination from the SCA system bus only. You must remove the entries from the application bus also before reinstalling an application that contains services (see Deleting J2C activation specifications in the Administering section of this information center.

Procedure
1. Log into the administrative console.
2. Display the destinations on the SCA system bus.

   Navigate to the panel by clicking **Service integration > Buses**
3. Select the SCA system bus destinations.

   In the display, click on **SCA.SYSTEM.*cellname*.Bus**, where *cellname* is the name of the cell that contains the module with the destinations you are deleting.
4. Delete the destinations that contain a module name that matches the module that you are removing.
   a. Click on the check box next to the pertinent destinations.
   b. Click **Delete.**

Results

The panel displays only the remaining destinations.

What to do next

Delete the J2C activation specifications related to the module that created these destinations.

Administering enterprise applications

Use the console's Enterprise Application page (viewed by clicking **Applications > Enterprise Applications**) to view and administer enterprise applications installed on the server.

To view the values specified for an application's configuration, click the application name from the list. The application details page opens and displays the application's configuration properties and, if appropriate, local topology. From this page, you can modify existing values and link to additional console pages for configuring the application.

To administer an enterprise application, select it by clicking the check box next to its name and then use one of the following buttons:

Table 33. Buttons for administering enterprise applications

Button	Resulting action
Start	Attempts to run the application. After the application starts successfully, the state of the application changes to one of the following: • `Started`: The application has started on all deployment targets • `Partial Start`: The application is still starting on one or more of the deployment targets
Stop	Attempts to stop the processing of the application. After the application stops successfully, the state of the application changes to one of the following: • `Stopped`: The application has stopped on all deployment targets • `Partial Stop`: The application is still stopping on one or more of the deployment targets
Install	Opens a wizard to help you deploy an enterprise application or module (such as a .jar, .war, or .ear file) onto a server.
Uninstall	Deletes the application from the WebSphere Application Server configuration repository and deletes the application binaries from the file system of all nodes where the application modules are installed after the configuration is saved.
Update	Opens a wizard to help you update application files deployed on a server. You can update the full application, a single module, a single file, or part of the application. If a new file or module has the same name as a file or module already on the server, the new file or module replaces the existing one. Otherwise, it is added to the deployed application.
Remove File	Deletes a file from the deployed application or module. This button deletes the file from the configuration repository and from the file system of all nodes where the file is installed.
Export	Opens the Export Application EAR files page so you can export an enterprise application to an EAR file. Use the Export action to back up a deployed application and to preserve its binding information.
Export DDL	Opens the Export Application DDL files page so you can export DDL files in the EJB modules of an enterprise application.

For more information on administering applications, see the WebSphere Application Server Information Center.

Related information

➦ Installing a module on a production server

This topic describes the steps involved in taking an application from a test server and deploying it into a production environment.

> serviceDeploy command
>
> Use the serviceDeploy command to package Service Component Architecture (SCA) compliant modules as Java applications that can be installed on a server. The command is useful when performing batch installs through wsadmin.
>
> Promotable properties
>
> Managing resources for mediation modules
>
> Mediation modules uses resources provided by the service integration technologies of WebSphere Application Server. Mediation modules can also make use of a range of resources, including those provided by the Java Message Service (JMS) and common event infrastructure. To administer the resources for mediation modules, you can use the WebSphere administrative console, commands, and scripting tools.

Application Scheduler

Application Scheduler allows an administrator to schedule the starting and stopping of applications that are installed on WebSphere Process Server. Use the Application Scheduler panel in the administrative console to control the scheduling of any installed application. Additionally, you can generate scheduler entries during the migration of a WebSphere InterChange Server repository that includes WebSphere InterChange Server scheduler entries. (See the topics on Migrating from WebSphere InterChange Server and the reposMigrate command). Use the Application Scheduler panel in the administrative console to administer these migrated scheduler entries as well.

In a stand-alone server environment, the Application Scheduler is automatically installed. When you create the stand-alone server profile the Application Scheduler is installed and configured on that server.

In a Network Deployment environment, the Application Scheduler is automatically installed for every managed server and cluster member created - no additional action is needed. See related tasks for instructions on creating new managed servers and cluster members.

In WebSphere InterChange Server an application which contained collaboration objects or connectors, could be started, paused and stopped at the component level, i.e., a component could be stopped while the remainder of the application was allowed to continue. In WebSphere Process Server scheduling of events is provided through the Application Scheduler. The Application Scheduler allows you to start and stop processes at the application level.

Accessing the Application Scheduler

Access the Application Scheduler either programmatically using the Application Scheduler MBean interface, or through the Application Scheduler panels of the administrative console.

For more information on accessing Application Scheduler see:
- "Accessing the Application Scheduler using the Application Scheduler MBean interface" on page 315
- "Displaying scheduler entries using the administrative console" on page 316

Accessing the Application Scheduler using the Application Scheduler MBean interface

Use the command line to invoke the Application Scheduler MBean

About this task

Perform the following to invoke Application Scheduler MBean.

Procedure

1. Set the properties SOAP_HOSTNAME and SOAP_PORT in the class com.ibm.wbiserver.migration.ics.Parameters. This class is in the migration-wbi-ics.jar file in the *WAS_HOME*\lib directory. SOAP_HOSTNAME is the name of the host where Application Scheduler is running. SOAP_PORT is the port where the Application Scheduler is running.

   ```
   Parameters.instance.setProperty(Parameters.SOAP_HOSTNAME, "localhost");
   Parameters.instance.setProperty(Parameters.SOAP_PORT, "8880");
   ```

   Note: If security is turned on, you must specify a userid and password in the soap properties file found at the location in*WAS_HOME*\profiles*profiles*\ properties\soap.client.props.

   This properties file name must be set in the Parameters instance shown here.
   ```
   Parameters.instance.setProperty(Parameters.SOAP_PROPERTIES,
   "WAS_HOME\profiles\profiles\properties\soap.client.props";
   ```

2. Create an instance of the class com.ibm.wbiserver.migration.ics.utils.MBeanUtil that implements calls to the AppScheduler Mbean.

   To instantiate an MBeanUtil, you must pass this query string to its constructor that invokes the correct Mbean based on its `name`, `type`, `server name` and `node name`.

   ```
   protected static final String WEBSPHERE_MB_QUERY_CONSTANT = "WebSphere:*";
   protected static final String NAME_QUERY_CONSTANT = ",name=";
   protected static final String WBI_SCHED_MB_NAME = "Scheduler_AppScheduler";
   protected static final String TYPE_QUERY_CONSTANT = ",type=";
   protected static final String WBI_SCHED_MB_TYPE = "WASScheduler";
   protected static final String SERVER_QUERY_CONSTANT = ",process=";
   serverName = "<server1>";
   protected static final String NODE_QUERY_CONSTANT = ",node=";
   nodeName = "<myNode>";

   String queryString = new StringBuffer(WEBSPHERE_MB_QUERY_CONSTANT)
     .append(NAME_QUERY_CONSTANT)
     .append(WBI_SCHED_MB_NAME)
     .append(TYPE_QUERY_CONSTANT)
     .append(WBI_SCHED_MB_TYPE)
     .append(SERVER_QUERY_CONSTANT)
     .append(serverName)
     .append(NODE_QUERY_CONSTANT)
     .append(nodeName).toString();

   MBeanUtil mbs = new MBeanUtil(queryString.toString());
   ```

3. Call Mbean methods using the invoke() method of the MbeanUtil instance and pass it the name of the method.

Here is an example of invoking the createSchedulerEntry method of the Scheduler Mbean. The first step is to create a `SchedulerEntry` and to set various parameters

like name, type, version, transition, entry status, recurrence type,
recurrence week, recurrence period, initial date, repeat interval and
component id.

```
try
{
 //First we set up the Schedule entry

 ScheduleEntry entry1 = new ScheduleEntry();
 entry1.setCName("BPEWebClient_localhost_server1");
 entry1.setCType("Application");
 entry1.setCVersion("ver1");
 entry1.setCTransition("startApplication");
 entry1.setSchedulerNumberOfRepeats(3); // Fire Three times
 entry1.setScheduleEntryStatus(TaskStatus.SCHEDULED);
 entry1.setRType(Recurrence.MINUTES);
 entry1.setRWeekNumber(-1);
 entry1.setRPeriod(2);
 entry1.setInitialDate(new Date(System.currentTimeMillis()+SIXTY_SECOND_OFFSET));
 entry1.setRepeatInterval(entry1.getInitialDate(), entry1.getRType(),
         entry1.getRWeekNumber(),
 entry1.getRPeriod());
 entry1.setComponentID(entry1.getCName(), entry1.getCType(),
         entry1.getCVersion(),  entry1.getCTransition());
```

Then invoke the Mbean's createSchedulerEntry method. We pass it the scheduler
entry entry1 as a parameter along with the name of the ScheduleEntry class.

```
mbs.invoke(schedulerExtMBName, "createScheduleEntry", new Object[]{entry1},
    new String[]{"com.ibm.wbiserver.scheduler.common.ScheduleEntry"});
```

Finally, read all the Schedule entries including the one that was just added by
calling the readAllScheduleEntries method.

```
    result = mbs.invoke("readAllScheduleEntries", null, null);
}
catch (MigrationException e)
{
    e.printStackTrace();
}
```

Displaying scheduler entries using the administrative console

Use the Application Scheduler panel of the administrative console to create,
modify, or delete scheduler events.

Before you begin

You must be at the administrative console for the server to perform this task.

About this task

To display this panel and view existing scheduler events, follow these steps.
1. Select **Servers > Application Servers >** *ServerName*.
2. Select **Application Scheduler** under the **Business Integration** subheading.
3. Select the **scope** (cell, node, server) of the entries to display. The existing
   scheduled events for that scope are listed; Scheduler Entries are normally
   defined at the server scope.

Results

You can now create a new scheduler event, edit existing scheduler events, or delete existing scheduler events.

Creating a scheduled event

The administrative console provides a panel for creating new scheduled events.

Before you begin

To create a new scheduled event, you must be at the Application Scheduler collection panel in the administrative console for the server. See "Displaying scheduler entries using the administrative console" on page 316 for more information.

About this task

There are times that you will have to create an event to fit a specific need. To create a new scheduled event, follow these steps.

Note: The fields with an "*" on the panel are required fields.

Procedure
1. Click **New**. The Add panel opens.
2. Configure the scheduled event.
   a. Select **Group Application**
   b. Select **Status**
   c. Type in the **Initial Date** with the following format *Abbrv month, dd, yyyy* for example, type in **Apr 15, 2005** for April 15, 2005.
   d. Type in the **Initial Time** using a 12-hour format*hh:mm*

      Note: You must also signify either **am** or **pm** and **time zone**.

      Note: After you have moved from this field, the **Next Fire Time** is automatically calculated.
   e. Select the **Action**.

      Optional: You can also fill in the **Recurrence** parameters.
      - Start-by period.

        If the Application Scheduler or Process Server is not running at the time an event is scheduled to fire, then the start-by period parameter defines a length of time or window (in minutes) commencing at the scheduled firing time of the event, during which an event will fire if the Application Scheduler or process server resumes operation. However, if they do not resume operation until after the Start-by-period has expired then the next fire time is calculated and the event will fire at that time.

        For example, if you set the start-by period to 60 - Minutes for an event which is schedule to fire at midnight but the server happens to be down at that time, provided that the server comes back online before 1:00 am, the event will fire.
      - Whether the schedule entry should recur at a specified time.
        – One or more times a minute, hour, day, month or year.
        – A certain day (Sunday thru Saturday) of a certain week (first, second, third, fourth or last) of every one or more months.
        – The last day of every one or more months.

3. Click **Apply** or **OK** to set the event.

> **Note:** To create another event, click **Reset** to clear the panel.

Results

Application Scheduler creates and displays a new scheduled event in the
Application Scheduler panel.

Event status and action descriptions:

Each event must have a status and an action.

Status

The status field shows what state the event is in for monitoring purposes. This
table lists each status.

Status	Description
Scheduled	A task is to fire at a predetermined date, time and interval. Each subsequent firing time is calculated.
Suspended	A task is suspended and will not fire until its status is changed to scheduled.
Complete	A task is completed.
Cancelled	A task has been cancelled. The task will not fire and it cannot be resumed, but it can be purged.
Invalid	Normally the reason that a task has a status of invalid is because either the task has been purged or the information used to query for that task is invalid.
Running	A task is in the midst of firing. **Note:** This status should be rarely seen since it just monitors the event for the very short duration that the event is firing.

Action

Each event must have an action associated with it. The action signifies what what
to do with the event. There are only two actions available for an event:

- **Start Application** - starts all applications that are under the system's
  deployment manager.
- **Stop Application** - stops all applications that are under the system's deployment
  manager.

Modifying a scheduled event

Modify migrated or existing scheduled events from the administrative console.

Before you begin

To modify a scheduled event, you must be at the Application Scheduler collection panel in the administrative console for the server. See "Displaying scheduler entries using the administrative console" on page 316 for more information.

About this task

There are times that you need to modify an event to fit a need. To modify an event, follow these steps.

Procedure
1. Click the **Schedule Entry ID** of the event that you want to modify. The Event panel opens.
2. Modify any of the following fields:

   Note: Since all applications on the server are listed, you must be careful when changing the status of an existing event. You may stop an application that is running on the server.
   - **Group Application**
   - **Status**
   - **Initial Date** with the following format *Abbrv month, dd, yyyy*
   - **Initial Time** using a 12-hour format*hh:mm*
   - **Action**

   Optional: You can also fill in the **Recurrence** parameters.
3. Click **Apply** or **OK** to set the modifications for the event.

   Note: If you modify a scheduled event, the server assigns a new Schedule Entry ID. The server deletes the currently scheduled event and schedules a new event with the new ID.

Results

The panel displays the modified event with the new ID in the Application Scheduler collection panel.

Deleting a scheduled event
Application Scheduler provides a panel for deleting scheduled events.

Before you begin

To delete a scheduled event, you must be at the Application Scheduler collection panel in the administrative console for the server. See "Displaying scheduler entries using the administrative console" on page 316 for more information.

About this task

As events become obsolete, you can delete them from the list of events in the collection panel. Follow these steps to delete a scheduled event.

Procedure
1. In the **Select** column, select the Schedule Entry to be deleted.
2. Click **Delete**.

Results

The Schedule Entry is deleted.

Administering relationships

The relationship manager is a tool for manually controlling and manipulating relationship data to correct errors found in automated relationship management or provide more complete relationship information. In particular, it provides a facility for retrieving as well as modifying relationship instance data.

How the relationship manager works

The relationship manager allows you to configure, query, view, and perform operations on relationship runtime data, including roles and their data. You create relationship definitions with the relationship editor. At run time, instances of the relationships are populated with the data that associates information from different applications. This relationship instance data is created when the maps or other WebSphere Process Server components run and need a relationship instance. The relationship service exposes a set of application programming interfaces (API's) to retrieve relationship metadata and to create, retrieve, and manipulate the instance data. The data is stored in the relationship tables that are specified in the relationship definition. The relationship manager provides a graphical user interface to interact with the relationships and relationship instances.

For each relationship instance, the relationship manager can display a hierarchical listing of its roles. Each role in the relationship has instance data, properties, and key attributes. The relationship tree also provides detailed information about each of the roles in the relationship instance, such as the type of entity, its value, and the date it was last modified. A relationship instance ID is automatically generated when the relationship instance is saved in the relationship table. The relationship manager displays this instance ID at the top level of the relationship tree.

Uses of the relationship manager

You can use the relationship manager to manage entities at all levels: the relationship instance, role instance, and attribute data and property data levels. For example, you can use the relationship manager to:
- Browse and inspect the values for existing relationships
- Create and delete relationship instances
- Modify the contents of a relationship instance, such as adding and deleting role instances
- Edit the data of a relationship role instance like role properties and logical state
- Activate and deactivate role instances
- Get role instances, given the key attribute, start and end date, and property value
- Salvage a situation when problems arise. For example, when corrupt or inconsistent data from a source application has been sent to the generic and destination application relationship table, you can use the relationship manager to rollback the data to a point in time when you know the data is reliable

For more information on relationships, see the WebSphere Integration Developer Information Center and the topics on the relationship services in the WebSphere Process Server Information Center.

Viewing relationships

Perform this task to view a list of relationships in the system, including the relationship name, display name, and static and identity attributes.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as a monitor, an operator, a configurator, or an administrator to perform this task.

About this task

To view the list of relationships in the system, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.

    The information is displayed in table format. Each relationship type is a link.

    Tip: You can customize the number of rows to display at one time. Click **Preferences** and modify the **Maximum row** field value. The default is 25.

Viewing relationship details

Perform this task to view detailed information for the selected relationship, including the relationship name, display name, associated roles with their attributes, property values, and static and identity attributes.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as a monitor, an operator, a configurator, or an administrator to perform this task.

About this task

To view detailed information for the selected relationship, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.
4. You can view the relationship details in two ways:
    a. Click the relationship name.
    b. Select the radio button next to the relationship name and click **Details**.

The relationship details include role attributes, which are displayed in table format and include the display name, object name, and managed attribute setting for the role.

To return to the list of relationships, click **Relationships** from the path at the top of the page or click **Back**.

Viewing role details

Perform this task to view detailed information for the selected role, including the relationship name, role name, display name, property values, keys, role object type, and managed attribute setting.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as a monitor, an operator, a configurator, or an administrator to perform this task.

About this task

To view detailed information for the selected role, perform the following steps.

Procedure
1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the Relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.
4. Click a relationship name to open the Relationship Detail page.
5. Under **Role schema information**, click an associated role name to open the Role Detail page.

What to do next

To return to the Relationship Detail page, click **Relationship Detail** from the path at the top of the page or click **Back**.

Querying relationships

Use this task to perform relationship-based instance queries.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as a monitor, an operator, a configurator, or an administrator to perform this task.

About this task

Select a query option (**All**, **By ID**, **, By property**, or **By role**) to get all or a subset of the instance data for a relationship. The return is the result set of that query and is displayed in table format with each row representing one relationship instance.

To query relationships, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.
4. Select the radio button next to the relationship name and click **Query**.
5. Click one of the query option tabs and specify the search criteria.

Option	Description
All tab	Get a list of all instances in the relationship. You can select to display all activated, all inactivated, or all activated and inactivated relationship instance data.
By ID tab	Get relationship instances in the range of the starting and ending instance identifiers. If you leave one field blank, the query returns only the single instance. The query returns all of the roles for the instances it finds.
By property tab	Get relationship instances by specific property values.
By role tab	Get relationship instances based on a role name, key attribute value, date range during which the role was created or modified, or specific property value.

6. After you have specified the query parameters, you have the following options:
   - Click **OK** to display the result data from the query.
   - Click **Cancel** to discard any changes made and return to the list of relationships.

Viewing relationship instances

Perform this task to view a list of relationship instances that match the relationship query. The results display in table view and include the relationship instance ID and the property values associated with the instance.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as a monitor, an operator, a configurator, or an administrator to perform this task.

About this task

To view a list of relationship instances that match the relationship query, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.

4. Select the radio button next to the relationship name and click **Query**.

5. Click one of the query option tabs (**All**, **By ID**, **By property**, or **By role**) and specify the search criteria. For descriptions of the query options, see "Querying relationships" on page 322

6. Click **OK** to open the Relationship Instances page.

Results

The list of relationship instances that match your query appears in table view, with each relationship instance shown in its own row. The total page and returned instance counts are displayed at the bottom of the page.

Tip: You can customize the number of rows to display at one time. Click **Preferences**, modify the **row** field value, and click **Apply**. The default is 25, with 1 being the minimum number of records to display at one time and all records being the maximum.

You can navigate through the pages, as follows:

- To view the next set of instances, click the forward arrow.
- To view the previous page of instances, click the back arrow.

Restriction: Filtering or sorting on a large relationship instance count might result in performance problems as it requires getting the full query result set from the server in order to do the sorting. For example, sorting the relationship instance data on a query that would return 20,000 relationship instances needs to sort on those 20,000 instances. The total count (bottom of page) gives an estimate of how many relationship instances you can expect and whether sorting or filtering on a large set of data might lead to long wait times.

For information on setting the query block size parameter to allow for customization of how many instances are read from the server at one time, see the help topic on configuring the relationship service.

Viewing relationship instance details

Perform this task to view detailed information for the selected relationship instance, including the relationship name, relationship instance ID, property values, participating roles, and role instance values (role instance ID, logical state, key attributes, and property values). You can view multiple roles concurrently.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as a monitor, an operator, a configurator, or an administrator to perform this task.

About this task

To view detailed information for the selected relationship instance, perform the following steps.

Procedure

1. Ensure that the administrative console is running.

2. In the navigation pane, click **Integration Applications > Relationship Manager**.

3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.

4. Select the radio button next to the relationship name and click **Query**.

5. Click one of the query option tabs (**All**, **By ID**, **By property**, or **By role**); specify the search criteria; and click **OK** to open the Relationship Instances page.

6. You can view the relationship instance details in two ways:
   - Click the relationship instance ID.
   - Select the radio button next to the relationship instance ID and click **Details**.

   To return to the list of relationship instances, click **Relationships Instances** from the path at the top of the page.

Editing relationship instance details

Perform this task to edit the property values for the selected relationship instance.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an operator or an administrator to perform this task.

About this task

To edit the property values for the selected relationship instance, perform the following steps.

Procedure

1. Ensure that the administrative console is running.

2. In the navigation pane, click **Integration Applications > Relationship Manager**.

3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.

4. Select the radio button next to the relationship name and click **Query**.

5. Click one of the query option tabs (**All**, **By ID**, **By property**, or **By role**); specify the search criteria; and click **OK** to open the Relationship Instances page.

6. Display the relationship instance details in one of two ways:
   - Click the relationship instance ID.
   - Select the radio button next to the relationship instance ID and click **Details**.

7. Modify the relationship instance property values, as necessary.

   Restriction: You can only edit the property values if they have been previously defined for the relationship instance.
   To delete the relationship instance, click **Delete** at the bottom of the page.

   From this page, you can also create new role instances or delete existing role instances by selecting them and clicking **Create** or **Delete**, respectively, below the role table. Clicking **Create** will open the New Role Instance page for entering key attribute values and property values for the new role instance. You can edit the property values of the role instance by clicking the selected role instance ID.

8. When you are finished making changes in the instance and within the roles of the instance, you have the following options:
   - Click **OK** to save the changes to the system immediately.
   - Click **Cancel** to discard any changes and return to the Relationship Instances page.

Creating new relationship instances

Perform this task to create a new relationship instance.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an operator or an administrator to perform this task.

About this task

To create a new relationship instance, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.
4. Select the radio button next to the relationship name and click **Create** to open the New Relationship Instance page.
5. Add the property value information in the **Value** field if you want values other than the default values, and click **OK** to save the new relationship instance locally.

   Note: You must also create a role instance for the relationship instance, as you cannot have a relationship instance without a role instance.

Deleting relationship instances

Perform this task to delete a selected relationship instance.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an operator or an administrator to perform this task.

About this task

To delete a selected relationship instance, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.

3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.
4. Select the radio button next to the relationship name and click **Query**.
5. Click one of the query option tabs (**All**, **By ID**, **By property**, or **By role**); specify the search criteria; and click **OK** to open the Relationship Instances page.
6. Select the radio button next to the ID of the relationship instance you want to delete.
7. Click **Delete**.

   The relationship instance is deleted immediately from the system.

Rolling back relationship instance data

Perform this task to roll back the relationship instance data to a specified date and time.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an operator or an administrator to perform this task.

About this task

The following actions are performed during the rollback:
- Relationship instances which are created during the given period get deleted (hard delete) from the database.
- Relationship instances which are activated get deleted (hard delete) from the database.
- Relationship instances which are deactivated in the given time period get activated.

To roll back the relationship instance data, perform the following steps.

Procedure
1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the relationships page for the server you want to manage by clicking **Relationships** next to the relationship services MBean.
4. Select the radio button next to the relationship name and click **Rollback**.
5. Enter the time period for the rollback in the **From date** and **To date** fields.

   Important: Make sure the WebSphere Process Server server and the database server are set to the same time zone or the rollback will fail.
6. Click **OK**.

   All instance data in the relationship created later than the specified date and time will be marked as deactivated.

Viewing role instance details

Perform this task to view detailed information for the selected role instance, including the role name, role element, key attributes and property values, status, and logical state.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as a monitor, an operator, a configurator, or an administrator to perform this task.

About this task

To view detailed information for the selected role instance, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.
4. Select the radio button next to the relationship name and click **Query**.
5. Click one of the query option tabs (**All**, **By ID**, **By property**, or **By role**); specify the search criteria; and click **OK** to open the Relationship Instances page.
6. Display the relationship instance details in one of two ways:
   - Click the relationship instance ID.
   - Select the radio button next to the relationship instance ID and click **Details**.
7. To view the details for the role instance, click its associated ID in the role instance table.

Editing role instance properties

Perform this task to edit the property values for the selected role instance.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an operator or an administrator to perform this task.

About this task

To edit the property values for the selected role instance, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.

4. Select the radio button next to the relationship name and click **Query**.

5. Click one of the query option tabs (**All**, **By ID**, **By property**, or **By role**); specify the search criteria; and click **OK** to open the Relationship Instances page.

6. Display the relationship instance details in one of two ways:
   - Click the relationship instance ID.
   - Select the radio button next to the relationship instance ID and click **Details**.

7. In the role instance table, click the role instance ID to display the role instance details.

8. Edit the role instance property information, as necessary, and click **OK** to save these changes locally.

   Restriction: You can only edit the property values if they have been previously defined for the relationship instance.

Creating new role instances

Perform this task to create a new role instance for a relationship.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an operator or an administrator to perform this task.

About this task

To create a new role instance for a relationship, perform the following steps.

Procedure

1. Ensure that the administrative console is running.

2. In the navigation pane, click **Integration Applications > Relationship Manager**.

3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.

4. Select the radio button next to the relationship name and click **Query**.

5. Click one of the query option tabs (**All**, **By ID**, **By property**, or **By role**); specify the search criteria; and click **OK** to open the Relationship Instances page.

6. Display the relationship instance details in one of two ways:
   - Click the relationship instance ID.
   - Select the radio button next to the relationship instance ID and click **Details**.

7. Locate the role for which you want to create a new instance and click **Create** below the role table to open the New Role Instance page.

8. Enter the key attribute and role property values in their respective **Value** fields and click **OK** to save the new role instance locally.

   Restriction: You can only set the key attribute value when creating the role instance. You cannot change this information after you have applied the changes back to the database. However, you can edit the property values later.

Deleting role instances

Perform this task to delete a selected role instance of a relationship.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an operator or an administrator to perform this task.

About this task

To delete a selected role instance of a relationship, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.
4. Select the radio button next to the relationship name and click **Query**.
5. Click one of the query option tabs (**All**, **By ID**, **By property**, or **By role**); specify the search criteria; and click **OK** to open the Relationship Instances page.
6. Display the relationship instance details in one of two ways:
   - Click the relationship instance ID.
   - Select the radio button next to the relationship instance ID and click **Details**.
7. Locate the role from which you want to delete the role instance.
8. Click the radio button next to the role instance you want to delete and click **Delete** below the role table.

   The role instance is deleted locally.

Removing relationship instance data from the repository

An application that uses relationships has associated relationship schema and data in a repository. The repository is the database configured to hold the relationship instance data. When you uninstall such an application from a production server, the server does not remove the relationship schema and data from the repository. To do so, you need to remove the existing relationship schema manually.

Before you begin

Make sure that you uninstall the application that uses the relationship schema from all servers that access that schema.

About this task

When you install an application containing relationships, the server creates the corresponding database schema objects including tables, indexes, sequences, and stored procedures. At run time, the tables are populated with the relationship instance data. If you uninstall the application that contains relationships, the tables and instance data are not removed from the database. This design can present a problem if you attempt to reinstall the application after modifying the relationship or role definitions.

Note: If you use the Unit Test Environment (UTE) test server in WebSphere Integration Developer (WID), the relationship schema and data are removed from the repository when an application project is removed.

If you reinstall the application with the same relationship, the old schema is reused. However, if the relationship or role definition is modified in such a way that makes it incompatible with the existing schema, the relationship service throws an exception and terminates the installation of the relationship. The logs show the following exception and message:

```
RelationshipServiceException("table <tablename> already exists, but the
table schema is different from current role definition")
```

The solution for this problem is to remove the existing relationship schema artifacts manually, using the tools supplied by the database platform of your repository, and to reinstall the application.

To remove the existing relationship schema from the repository, perform the following steps.

Procedure

1. Locate the database. The database location depends on the database platform.

Option	Description
Database platform	Location
Derby	WASHOME\derby\databases\RepositoryDB
Other databases	The location is configured during installation and profile creation of the server. For example, if you configured the server automatically and selected the default database name, the name of the database is WPRCSDB. For DB2® on i5/OS, the referenced container is a collection instead of a database. It is the collection name rather than the database name that is configured during installation and profile creation; and it is the collection rather than the database that is by default named WPRCSDB.

2. Delete the database artifacts making up a relationship: Using the tools for your database platform, perform the following steps to delete all database objects for a given relationship.

   a. Before removing any data from the database in the following steps, make a backup of the database.

      Note: For DB2 on i5/OS, make a backup of the collection before removing any data.

   b. Find the relationships tables. The following tables are created at the time the relationships are installed.

Table	Format
1 table for relationship properties	_<relname>_P_uniqueidentifier
1 table for generating instance IDs for each relationship (on Derby)	_<relname>_S_uniqueidentifier
1 table for role properties for each application-specific role	_<relname>_<rolename>_P_uniqueidentifier

Table	Format
1 table for each application-specific role (for static relationships 1 table for the generic role is also created)	_<relname>_<rolename>_RT_uniqueidentifier

> **Restriction:** Only the first four characters of the relationship name are used. If the database holds tables for multiple relationships, you should distinguish relationship names within the first 4 characters.

  c. Find the stored procedures. Stored procedure objects have the following format:

     _<relname>_RS_uniqueidentifier or
     <relname><rolename>_RS_uniqueidentifier

  d. Find the sequences. Sequence objects have the following format:

     _<relname>_S_uniqueidentifier

> **Restriction:** Sequences are not supported under Derby.

  e. Using the tools for your database platform, delete the following:
    1) tables
    2) stored procedures
    3) sequences (except for Derby)

Results

The relationship instance data is removed from the database repository.

What to do next

Now you can reinstall the application.

Tutorial: Relationship manager administration

This tutorial demonstrates the basic functions of the WebSphere Process Server relationship manager. Relationships are used to correlate identifiers from different environments for the same item of data. For example, in one environment, states are identified by two-letter abbreviations (AZ, TX). In another environment, different abbreviations are used (Ariz., Tex.). A relationship would be created to correlate "AZ" in the first environment to "Ariz" in the second environment.

The sample relationship referenced here correlates customer IDs. Many business applications maintain databases of customers, and most of these applications assign their own ID to each customer. In an enterprise environment, the same customer likely has a different ID in each business application. In this tutorial, a relationship is defined to correlate customer IDs. The relationship name is "SampleCustID". Two roles are defined for this relationship. One role is for the Customer Information System (CIS), and the other role is for the General Ledger (GL) application. This relationship was created by the relationship services sample along with the roles and a small amount of sample data.

The relationship manager is designed to add, modify, and remove role instances of a relationship instance as well as add, modify, and remove relationship instances. WebSphere Integration Developer should be used to create and deploy new relationship definitions. The definitions are stored as XML files that are deployed as part of a J2EE application to a particular server.

Objectives of this tutorial

After completing this tutorial, you will be able to change the values of relationship instances.

Time required to complete this tutorial

This tutorial requires approximately 10 minutes to complete.

Prerequisites

This tutorial uses a relationship that is created by the relationship services technical sample. Before following the steps of this tutorial, go to the samples gallery and perform the steps described in the relationship services sample to create the required relationship and roles.

> **Related information**
>
> ➡ Accessing the Samples (Sample Gallery)

Changing the values of a relationship instance
About this task

One of your customers has a customer ID of A004 in your CIS application. This same customer has a customer ID of 801 in your GL application. However, due to a data entry error, the relationship instance that correlates the customer IDs of this customer currently has a value of 901 instead of 801 for the GL customer ID. This tutorial takes you through the steps to correct this entry in the relationship.

Procedure

1. Open the WebSphere Process Server administrative console.
2. If security is enabled, log in as a user with administrator privileges.
3. In the navigation pane, click **Integration Applications > Relationship Manager**.
4. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.

   A relationship named SampleCustID should be visible.
5. Select the radio button next to SampleCustID and click **Query**.
6. Click the query **By role** tab; and in the **Role name** field, select MyGLCustomer_0 from the drop-down list, enter 901 in the **Value** field under **Key attributes**, and click **OK**.

   This locates the relationship instance for the requested customer and opens the Relationship Instances page.
7. Click the relationship instance ID.

   This will display the relationship instance data for customer ID 901 in the GL application, including all the associated role instances.
8. In the MyGLCustomer_0 role table, select the role instance ID with the key attribute value 901 and click **Delete** below the role table.

   Note: It should not have any property values associated with it. If any other data appears, you need to look at the role instance and record any data you want to keep.
9. Click **Create** to open the New Role Instance page for creating a new role instance for this relationship instance.

10. Enter 801 in the **Value** field under **Key attributes** and click **OK**.

    The new role instance is saved, and you should see a new role instance in the table.

    Results

    You now have the correct customer ID value in the relationship instance for the GL application.

Administering the relationship service

The relationship service maintains relationships and roles in the system. It manages relationship and role definitions and metadata and makes it possible to specify the definition of a relationship and manipulate the instances derived from the definition.

The relationship service makes it possible to capture relationships across different objects. Participants in the relationship are distinguished by the roles they serve. For instance, a Person object "Joe" can have an ownership relationship with a Car object "Subaru with license plate XYZ 123." In this example, Joe participates in the relationship with the role "owner" while the car participates in the relationship under the role "owned object."

Relationship and role definitions

Relationships and roles are described in definitions that you design through the graphical interface of the relationship editor tool in WebSphere Integration Developer. The relationship definition is a template that describes what the relationship should look like, identifying the roles each participant in the relationship can assume. The role definition captures the structure and constraint requirements for the participants. Relationship definitions are stored as XML files that are deployed as part of a J2EE application to a particular server.

For detailed background and task information on creating relationships, identifying relationship types, and using the relationship editor, see the WebSphere Integration Developer Information Center.

How relationships work

At run time, when maps or other WebSphere Process Server components run and need a relationship instance, the instances of the relationships are either created or retrieved, depending on the scenario. The relationship and role instance data can be manipulated through three means:

* WebSphere Process Server component Java snippet invocations of the relationship service APIs
* Relationship transformations in the WebSphere Process Server business object mapping service
* Using the relationship manager tool

The relationship and role instance data is saved in relationship tables that are stored in the database in the default data source that you specify when you configure the relationship service.

The relationship service runs on each server at the cell level. The **Relationship Manager** home page **About** section shows the number of servers in the cell that

are running relationship services; the **Relationships** section shows each server name that is running relationship services. Before working with relationship instances, you need to select the server that has the instances of the relationships and roles you want to manage.

For detailed information on using the relationship manager, see the topics on the relationship manager in the WebSphere Process Server Information Center.

The following topics describe the configuration tasks to perform for the relationship services for your WebSphere Process Server environment.

Configuring the relationship service

After installing the product, you need to set the configuration properties for the relationship service.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as a configurator or an administrator to perform this task. Any WebSphere security role can view this configuration.

About this task

To set the data source and query block size (relationship instance count) properties for the relationship service, perform the following steps.

Procedure
1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager.**
3. Click **Relationship Services configuration**.

   The configuration tabbed page displays, showing the name and version (read-only) of the currently installed relationship service.
4. In the **Query block size (relationship instance count)** field, specify the maximum cache that the relationship service should set aside for relationship queries. This setting determines the size of the query results set. By default, 5000 relationship instances are read at once. This field controls the server size memory usage and provides the administrator with a level of control over how much memory resource is consumable by any given query.
5. In the **Data source** field, specify the default data source for the relationship service by entering the Java Naming and Directory Interface (JNDI) name of a data source defined at the cell level. This is where the tables for the relationship service are stored. Each relationship-related schema is created in this data source by default.
6. You then have the following options:
   - Click **OK** to save your changes and return to the previous page.
   - Click **Reset** to clear your changes and restore the currently configured values or most recently saved values.
   - Click **Cancel** to discard any unsaved changes on the page and return to the previous page.

Viewing relationships managed by the relationship service

Perform this task to view a list of the existing relationships that this relationship service manages.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, any WebSphere security role can view this configuration.

About this task

To view the relationship list, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager.**
3. Click **Relationship Services configuration > Relationships**.

   The Relationship collection page displays. Each row shows the version and data source for the associated relationship.

   Tip: To customize the number of rows that display at one time, click **Preferences**. Modify the **Maximum rows** field value and click **Apply**. The default is 25. The total relationship count managed by this relationship service is displayed at the bottom of the page.

What to do next

To see the configuration properties for a relationship, click the relationship name in the relationship collection table.

Viewing relationship properties

Perform this task to view the configuration properties that the relationship service manages at both the relationship service level–as it applies to the relationship service–and at the individual relationship level–as it applies to individual relationships.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, any WebSphere security role can view this configuration.

About this task

To view the configuration properties, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Click **Relationship Services configuration > Relationships**.
4. In the relationship collection table, click the name of the relationship whose properties you want to view.

The configuration tabbed page displays, showing the name, version, and data source currently in use for the relationship (read-only).

Note: The version is used for migration purposes. If the old relationship data needs to coexist in the new system, then the old infrastructure version will be set to the old version. Otherwise, it will be set to the current version.

5. To return to the Relationship collection page, click **Back**.

Administering extended messaging resources

Use the administrative console to configure resources needed by the extended messaging service and the applications that use the service. You can enable the extended messaging service, configure listener port extensions to handle late responses, and add or modify input and output ports for applications that use extended messaging.

Extended messaging enables container-managed messaging. It extends the base Java Message Service (JMS) support, the Enterprise Java Bean (EJB) component model, and support for EJB 2.0 message-driven beans to allow use of the existing container-managed persistence and transactional behavior.

Extended messaging uses the bean-managed messaging implementation to provide the JMS interfaces, which ensures that both bean-managed and extended messaging use consistent JMS support. JMS usage is simplified since its support is managed by the extended messaging service.

For a complete description of extended messaging, see the following articles in the WebSphere Business Integration Server Foundation information center:
- Extended messaging: Overview
- Using extended messaging in applications

Note: Before using the Extended Messaging Service, note the following limitations:
- The Extended Messaging Service feature has been deprecated in WebSphere Process Server 6.x. Although you can continue to use extended messaging with new and existing applications in this release, consider replacing these applications with ones that use standard JMS APIs, or replace them with equivalent messaging technologies.
- i5/OS The Extended Messaging Service feature is not supported on i5/OS systems.

Enabling the Extended Messaging Service

Enable the Extended Messaging Service to provide runtime support for container-managed messaging (extended messaging). Use the Extended Messaging Service page to specify whether this service starts automatically when the application server starts or whether it must be started manually.

About this task

Required security role for this task: When security and role-based authorization are enabled, you must log in as an administrator or configurator to perform this task.

To enable the Extended Messaging Service, perform the following steps.

Procedure

1. Ensure that the administrative console is running.

2. Click **Servers > Application servers >** *server_name* **> Extended Messaging Service** to display the Extended Messaging Service page.

3. If you want to enable the Extended Messaging Service to start automatically with server startup, select the **Enable service at server startup** check box. If you want to start the service manually, ensure the check box is cleared.

4. Click **OK**.

5. When prompted, click **Save** on the console task bar to save your changes to the master repository.

6. If you are using the WebSphere MQSeries bindings transport in your configuration, set the value of the MQ_INSTALL_ROOT environment variable as follows:

   a. From within the administrative console, click **Environment** → **WebSphere Variables**.

   b. Click MQ_INSTALL_ROOT to display the configuration page for the environment variable.

   c. In the **Value** field, delete the default value (`${WAS_INSTALL_ROOT}/lib/WMQ`) and replace it with the explicit installation path (for example, `D:/IBM/WebSphereMQ` on a Windows system).

   d. Click **OK**.

7. Stop and restart the application server in order for the changes to take effect.

Configuring listener port extensions to handle late responses

To enable a listener port to handle late responses, configure an extension that specifies how often the port checks for responses and how long it waits for those responses.

About this task

Late responses occur when the messaging infrastructure delays a response to a message sent by a sender bean, thereby preventing the application from receiving that response. Extended messaging can retrieve these late response messages and pass them to a message-driven bean provided by the application to handle late responses.

Required security role for this task: When security and role-based authorization are enabled, you must log in as an administrator or configurator to perform this task.

To create and enable a listener port extension that handles late responses, perform the following steps.

Procedure

1. Ensure you have a listener port defined and configured, and that you have deployed the sender bean with the **Handle late responses** option enabled.

   Note: For more information about deploying a sender bean with this option enabled, refer to the WebSphere Business Integration Server Foundation Information Center.

2. From the administrative console, click **Servers > Application servers >** *server_name* **> Extended Messaging Service > Listener Port Extensions**.

3. From the Listener Port Extensions page, click **New** to create a new listener port extension.
4. From the New Listener Port Extension page, select the **Enabled** check box to enable the extension and late response handling.
5. In the **Request Interval** field, either accept the default value or specify a new value to indicate how often the listener port checks for late responses.
6. In the **Request Timeout** field, either accept the default value or specify a new value to indicate how long the listener port waits for a late response. The listener port discards any responses received after the specified timeout value.
7. Use the **Listener Ports** drop-down menu to specify the listener port to use for the extension.
8. Click **OK**.
9. When prompted, click **Save** on the console task bar to save your changes to the master repository.
10. Stop and restart the application server in order for the changes to take effect.

What to do next

After you create a listener port extension, you can modify its configuration as necessary by using the Listener Port Extensions settings page.

Selecting extended messaging providers

Select the extended messaging provider you want to administer by clicking the appropriate scope on the Extended Messaging Provider page. Each scope (cell, node, and server) that contains applications that use extended messaging has its own extended messaging provider to manage resources. You can create, modify or delete input ports, output ports, or other custom properties for each provider.

About this task

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as administrator, operator, configurator, or monitor to perform this task.

To select the extended messaging provider you want to administer, perform the following steps.

Procedure
1. From the administrative console, click **Resources > Extended Messaging Provider**.
2. From the Extended Messaging Provider page, select the appropriate scope for the extended messaging provider you want to administer.
   - **Cell**: The most general scope. Extended messaging resources defined at the cell scope are visible from all nodes and servers, unless they have been overridden.
   - **Node**: Extended messaging resources defined at the node scope override any duplicates defined at the cell scope. They are visible to all servers on the same node, unless they have been overridden at a server scope on that node.
   - **Server**: Extended messaging resources defined at the server scope override any duplicate definitions defined at the cell or parent node scope. They are visible only to a specific server.

For detailed information about scopes, see the WebSphere Application Server Information Center.

3. Click **Apply**.

Results

The administrative console updates the **Scope**, **Name**, and **Description** fields on the bottom of the page to reflect the values for the selected resource provider.

What to do next

You can now create, modify or delete input ports, output ports, or other custom properties for the selected extended messaging provider.

Configuring input ports

Use the administrative console to create new or modify existing input ports for each receiver bean that is constructed from a session bean. Input ports define properties for the receiving Java Message Service (JMS) destination, specify how to select and handle messages, and provide details for any required reply destinations.

About this task

Note: You do not need to create input ports for receiver beans that are constructed from message-driven beans; the necessary details are associated with the deployed message-driven bean and the Message Listener Service.

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as administrator or configurator to perform this task.

To add or modify an input port, perform the following steps.

Procedure

1. From the administrative console, click **Resources > Extended Messaging Provider**.
2. From the Extended Messaging Provider page, select the appropriate scope for the resource provider you want to work with.
3. Click **Apply**.
4. Click **Input Ports** from the Additional Properties table.
5. From the Input Port collection page, do one of the following:
   - If you are creating a new input port, click **New**.
   - If you want to modify an existing input port, click the port name.
6. From the Input Port settings page, specify the appropriate properties for the input port.
7. Click **OK**.
8. When prompted, click **Save** on the console task bar to save your changes to the master repository.
9. Stop and restart the application server in order for the changes to take effect.

Input port settings:

When you create a new input port or modify an existing input port, you must specify certain properties. Use the information in this topic to determine whether a property is optional or required and what data type it accepts.

An input port has the following configuration properties:

Scope   The scope at which the extended messaging provider is defined. The value represents the location of the configuration file. The administrative console automatically populates this field. You cannot edit the value.

Name   The name of the input port, used for administrative purposes. This field requires a string value.

JNDI Name
The Java Naming and Directory Interface (JNDI) name for the input port. This field requires a string value.

Description
A description of the input port, used for administrative purposes. This field is optional, and it accepts a string value.

Category
A category string to use when classifying or grouping the resource. This field is optional, and it accepts a string with a maximum of 30 ASCII characters.

JMS Connection Factory JNDI Name
The JNDI name for the Java Message Service (JMS) connection factory used by the input port. This field requires a string value (for example, `jms/connFactory1`).

JMS Destination JNDI Name
The JNDI name for the JMS destination used by the input port. This field requires a string value (for example, `jms/destn1`).

JMS Acknowledgement Mode
The JMS mode that is used to acknowledge messages. This field is required for message-driven beans that use bean-managed transaction demarcation (in other words, the transaction type is set to `Bean`).

The following are valid values for this field:

- Auto Acknowledge: The session automatically acknowledges a message in either of the following cases:
  - When the session successfully returns from a call to receive a message
  - When the session calls a message listener to process the message and receives a successful response from that listener
- Dups OK Acknowledge: The session acknowledges only the delivery of messages. This can result in the delivery of duplicate messages if JMS fails.

The default mode is Auto Acknowledge.

Destination Type
The JMS resource type. This field requires one of the following values: :

- Queue: The receiver bean receives messages from a queue destination.
- Topic: The receiver bean receives messages from a topic destination.

The default value is Queue.

Subscription Durability

Specifies whether a JMS topic subscription is durable. This field is required if the JMS destination type is a topic. The following are valid values for this field:

- Durable: A subscriber registers a durable subscription with a unique identity that is retained by JMS. Subsequent subscriber objects with the same identity resume the subscription in the state in which it was left by the previous subscriber. If there is no active subscriber for a durable subscription, JMS retains the subscription's messages until they are received or they expire.

- NonDurable: Nondurable subscriptions last for the lifetime of their subscriber. A client sees the messages published on a topic only while its subscriber is active. If the subscriber is inactive, the client misses the messages published on its topic.

The default value is NonDurable.

Reply JMS Connection Factory JNDI Name

The JNDI name of the JMS connection factory that is used for replies. This field requires a string value (for example, `jms/connFactory1`).

Reply JMS Destination JNDI Name

The JNDI name of the JMS destination that is used for replies. This field requires a string value (for example, `jms/destn1`).

Configuring output ports

Use the administrative console to create new or modify existing output ports for sender beans. Output ports specify the properties sender beans need to define the destinations for sent messages. They also specify optional properties when responses are expected. Output ports are associated with sender beans at deployment time.

About this task

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as administrator or configurator to perform this task.

To add or modify an output port, perform the following steps.

Procedure

1. From the administrative console, click **Resources > Extended Messaging Provider**.
2. From the Extended Messaging Provider page, select the appropriate scope for the extended messaging provider you want to modify.
3. Click **Apply**.
4. Click **Output Ports** from the Additional Properties table.
5. From the Output Port collection page, do one of the following:
   - If you want to add a new output port, click **New**.
   - If you want to modify an existing output port, click the port name.
6. From the Output Port settings page, specify the appropriate properties for the output port.
7. Click **OK**.
8. When prompted, click **Save** on the console task bar to save your changes to the master repository.

9. Stop and restart the application server in order for the changes to take effect.

Output port settings:

When you create a new output port or modify an existing output port, you must specify certain properties. Use the information in this topic to determine whether a property is optional or required and what data type it accepts.

An output port has the following configuration properties:

Scope The extended messaging provider scope; the value represents the location of the configuration file. The administrative console automatically populates this field. You cannot edit the value.

Name The name of the output port, used for administrative purposes. This field requires a string value.

JNDI Name
> The Java Naming and Directory Interface (JNDI) name for the output port. This field requires a string value.

Description
> A description of the output port, used for administrative purposes. This field is optional, and it accepts a string value.

Category
> A category string to use when classifying or grouping the resource. This field is optional. It accepts a string value with a maximum of 30 ASCII characters.

JMS Connection Factory JNDI Name
> The JNDI name for the Java Message Service (JMS) connection factory used by the output port. This field requires a string value (for example, `jms/connFactory1`).

JMS Destination JNDI Name
> The JNDI name for the JMS destination used by the output port. This field requires a string value (for example, `jms/destn1`).

JMS Delivery Mode
> The JMS mode used to deliver messages. You must select one of the following values for this field:
> * Persistent: Messages put onto the destination are persistent.
> * Nonpersistent: Messages put onto the destination are not persistent.
>
> The default value is Persistent.

JMS Priority
> The message priority for the queue destination. This field requires an integer value from 0 to 9. The default value is 4.

JMS Time To Live
> The time, in milliseconds, a message remains in the queue. After the specified time elapses, the message expires.
>
> This field requires an integer with a value from 0 to n:
> * 0: Messages never time out.
> * n: Messages time out after n milliseconds.
>
> The default value is 0.

JMS Disable Message I.D.
Specifies whether the system generates a JMS message ID. This is a required field; select one of the following values:

- Selected: The system does not generate JMS message IDs.
- Cleared: The system generates JMS message IDs automatically.

By default, JMS message IDs are generated.

JMS Disabled Message Timestamp
Specifies whether the system generates a JMS message timestamp. This is a required field; select one of the following values:

- Selected: The system does not add message timestamps to sent messages.
- Cleared: The system automatically adds message timestamps to sent messages.

By default, the system adds message timestamps to sent messages.

Response JMS Connection Factory JNDI Name
The JNDI name of the JMS connection factory that is used for responses handled by the output port. This field requires a string value (for example, `jms/connFactory1`).

Response JMS Destination JNDI Name
The JNDI name of the JMS destination that is used for responses handled by the output port. This field requires a string value (for example, `jms/destn1`).

Chapter 11. Working with events

Events are requests or responses sent from one component to another. You can process events in a specific sequence; in addition, you can view, discard, modify, or resubmit any events that fail.

Processing events in sequence

Event sequencing guarantees that WebSphere Process Server components process events from asynchronous invocations in the order in which they are delivered. Event order is maintained throughout the entire business integration scenario.

An *event* is a request or a response that is sent from one component to another. The event encapsulates data and invocation metadata (for example, the name of the target component, the operation, and the parameters).

Note: Event sequencing is supported only for requests sent with an asynchronous invocation.

Why use event sequencing?

Some implementations require the target component to process events in the same order in which they were sent by the source application; processing them out of order can cause errors or exceptions. For example, if a source application generates an event to create a business object and then generates an event to update that business object, the create event must be processed first.

In an asynchronous invocation, events are stored in destinations on a service integration bus and can be handled by multiple instances of Message Driven Beans (MDBs). As a result, they may be processed non-sequentially, which can cause failures. To avoid this problem, use event sequencing.

How does event sequencing work?

Enable event sequencing by using the *event sequencing qualifiers* available in WebSphere Integration Developer. The qualifiers must be set on each method that requires event sequencing; they tell the runtime environment that invocations to these methods need to be sequenced.

Each qualifier has an event sequencing key that determines how events are sequenced. The key's value is comprised of one or more attributes of the business objects associated with an invocation. All events that share the same key are grouped together and processed in sequence. Events that do not have an event sequencing key continue to be processed as normal, in parallel with the sequenced events.

A sequenced event acquires a lock before being sent to the target component for processing. As soon as the business logic for the event has executed, the lock is released and given to the next event with the same event sequencing key. If the event cannot acquire the necessary lock, the execution of the invocation is suspended until the lock is acquired.

Related concepts

"Example: Event sequencing"
To understand how event sequencing works, consider a situation in which a source application (Component A) asynchronously invokes a target application (Component B) to create new orders, and then updates those orders with revised data.

"Considerations for implementing event sequencing" on page 348
Use the information in these topics to help you plan for, implement, and troubleshoot event sequencing in your business integration environment.

Example: Event sequencing

To understand how event sequencing works, consider a situation in which a source application (Component A) asynchronously invokes a target application (Component B) to create new orders, and then updates those orders with revised data.

Component A looks up Component B and invokes the create method to create an order, using the Order business object. The Order business object has the following attributes:

Attribute	Type
ID	string
customer	string
productName	string
quantity	integer

Component A then calls the update method to update the data in the newly created order.

In this example, assume there are five separate events that have been sent from Component A to Component B in the order specified below:
- Create1: This invocation calls the create method and passes the Order business object with an ID of 1 and quantity of 10.
- Create2: This invocation calls the create method and passes the Order business object with an ID of 2 and a quantity of 8.
- Update1: This invocation calls the update method and passes the Order business object with an ID of 1 and a quantity of 15.
- Update2: The third invocation calls the update method and passes the Order business object with an ID of 1 and a quantity of 12.
- Update3: This invocation calls the update method and passes the Order business object with an ID of 2 and a quantity of 10.

For each event, a message is put onto a service integration bus destination in the same order as the invocations. A Message Driven Bean (MDB) reads the message and sends it to the target component (in this case, Component B) for processing. Although there is only one MDB per module, there are multiple instances of that MDB and these five messages are processed in parallel. It is possible that the MDB thread that is processing the message for Update2 will complete before the thread that is processing the message for the Create1 event; if this happens, the Update2 event will fail because the order has not yet been created.

To prevent these sorts of errors, this example implements event sequencing. In the sample component definition below, event sequencing qualifiers are specified for both the create and update methods. Both of these methods use the same event

sequencing key (set to the ID attribute of the Order business object) and are placed into the same event sequencing group. The third method, retrieve, is not sequenced.

```
<interfaces>
   <interface xsi:type="wsdl:WSDLPortType" portType="ns1:ProcessOrder">
      <method name="create">
         <scdl:interfaceQualifier xsi:type="es.EventSequencingQualifier">
         <es:eventSequencing sequencingGroup="default" conintueOnError="true">
            <keySpecification>
               <parameter name="Order">
                  <xpath>ID</xpath>
               </parameter>
            </keySpecification>
         </es:eventSequencing>
         </scdl:interfaceQualifier>
      </method>
      <method name="update"/>
         <scdl:interfaceQualifier xsi:type="es:EventSequencingQualifier">
         <es.eventSequencing sequencingGroup="default' continueOnError="true">
            <keySpecification>
               <parameter name="Order">
                  <xpath>ID</xpath>
               </parameter>
            </keySpecification>
         </es:eventSequencing>
         </scdl:interfaceQualifier>
      <method name="retrieve/">
   </interface>
</interfaces>
```

With event sequencing enabled, the five events in this example are processed as follows:

1. Component A sends the Create1 request. It is placed on the destination and handled by an instance of the MDB.
2. The Create1 event acquires a lock and is sent to Component B for processing.
3. Component A sends the Update1 request. It is placed on the destination and handled by an instance of the MDB.
4. The Update1 event tries to acquire a lock. If the Create1 event (which shares the same event sequencing key value as Update1) still has the lock, processing for this event is suspended until the lock on Create1 is released.
5. Component A sends the Create2 request. It is placed on the destination and handled by an instance of the MDB.
6. The Create2 request (which has a different value for the event sequencing key) acquires a lock and is sent to Component B for processing.
7. Component A sends the Update2 request. It is placed on the destination and handled by an instance of the MDB.
8. The Update2 event tries to acquire a lock. If either the Create1 or Update1 event (which share the same event sequencing key value as Update2) still holds a lock, processing for this event is suspended. It will not be processed until the Update1 event has acquired the lock, been processed, and the lock has been released.
9. Component A sends the Update3 request. If the Create2 event (which shares the same event sequencing key value as Update3) still has the lock, processing for this event is suspended until the lock on Create2 is released.

Related concepts

"Processing events in sequence" on page 345
Event sequencing guarantees that WebSphere Process Server components

process events from asynchronous invocations in the order in which they are delivered. Event order is maintained throughout the entire business integration scenario.

Considerations for implementing event sequencing

Use the information in these topics to help you plan for, implement, and troubleshoot event sequencing in your business integration environment.

Related concepts

"Processing events in sequence" on page 345
Event sequencing guarantees that WebSphere Process Server components process events from asynchronous invocations in the order in which they are delivered. Event order is maintained throughout the entire business integration scenario.

Supported components and invocations

Before you implement event sequencing, consider the types of invocations and components you are using and whether they support sequencing.

Event sequencing is supported for all requests from Service Component Architecture (SCA) components that meet the following requirements:
* Components must use Web Services Description Language (WSDL) interfaces.
* Components must use asynchronous invocation.

    Note: The client is responsible for maintaining event order before events are put on SCA destinations. If sequencing is required, the client must do the SCA invocations within a single thread.

It is not supported for responses.

You do not need to use event sequencing for events that are implicitly sequenced during a synchronous invocation to a component with a synchronous implementation. If the client is using a single thread for invocations, the call automatically waits until the target has finished processing the event. No further invocations can be made until the event is returned.

Declaring event sequencing for a component

After you have determined which methods on a component need to use event sequencing, use WebSphere Integration Developer to update the component definition to include an event sequencing qualifier to the each of those methods.

Event sequencing qualifiers extend types defined in the Service Component Definition Language (SCDL), enhancing the quality of service for Service Component Architecture (SCA) components. The event sequencing qualifier contains a keySpecification element to identify the events to sequence. There must be one keySpecification element for each method that uses event sequencing. The parameter element is used with each keySpecification; it indicates the business object attribute or attributes that will provide the value for the event sequencing key.

In addition, the event sequencing qualifier uses the sequencingGroup attribute to group methods that need to be sequenced together. All events that are generated by any method in the same group are processed sequentially.

In the example component definition below, event sequencing has been defined on the create and update methods. The keySpecification element for both defines the parameter as the ID attribute of the newOrder business object.

```
<interfaces>
   <interface xsi:type="wsdl:WSDLPortType" portType="ns1:ProcessOrder">
      <method name="create">
         <scdl:interfaceQualifier xsi:type="es.EventSequencingQualifier">
         <es:eventSequencing sequencingGroup="default" conintueOnError="true">
            <keySpecification>
               <parameter name="newOrder">
                  <xpath>ID</xpath>
               </parameter>
            </keySpecification>
         </es:eventSequencing>
         </scdl:interfaceQualifier>
      </method>
      <method name="update"/>
         <scdl:interfaceQualifier xsi:type="es:EventSequencingQualifier">
         <es.eventSequencing sequencingGroup="default' continueOnError="true">
            <keySpecification>
               <parameter name="newOrder">
                  <xpath>ID</xpath>
               </parameter>
            </keySpecification>
         </es:eventSequencing>
         </scdl:interfaceQualifier>
   </interface>
</interfaces>
```

When declaring event sequencing on a component, ensure that the component is invoked in a managed thread. The managed thread provides the session information required to properly sequence events.

Using event sequencing with export bindings

Event sequencing is supported with EIS, JMS, WebSphere MQ, and WebSphere MQ JMS export bindings. To ensure that the exports process and deliver messages in the correct sequence, you must configure the export bindings appropriately.

Consider the following requirements when using event sequencing on a target component that handles export bindings:

- An adapter component must use the non-optimized path for a Java Message Service (JMS) export when event sequencing is used on the target component.
- To enable event sequencing for JMS export bindings, you must limit the number of concurrent Message Driven Beans (MDBs) that are processing incoming messages. Do this by setting the maxConcurrency custom property on the ActivationSpec to a value of 1.
- To enable event sequencing for a WebSphere MQ JMS export , you must limit the number of concurrent listener threads that will deliver messages to the Message Driven Bean. Do this by setting the maxSessions property to a value of 1.
- To enable event sequencing for a native MQ export, you must use WebSphere Integration Developer to set the eventSequencing property.

  Related information

  Export bindings and event sequencing

  ⬆ Enabling event sequencing for an EIS Export binding

  ⬆ Enabling event sequencing for a JMS Export binding

⏏ Enabling event sequencing for a WebSphere MQ JMS Export bindings

Using event sequencing in a network deployment environment

Event sequencing can be used in a network deployment environment, with or without a high availability manager. Consult the table in this topic to ensure that your particular topology is supported.

Note that Service Component Architecture (SCA) destinations for any component using event sequencing cannot be partitioned. Therefore, if you are using clusters, you can have only one active messaging engine per cluster.

Table 34. Event sequencing support in a network deployment environment

Topology	Is event sequencing supported?
Standalone server	Yes
No clusters	Yes
Applications are clustered. Messaging engines and destinations are not clustered.	Yes
Messaging engines are clustered. Applications and destinations are not clustered.	Yes
Messaging engines and destinations are clustered. Applications are not clustered.	No. Clustered destinations are partitioned and cannot be used with event sequencing.
Applications and messaging engines are clustered (same cluster). Destinations are not clustered.	Yes
Applications, messaging engines, and destinations are clustered (same cluster).	No. Clustered destinations are partitioned and cannot be used with event sequencing.
Applications and messaging engines are clustered (different clusters). Destinations are not clustered.	Yes
Applications, messaging engines, and destinations are clustered (different clusters).	No. Clustered destinations are partitioned and cannot be used with event sequencing.

Using event sequencing in a high availability environment

High availability (HA) support means that system subcomponents, such as the event sequencing runtime, are made highly available and the workload can be distributed in the case of a node or daemon failure.

Although event sequencing requires a singleton service to process the event messages on a destination, an HA manager provides the necessary services to ensure that this process is not a single point of failure. Instead, the event sequencing runtime fails over to another server in the cluster in the event of a system crash.

Limitations in event sequencing

Certain types of components and invocations offer limited support for event sequencing.

Limitations for the current release of event sequencing include the following:
- Event sequencing on operations bound to a Business Process Execution Language (BPEL) process with a non-initiating receive is not recommended. In

long-lived business processes, event sequencing relies on a work completion contract to determine when to release a lock; this work completion contract is activated whenever a new process instance is created. However, no new process instance is created when there is a non-initiating receive. As a result, it is difficult for the event sequencing runtime to accurately detect a completed work contract and it can release the lock either too early or too late.

- Event sequencing on operations bound to a Business State Machine with a non-initiating receive is not recommended.
- Work completion contracts are supported only for BPEL components. To effectively use event sequencing on any other type of component that has asynchronous invocations, it is recommended that you use the request-response method signature. The event sequencing runtime interprets a response as a signal that the work is complete and releases the lock.

  Note: If you cannot declare a method as a request-response operation, you might need to specify event sequencing on downstream components, making sure you use the same event sequencing key for all methods.

Event sequencing is not supported in the following scenarios:
- Using unmanaged threads or non-SCA bindings to send events to their destinations without proper session context.
- Using synchronous invocations to components with asynchronous invocations.

Administering locks

The lock manager handles event sequencing locks. You can use the esAdmin command to list, delete, or unlock any lock in the lock manager.

The lock manager supports two operations on event locks:
- Lock: The lock operation attempts to acquire a lock and stores the lock request in a database. After a lock is granted, processing resumes for the invocation that requested the lock.
- Unlock: The unlock operation releases the current lock and grants the lock to the next lock request.

Requests for the same lock are put into a queue in the order in which they are received. Locks are persisted to the default WebSphere Process Server database and data source to ensure they can be recovered in the case of a server failure.

The esAdmin command enables you to administer the active and queued locks currently in the lock manager. The following sections provide more detail on using esAdmin.

Note: If you are using partitioned databases, run the esAdmin command once for each deployment target. In a clustered environment, you can run it on any cluster, but do not run it on the deployment manager.

Listing locks

The esAdmin command can list all active and queued locks in the lock manager, or only those locks associated with a specific module, component, or method.

Use one of the following methods with esAdmin:
- listAll: Lists all active and queued locks in the lock manager.

- listLocks: Lists a subset of the active and queued locks in the lock manager. Specify one or more of the following parameters to return a filtered list of locks:
  - moduleName
  - componentName
  - methodName

  For example, use

  ```
  esAdmin listLocks CustMod CustComp
  ```

  to return a list of active and queued locks for the CustComp component that is part of the CusMod module.

The command returns output that looks like the following:

Table 35. Sample output from esAdmin listLocks command

Lock Id	Sequence Id	Owner Id	Module	Component	Method	System Message Id
7564504	2	695376	CustMod	CustComp	createCust	A09-427BE_5002
7564504	3	232757	CustMod	CustComp	createCust	ADF-053RT_5004

In the output above, the sequence ID is the order in which the lock requests are queued; the lowest number in the sequence currently holds the lock. The system message ID specifies the ID for the corresponding service integration bus message; you can use this information to correlate lock requests with the messages on the destinations.

Releasing locks

Use the esAdmin command to release a single lock, as follows:

```
esAdmin unlock lockId
```

where *lockId* is the unique lock ID returned by the `esAdmin listLock` or `esAdmin listAll` command.

This command is useful when you encounter a deadlock; you can release the lock that is deadlocked and grant it to the next lock request in the queue.

Deleting locks

If you need to delete one or more locks, first stop the module associated with the lock. Then, use the esAdmin command to delete the lock from the database, as follows:

```
esAdmin deleteLocks moduleName
```

You must restart the module in order for the destinations to resume processing event messages.

Use the esAdmin deleteLocks command with caution. All locks in the specified module are deleted from the lock manager database.

Troubleshooting event sequencing

Refer to the information in this topic if you are experiencing difficulty with event sequencing.

Problems with the event sequencing qualifier

Ensure that your component definition is correct:

- Is the event sequencing qualifier set on the method? Event sequencing validation fails if the qualifier is erroneously set on the interface.
- Is the parameter name valid?
- Is the xpath element valid, and does it correctly resolve to a primitive?
- Is there a single eventSequencing element for the method? Each method supports only one eventSequencing element.
- Is there a single keySpecification element for the method? Each method supports only one keySpecification element.

Deadlocks

Deadlocks occur when an invoked operation with a lock invokes another operation on the same component using the same event sequencing key and group. You can resolve a deadlock by using the esAdmin command to list and release the current lock.

To avoid deadlocks, carefully consider dependencies when implementing event sequencing. Ensure that operations with circular dependencies are in different event sequencing groups.

Deadlocks with a BPEL process

Deadlocks can occur when event sequencing is used with Business Process Execution Language (BPEL) processes. Deadlocks are caused by setting event sequencing qualifiers on operations that correspond to both of the following activities:

- Multiple instantiating receive or pick activities, where the `createInstance` attribute is set to `yes`
- Correlation set specifications with an `initiation` attribute set to `join`

Resolve this type of deadlock by using the esAdmin command to list and release the current lock. To prevent further deadlocks, ensure that these types of dependent operations are put into different event sequencing groups.

Performance issues

If you are experiencing memory problems on the messaging engine server used for event sequencing components, try modifying the runtime event sequencing property maxActiveMessages in the *install_root*/properties/ eventsequencing.properties file.

The maxActiveMessages property defines the number of messages currently locked on a component destination; too many large messages can negatively affect performance and cause memory problems. Note that a value of 0 (zero) means that an unlimited number of messages are allowed. By default, the maxActiveMessages property is set to 100.

To modify the maxActiveMessages property, perform the following steps.
1. Open the eventsequencing.properties file in a text editor.
2. Make the appropriate modifications for your environment.

3. Save and close the file.
4. Stop and restart any applications that are part of the event sequencing component in order for the changes to take effect.

Managing failed events

The WebSphere Process Server Recovery service monitors for failed operations between Service Component Architecture (SCA) components. If an operation fails, the Recovery service captures data about the event and the failure. You can then use the failed event manager to view, modify, resubmit, or delete the failed event.

What is a failed event?

In the context of WebSphere Process Server, an event is a request that is received by a WebSphere Process Server application. It can come from an external source (such as an inbound application adapter) or an external invocation to a Web service. The event is comprised of a reference to the business logic it wants to operate and its data, stored in a Service Data Object (a business object). When an event is received, it is processed by the appropriate WebSphere Process Server application business logic.

A single thread of execution can branch off into multiple branches (or threads); the individual branches are linked to the main invoking event by the same session context.

If this business logic in one of these branches cannot execute completely due to system failure, component failure, or component unavailability, the event moves into the failed state. If multiple branches fail, a failed event is created for each. The WebSphere Process Server Recovery service handles the following types of failed events:

- Event failures that occur during an asynchronous invocation of a Service Component Architecture (SCA) operation
- Event failures that are caused by a runtime exception (in other words, any exception that is not declared in the methods used by the business logic)

The Recovery service does not handle failures from synchronous invocations or from Business Process Execution Language (BPEL) asynchronous request/reply invocations.

Failed events typically have source and destination information associated with them. The source and destination are based on the failure point (the location where the invocation fails), regardless of the type of interaction. Consider the following example, where Component A is asynchronously invoking Component B. The request message is sent from A to B, and the response message is sent from B to A.

- If the exception occurs during the initial request, Component A is the source and Component B is the destination for the purposes of the failed event manager.
- If the exception occurs during the response, Component B is the source and Component A is the destination for the purposes of the failed event manager.

This is true for all asynchronous invocations.

How are failed events managed?

The Recovery service sends failed asynchronous interactions to failed event destinations that have been created on the SCA system bus (SCA.SYSTEM.*cell_name*.Bus). The data for failed events is stored in the failed event

database (by default, WPCRSDB) and is made available for administrative purposes through the failed event manager interface.

An administrator uses the failed event manager to browse and manage all WebSphere Process Server failed events. Common tasks for managing failed events include:

- Browsing all failed events
- Searching for failed events by specific criteria
- Editing data for a failed event
- Resubmitting failed events
- Deleting failed events

To access the failed event manager, click **Integration Applications** ↳ **Failed Event Manager**.

Security considerations for recovery

If you have enabled security for your WebSphere Process Server applications and environment, it is important to understand how role-based access and user identity affect the Recovery subsystem.

Role-based access for the failed event manager
> The failed event manager uses role-based access control for the failed event data and tasks. Only the administrator and operator roles are authorized to perform tasks within the failed event manager. Users logged in as either administrator or operator can view all data associated with failed events and can perform all tasks.

Event identity and user permissions
> A failed event encapsulates information about the user who originated the request. If a failed event is resubmitted, its identity information is updated to reflect the user who resubmitted the event. Because different users logged in as administrator or operator can resubmit events, these users must be given permissions to the downstream components required to process the event.

For more information about implementing security, see Securing applications and their environment.

Finding failed events

Failed events are stored in the failed event database and are retrieved through the search functionality of the failed event manager. You can search for all failed events on the all the servers within the cell, or for a specific subset of events.

Before you begin

You must be logged in as administrator or operator to perform this task.

About this task

This topic describes how to find all failed events on the server, with references to topics for conducting other searches based on session ID, source, destination, date, business object type, exception text, or a combination of those criteria.

To retrieve a complete list of failed events, use the following procedure.

Procedure

1. Ensure the administrative console is running.
2. Click **Integration Applications → Failed Event Manager** to enter the failed event manager.
3. Click **Failed events on this server → Get all failed events**.

Results

The Search Results page opens, displaying a list of all the WebSphere Process Server failed events in the cell.

What to do next

You can now view and modify data in a failed event, resubmit it, or delete it.

Searching for failed events by session ID

Use the Search page's **By Session** tab to find only those events that failed within a specific session.

Before you begin

You must be logged in as administrator or operator to perform this task.

About this task

Every event executes within a session; if that event fails, the failed event manager encapsulates specific session information for the failed execution branch in the Session ID parameter. The same session ID is given to all resources and processes that are part of a session, including Common Base Events and business processes.

To search for failed events by session ID, use the following procedure.

Procedure

1. Ensure the administrative console is running and then click **Integration Applications → Failed Event Manager**, to enter the failed event manager.
2. From the main failed event manager page, click **Search by session**.
3. Use the **Session ID** field to specify the session ID to search against.
4. Click **OK** to begin the search.

Results

The Search Results page opens, listing all failed events that originated during the specified session.

Searching for failed events by destination

Use the **By Destination** tab on the Search page to find only those failed events that are associated with a specific destination module, component, or method. The failed event manager determines the destination based on the point of failure, regardless of the type of interaction.

Before you begin

You must be logged in as administrator or operator to perform this task.

About this task

When performing a search, note the following:
- The values for the fields are case sensitive.
- The fields accept the asterisk (*) wildcard character.
- If you leave any field on this tab blank, the blank field is treated as a wild card. The failed event manager will search in all components, modules, or methods.
- You can search on a single destination criteria or on multiple criteria. Searching on two or more of the destination criteria provides a more refined list of failed events.

To search for failed events by destination, use the following procedure.

Procedure
1. Ensure the administrative console is running, and then click **Integration Applications** → **Failed Event Manager** to enter the failed event manager.
2. From the main failed event manager page, click **Search by destination**.

   The Search page opens with the **By Destination** tab selected.
3. Specify the search criteria you want to use. You can use any combination of the following fields to customize your search:
   - **Destination module**: use this field to specify the failed event's destination module.
   - **Destination component**: use this field to specify the failed event's destination component.
   - **Destination method**: use this field to specify the failed event's destination method.
4. Click **OK** to begin the search.

Results

The Search Results page opens and displays a list of all failed events that were destined for the specified module, component, or method.

Searching for failed events by source

Use the **By Source** tab on the Search page to find only those failed events that originated from a specific source module, component, or both. The failed event manager determines the source based on the point of failure, regardless of the type of interaction.

Before you begin

You must be logged in as administrator or operator to perform this task.

About this task

When performing a search, note the following:
- The values for the fields are case sensitive.
- The fields accept the asterisk (*) wildcard character.
- If you leave either field on this tab blank, the blank field is treated as a wild card. The failed event manager will search in all components or modules.
- To get the most refined list of failed events, use both the **Source module** and **Source component** fields.

To search for failed events by source, use the following procedure.

Procedure
1. Ensure the administrative console is running, and then click **Integration Applications** → **Failed Event Manager** to enter the failed event manager.
2. From the main failed event manager page, click **Search by source**.
   The Search page opens with the **By Source** tab selected.
3. Specify the search criteria. You can use one or both of the following fields:
   - **Source module**: use this field to specify the module that the failed event originated from.
   - **Source component**: use this field to specify the component that the failed event originated from.
4. Click **OK** to begin the search.

Results

The Search Results page opens and displays a list of all failed events that originated from the specified module, component, or both.

Searching for failed events by date
Use the **By Date** tab on the Search page to find only those events that failed during a specific time period.

Before you begin

You must be logged in as administrator or operator to perform this task.

About this task

When performing a search by date, note the following:
- The format for the date and time are locale-specific. An example of the appropriate format is provided with each field.

  Note: The values you supply must match the required format exactly. If you provide an incorrectly formatted value, the failed event manager displays a warning and substitutes the default value for that field.
- The time is always local to the server. It is not updated to reflect the local time of individual machines running the administrative console.
- You must specify a value for both fields on this tab.

To search for failed events by date, use the following procedure.

Procedure
1. Ensure the administrative console is running, and then click **Integration Applications** → **Failed Event Manager** to enter the failed event manager.
2. From the main failed event manager page, click **Search by date**.
3. Use the **From Date** field to specify the starting date and time. Because the required format for the value varies by locale, the failed event manager provides a locale-appropriate example above this field. Ensure the value you enter is formatted in the same manner as the example provided. (For instance, the required format for the en_US locale is *MM/DD/YY HH:MM Meridiem*; therefore, a correctly formatted value for this field looks like 11/10/05 4:30 PM.)

4. Use the **To Date** field to specify the ending date and time. Because the required format for the value varies by locale, the failed event manager provides a locale-appropriate example above this field. Ensure the value you enter is formatted in the same manner as the example provided. (For instance, the required format for the en_US locale is *MM/DD/YY HH:MM Meridiem*; therefore, a correctly formatted value for this field looks like 11/17/05 4:30 PM.)

5. Click **OK** to begin the search.

   The Search Results page opens and displays a list of all failed events that originated during the specified time period.

Searching for failed events by business object type

Use the **By Type** tab of the Search page to find only those failed events that are associated with a specific business object.

Before you begin

You must be logged in as administrator or operator to perform this task.

About this task

To search for failed events by business object type, use the following procedure.

Procedure

1. Ensure the administrative console is running, and then click **Integration Applications** → **Failed Event Manager** to enter the failed event manager.

2. From the main failed event manager page, click **Search by business object type**.

   The Search page opens with the **By Type** tab selected.

3. Specify the business object type you want to search against by using one of the following items:

   • **Select the business object type**: use this drop-down menu to select the type of business object associated with the failed events. This menu contains a list of all business object types found in the failed events on the server.

   • **Other business object type**: use this field to specify the type of business object associated with the failed events. The field accepts the asterisk (∗) wildcard character. All values are case sensitive.

4. Click **OK** to begin the search.

Results

The Search Results page opens and displays a list of all failed events that are associated with the specified business object type.

Searching for failed events by exception

Use the **By Exception** tab of the Search page to find only those failed events that are associated with a specific exception. You can specify part or all of the exception text.

Before you begin

You must be logged in as administrator or operator to perform this task.

About this task

To search for failed events by exception type, use the following procedure.

Procedure

1. Ensure the administrative console is running, and then click **Integration Applications** → **Failed Event Manager** to enter the failed event manager.
2. From the main failed event manager page, click **Search by exception text**.

   The Search page opens with the **By exception** tab selected.
3. In the **Exception text** field, type the text associated with the exception you want to search against.

   You can specify all or part of the exception text, as well as the asterisk (*) wildcard character to make the search easier. The values in this field are case sensitive.

   Note: If you leave the **Exception text** field blank, it is treated as a wild card; all failed events are returned.
4. Click **OK** to begin the search.

Results

The Search Results page opens and displays a list of all failed events that are associated with the specified exception text.

Performing an advanced search for failed events

Use the **Advanced** tab of the Search page to perform a more refined search for failed events by using a combination of the criteria found on the other search tabs.

Before you begin

You must be logged in as administrator or operator to perform this task.

About this task

Note the following:

- Unless otherwise noted below, all fields accept the asterisk (*) wildcard character.
- Leaving a field blank causes it to be treated as a wild card.
- The advanced search is not optimized; executing an advanced search on a large set of failed events can reduce performance.

To perform an advanced search, use the following procedure.

Procedure

1. Ensure the administrative console is running, and then click **Integration Applications** → **Failed Event Manager** to enter the failed event manager.
2. From the main failed event manager page, click **Advanced search**.

   The Search page opens with the **Advanced** tab selected.
3. Specify the search criteria you want to use. You can use any combination of the following fields to customize your search:

   - **Destination module**: use this field to specify the failed event's destination module.
   - **Destination component**: use this field to specify the failed event's destination component.

- **Session ID**: use this field to specify the session in which the event executed. This field does not accept the asterisk (∗) wildcard character.
- **Destination method**: use this field to specify the failed event's destination method.
- **Source module**: use this field to specify the module that the failed event originated from.
- **Source component**: use this field to specify the component that the failed event originated from.
- **From Date**: use this field to specify the starting date and time if you want to search within a specific time period. This field does not accept the asterisk (∗) wildcard character.
- **To Date**: use this field to specify the ending date and time if you want to search within a specific time period. This field does not accept the asterisk (∗) wildcard character.
- **Business object type**: use this field to specify the type of business object associated with the failed events.
- **Exception text**: use this field to specify the text associated with the exception you want to search against.

4. Click **OK** to begin the search.

Results

The Search Results page opens and displays a list of all failed events that meet the specified criteria.

Working with data in failed events

Each failed event has data associated with it; often, that data can be edited before an event is resubmitted. There are two basic types of data for a failed event: data about the event, and business data.

Data about the failed event

Each failed event has the following data associated with it:
- The unique message ID and session ID for the event
- The service invocation type between SCA components
- The names of the module and component from which the event originated (the source). The failed event manager determines the source of an event based on the location where the invocation failed.
- The names of the destination module, component and method for the event. The failed event manager determines the event's destination based on the location where the invocation failed.
- The time the event failed
- The exception thrown when the event failed

This data cannot be edited. In addition, failed events can have associated trace and expiration data, both of which can be edited.

Business data

Events typically include business data. Business data can be encapsulated in a business object, or it can be simple data that is not part of a business object.

Business data is edited with the business data editor available in the failed event manager.

Browsing data in failed events

Use the failed event manager to view failed event data and any business data associated with the event.

Before you begin

You must be logged as administrator or operator to perform this task.

About this task

Each failed event has two types of data associated with it:

- Failed event data: information about the failed event itself, including the source and destination for the event, the time it failed, the exception it failed with, its message and session IDs, and its trace and expiration settings.
- Business data: information contained in the event. The business data can be encapsulated in a business object, or it can be simple data that is not part of a business object.

To browse failed event data, use the following procedure.

Procedure

1. Ensure that the failed event manager is open and that you have retrieved a list of the failed events on your system.
2. From the Search Results page of the failed event manager, click the ID (found in the Message ID column) of the failed event whose data you want to browse.

   The Failed Event Details page opens and displays all of the information about the event.
3. If you want to browse the business data associated with the failed event, click **Edit business data**.

   The Business Data Editor collection page opens, displaying the business data associated with the failed event. Each parameter name in the hierarchy is a link. If the parameter is a simple data type, clicking its name will open up a form so you can edit the parameter's value. If the parameter is a complex data type, clicking its name will expand the hierarchy further.

Editing trace or expiration data in a failed event

The Failed Event Details page enables you to set or modify values for the trace control and expiration date associated with a failed event.

Before you begin

You must be logged in as administrator or operator to perform this task.

About this task

Important: Any edits you make to the trace or expiration data are only saved locally until you resubmit the event. If you perform any other action before resubmitting the event, all edits are lost.

Failed events can be resubmitted with trace to help you monitor the event processing. Tracing can be set for a service or a component, and it can be sent to a

log or to the Common Event Infrastructure (CEI) server. When you view the failed event data on the Failed Event Details page, the default trace value SCA.LOG.INFO;COMP.LOG.INFO is shown for the event. If you resubmit the event with this default setting, no trace occurs when the session calls an SCA service or executes a component.

Some failed events also have an expiration. If a user has specified an expiration with the asynchronous call that sends the event, that data persists even if the event fails, and the expiration time appears in the **Resubmit Expiration Time** field on the Failed Event Details page. Expired failed events cannot be resubmitted successfully. To prevent a second failure, you can edit the expiration date for the event to ensure that it is not expired when it is resubmitted.

To edit trace or expiration data in a failed event, use the following procedure.

Procedure
1. Ensure that the failed event manager is open and that you have retrieved a list of the failed events on your system.
2. From the failed event manager's Search Results page, click the ID (found in the Message ID column) of the failed event whose data you want to edit.

   The Failed Event Details page opens.
3. If the event has an expiration date that causes it to expire before it is resubmitted, edit the expiration in the **Resubmit expiration time** field.

   The expiration time shown is local to the server. The value for this field must be formatted according to your specified locale. An example of the correct format for your locale is provided above the field.
4. If you want to enable tracing for the failed event, specify a new value in the **Trace Control** field. For detailed information about trace values, see the Monitoring topics in the WebSphere Process Server Information Center.
5. Do one of the following:
   - If the edited data is correct and you want to resubmit the event, click **Resubmit** to make the changes at a server level.
   - If you want to remove the changes you made, click **Undo local changes**.

   The edited failed event is resubmitted for processing and is removed from the failed event manager.

   Related tasks

   "Finding failed events" on page 355
   Failed events are stored in the failed event database and are retrieved through the search functionality of the failed event manager. You can search for all failed events on the all the servers within the cell, or for a specific subset of events.

Editing business data in a failed event

Business data can be encapsulated into a business object, or it can be simple data that is not part of a business object. A failed event can have both simple data and a business object associated with it. Use the business data editor to edit the business data associated with a failed event before you resubmit it.

Before you begin

You must be logged in as administrator or operator to perform this task.

About this task

For each failed event, the editor displays the associated business data in a hierarchical format; the navigation tree at the top of the table is updated as you navigate through the parameters to give you a clear picture of where you are in the hierarchy.

You can edit only simple data types (for example, String, Long, Integer, Date, Boolean). If a data type is complex (for example, an array or a business object), you must navigate through the business data hierarchy until you reach the simple data types that make up the array or business object. Complex data is denoted by an ellipsis (...) in the Parameter Value column.

Important: Any edits you make to business data are saved locally. Changes are not made to the corresponding business data in the server until you resubmit the failed event.

To edit business data associated with a failed event, use the following procedure.

Procedure
1. Ensure that the failed event manager is open and that you have retrieved a list of the failed events on your system.
2. From the failed event manager's Search Results page, click the ID (found in the Message ID column) of the failed event whose data you want to edit.
   The Failed Event Details page opens.
3. From the Failed Event Details page, click **Edit business data** to access the Business Data Editor collection page.
   This page displays a hierarchical view of all of the data associated with the failed event.
4. Navigate through the business data hierarchy by clicking on the name of each parameter (these appear as links in the Parameter Name column). When you have located the parameter whose value you want to edit, click its name.
   If the parameter has an editable value, the Business Data Editor page opens.
5. In the **Parameter value** field, specify the new value for the parameter.
6. Click **OK**.
   The change is saved locally and you are returned to the Business Data Editor collection page.
7. If you want to remove the changes you made, click **Undo local business data changes**.
   All of the edits are removed and the business data is returned to its original state.
8. If the edited business data is correct, click **Resubmit** to make the changes at a server level.
   The edited failed event is resubmitted for processing and is removed from the failed event manager.

Resubmitting failed events

If you want to try to send an event again, you must resubmit it from the failed event manager. You can resubmit an event without changes, or you can edit the business data parameters before resubmitting it.

When a failed event is resubmitted, the processing resumes only for the failed branch, not for the entire event.

Tracing is available for resubmitted events to help monitor the event's processing. Tracing can be set for a service or a component, and its output can be sent to a log or to the Common Event Infrastructure (CEI) server.

You can also use the event's unique message ID to track its success or failure. If a resubmitted event fails again, it is returned to the failed event manager with its original message ID and an updated failure time.

Resubmitting an unchanged failed event

You can resubmit one or more unchanged failed events to be processed again. Processing resumes only for the failed branch, not for the entire event.

About this task

You must be logged in as administrator or operator to perform this task.

Procedure

1. Ensure that the failed event manager is open and that you have retrieved a list of the failed events on your system.
2. From the Search Results page, select the check box next to each failed event you want to resubmit.
3. Click **Resubmit**.

Results

Each selected event is resubmitted for processing and is removed from the failed event manager.

Resubmitting a failed event with trace

You can monitor the resubmission of a failed event to determine whether it now succeeds. The failed event manager provides optional tracing for all failed events.

About this task

Tracing can be set for a service or a component, and it can be output to a log or to the Common Event Infrastructure (CEI) server. For detailed information about setting and viewing trace, see the Monitoring topics in the information center.

You must be logged in as administrator or operator to perform this task.

Procedure

1. Ensure that the failed event manager is open and that you have retrieved a list of the failed events on your system.
2. From the Search Results page, select the check box next to each failed event you want to resubmit.
3. Click **Resubmit with trace**.
4. From the Resubmit with Trace page, specify the level of trace you want to use in the **Trace control** field.

   By default, the value is SCA.LOG.INFO;COMP.LOG.INFO. With this setting, no trace occurs when the session calls an SCA service or executes a component.
5. Click **OK** to resubmit the failed event and return to the Search Results page.

What to do next

To view the trace log for a resubmitted event, open the corresponding component logger or use the CEI log viewer.

Finding Common Base Events related to a failed event

A failed event can be related to one or more Common Base Events. The failed event manager provides a link to view related Common Base Events in the Common Base Event Browser.

Before you begin

You must be logged in as administrator or operator to perform this task.

About this task

Examining related Common Base Events can give you additional information about how or why the original event failed. The failed event and any related Common Base Events are linked by the same session ID.

To find and view related Common Base Events, use the following procedure.

Procedure

1. From within the administrative console, use the failed event manager to locate the failed event you want to investigate. See "Finding failed events" on page 355 for instructions on how to search for failed events.
2. From the Failed Event Details page for that event, click **Browse Related Common Base Events**.

Results

The Common Base Event Browser opens in a new browser window and lists any Common Base Events related to the original failed event.

Finding business process instances related to a failed event

If a failed event is generated from a business process, the failed event manager provides a link to view that business process instance in Business Process Choreographer Explorer.

Before you begin

You must be logged in as administrator or operator to perform this task.

About this task

Examining the business process instance that generated the failed event can give you additional information about how or why the event failed. The business process instance and the failed event are linked by a common session ID.

Note: Not all failed events are generated from a business process instance.

To find and examine a business process instance related to a failed event, use the following procedure.

Procedure

1. From within the administrative console, use the failed event manager to locate the failed event you want to investigate. See "Finding failed events" on page 355 for instructions on how to search for failed events.

2. From the Failed Event Details page for that event, click **Open calling process in Business Process Choreographer Explorer**.

Results

The Business Process Choreographer Explorer opens in a new browser window and displays information about the related process instance.

Deleting failed events

If you do not want to resubmit a failed event, or if you have failed events that have expired, use the failed event manager to delete them from the server. The failed event manager provides three options for deleting failed events.

Before you begin

You must be logged in as administrator or operator to perform this task.

About this task

To delete one or more failed events, use the following procedure.

Procedure

1. Ensure that the failed event manager is open and that you have retrieved a list of the failed events on your system.

2. From the failed event manager's Search Results page, do one of the following:

   • If you want to delete one or more specific failed events, select the check box next to each event and then click **Delete**.

   • If you want to delete only those failed events that have expired, click **Delete expired events**. Note that this deletes only the expired events in the current set of search results.

   • If you want to delete all failed events on the server, click **Clear all on server**.

Troubleshooting the failed event manager

This topic discusses problems that you can encounter while using the failed event manager.

Note: This topic does not discuss how to use the failed event manager to find, modify, resubmit, or delete failed events on the system. For information about managing failed events, see *Managing WebSphere Process Server failed events* in the information center.

Select the problem you are experiencing from the table below:

Problem	Refer to the following
I am having trouble with reduced performance during an advanced search	"Advanced search feature is not optimized" on page 368
I am having trouble entering values in the Search page's **By Date** tab	"Values in the By Date tab automatically change to default if entered incorrectly" on page 368

Problem	Refer to the following
I am having trouble deleting expired events	"Using the Delete Expired Events function appears to suspend the failed event manager"
I am having trouble with failed events not being created	"Failed events are not being created" on page 369

Advanced search feature is not optimized

The failed event manager's advanced search feature is not optimized. Therefore, you may experience reduced performance when using the Advanced search tab with a large set of failed events.

Values in the By Date tab automatically change to default if entered incorrectly

The Search page's **By Date** tab contains two fields: **From Date** and **To Date**. Both fields are required. The values are locale-dependent, and they must be formatted exactly as shown in the example above the field. Any inconsistency in the value's format (for example, including four digits in the year instead of 2, or omitting the time) will cause the failed event manager to issue the following warning and substitute a default value in the field:

```
CWMAN0017E: The date entered could not be parsed correctly:
your_incorrectly_formatted_date. Date: default_date is being used.
```

The default value of the **From Date** field is defined as `January 1, 1970, 00:00:00 GMT`.

Important: The actual default value shown in your failed event manager implementation will vary depending on your locale and time zone. For example, the From Date field defaults to `12/31/69 7:00 PM` for a machine with an en_US locale in the Eastern Standard Time (EST) time zone.
The default value for the **To Date** field is always the current date and time, formatted for your locale and time zone.

To avoid this problem, always enter your dates and times carefully, following the example provided above each field.

Using the Delete Expired Events function appears to suspend the failed event manager

If you use the Delete Expired Events button in situations where there are many failed events in the current search results, or where those events contain a large amount of business data, the failed event manager can appear to be suspended indefinitely.

In this situation, the failed event manager is not actually suspended: it is working through the large data set, and will refresh the results set as soon as the command completes.

Failed events are not being created

If the Recovery subsystem is not creating failed events, go through the following checklist of potential causes:

- Ensure that the wpsFEMgr application is running. If necessary, restart it.
- Ensure that the failed event manager's database has been created, and that the connection has been tested.
- Ensure that the necessary failed event destination has been created on the SCA system bus. There should be one failed event destination for each deployment target.
- Ensure that the Quality of Service (QoS) **Reliability** qualifier has been set to Assured for any Service Component Architecture (SCA) implementation, interface, or partner reference that participates in events you want the Recovery service to handle.

Chapter 12. Troubleshooting WebSphere Process Server administration

Troubleshooting is the process of finding and eliminating the cause of a problem. This group of topics helps you identify and resolve problems that can occur during typical administration tasks.

For information on troubleshooting Business Process Choreographer or Common Event Infrastructure components, see one of the following locations:

- The WebSphere Process Server for Multiplatforms, version 6.1, information center
- The *Business Process Choreographer* PDF
- The *Common Event Infrastructure* PDF

Troubleshooting the failed event manager

This topic discusses problems that you can encounter while using the failed event manager.

Note: This topic does not discuss how to use the failed event manager to find, modify, resubmit, or delete failed events on the system. For information about managing failed events, see *Managing WebSphere Process Server failed events* in the information center.

Select the problem you are experiencing from the table below:

Problem	Refer to the following
I am having trouble with reduced performance during an advanced search	"Advanced search feature is not optimized" on page 368
I am having trouble entering values in the Search page's **By Date** tab	"Values in the By Date tab automatically change to default if entered incorrectly" on page 368
I am having trouble deleting expired events	"Using the Delete Expired Events function appears to suspend the failed event manager" on page 368
I am having trouble with failed events not being created	"Failed events are not being created" on page 369

Advanced search feature is not optimized

The failed event manager's advanced search feature is not optimized. Therefore, you may experience reduced performance when using the Advanced search tab with a large set of failed events.

Values in the By Date tab automatically change to default if entered incorrectly

The Search page's **By Date** tab contains two fields: **From Date** and **To Date**. Both fields are required. The values are locale-dependent, and they must be formatted exactly as shown in the example above the field. Any inconsistency in

the value's format (for example, including four digits in the year instead of 2, or omitting the time) will cause the failed event manager to issue the following warning and substitute a default value in the field:

CWMAN0017E: The date entered could not be parsed correctly: *your_incorrectly_formatted_date*. Date: *default_date* is being used.

The default value of the **From Date** field is defined as January 1, 1970, 00:00:00 GMT.

Important: The actual default value shown in your failed event manager implementation will vary depending on your locale and time zone. For example, the From Date field defaults to 12/31/69 7:00 PM for a machine with an en_US locale in the Eastern Standard Time (EST) time zone.
The default value for the **To Date** field is always the current date and time, formatted for your locale and time zone.

To avoid this problem, always enter your dates and times carefully, following the example provided above each field.

Using the Delete Expired Events function appears to suspend the failed event manager

If you use the Delete Expired Events button in situations where there are many failed events in the current search results, or where those events contain a large amount of business data, the failed event manager can appear to be suspended indefinitely.

In this situation, the failed event manager is not actually suspended: it is working through the large data set, and will refresh the results set as soon as the command completes.

Failed events are not being created

If the Recovery subsystem is not creating failed events, go through the following checklist of potential causes:
- Ensure that the wpsFEMgr application is running. If necessary, restart it.
- Ensure that the failed event manager's database has been created, and that the connection has been tested.
- Ensure that the necessary failed event destination has been created on the SCA system bus. There should be one failed event destination for each deployment target.
- Ensure that the Quality of Service (QoS) **Reliability** qualifier has been set to Assured for any Service Component Architecture (SCA) implementation, interface, or partner reference that participates in events you want the Recovery service to handle.

Troubleshooting Service Component Architecture and WebSphere MQ communications

Communication between Service Component Architecture (SCA) modules and WebSphere MQ queue managers depends on the binding between the imports and exports within the SCA module and the queues in WebSphere MQ servers. Use this information to determine the servers that are not processing WebSphere MQ messages.

Before you begin

This task assumes that you have noticed requests dependant on WebSphere MQ are not being processed and that you have access to the administrative console. You should also either have the ability to make changes to the WebSphere MQ queue manager or be in contact with the WebSphere MQ administrator.

About this task

Service Component Architecture (SCA) modules depend on the bindings between the server and the WebSphere MQ queue manager. Communications between the two entities could keep messages from processing completely. The following steps should help you discover the cause of the disruption and what to do to get the messages processed again.

Procedure

1. Display the SCA module communicating with WebSphere MQ to make sure it is still processing. Navigate to this page using **Applications > SCA Modules**.
2. Display the queue manager to make sure it is still operational. Perform this task at the WebSphere MQ administration console.
3. Display the bindings between the SCA module and the queue manager to make sure the binding is correct. If the binding is incorrect, change the binding. Navigate to this page using **Applications → SCA modules** >*moduleName* > **Imports|Exports** > *importName|exportName* > **Bindings** > *bindingName* [type].
4. Locate any messages that may indicate failed transactions. You will have to investigate system, SCA-specific message areas, WebSphere MQ-specific message areas, the failed event queue and other locations to determine what has failed.

   a. Examine SystemOut.log for any messages that would indicate processing failures.

      If there is an WebSphere MQ error, there will be an MQException linked somewhere in the stack trace with a WebSphere MQ reason code (for example, 2059 is "queue manager unavailable").

   b. Check AMQERR*xx*.LOG and the WebSphere MQ FFDC files to determine the cause of a WebSphere MQ error.

   c. Examine the application queues to determine if there are any unprocessed messages. Make sure you examine both WebSphere MQ and Service Integration Bus (SIB) queues.

   d. Examine the WebSphere MQ dead letter queue and the SIB exception destination.

   e. Examine the failed event queue to determine if there are any messages related to the applications of interest. See Finding failed events for

information about locating the failed events. See "Managing WebSphere Process Server failed events" for information about locating the failed events.

Troubleshooting event sequencing

Refer to the information in this topic if you are experiencing difficulty with event sequencing.

Problems with the event sequencing qualifier

Ensure that your component definition is correct:
- Is the event sequencing qualifier set on the method? Event sequencing validation fails if the qualifier is erroneously set on the interface.
- Is the parameter name valid?
- Is the xpath element valid, and does it correctly resolve to a primitive?
- Is there a single eventSequencing element for the method? Each method supports only one eventSequencing element.
- Is there a single keySpecification element for the method? Each method supports only one keySpecification element.

Deadlocks

Deadlocks occur when an invoked operation with a lock invokes another operation on the same component using the same event sequencing key and group. You can resolve a deadlock by using the esAdmin command to list and release the current lock.

To avoid deadlocks, carefully consider dependencies when implementing event sequencing. Ensure that operations with circular dependencies are in different event sequencing groups.

Deadlocks with a BPEL process

Deadlocks can occur when event sequencing is used with Business Process Execution Language (BPEL) processes. Deadlocks are caused by setting event sequencing qualifiers on operations that correspond to both of the following activities:
- Multiple instantiating receive or pick activities, where the `createInstance` attribute is set to `yes`
- Correlation set specifications with an `initiation` attribute set to `join`

Resolve this type of deadlock by using the esAdmin command to list and release the current lock. To prevent further deadlocks, ensure that these types of dependent operations are put into different event sequencing groups.

Performance issues

If you are experiencing memory problems on the messaging engine server used for event sequencing components, try modifying the runtime event sequencing property maxActiveMessages in the *install_root*/properties/ eventsequencing.properties file.

The maxActiveMessages property defines the number of messages currently locked on a component destination; too many large messages can negatively affect performance and cause memory problems. Note that a value of 0 (zero) means that an unlimited number of messages are allowed. By default, the maxActiveMessages property is set to 100.

To modify the maxActiveMessages property, perform the following steps.
1. Open the eventsequencing.properties file in a text editor.
2. Make the appropriate modifications for your environment.
3. Save and close the file.
4. Stop and restart any applications that are part of the event sequencing component in order for the changes to take effect.

Troubleshooting your deployment environment

When processing appears sluggish or requests fail, use a focused approach to determine the source of the problem in the environment. The approach described is for non-standalone server environments.

Before you begin

You must be logged into the administrative console of the deployment manager to perform this task.

Required security role for this task: When security and role-base authorization are enabled, you must log in as an administrator or operator to perform this task.

About this task

Investigate the state of your deployment environment if you notice any of the following symptoms:
- Unavailable applications
- Sluggish applications
- Stopped applications
- Decreased throughput
- Sluggish performance

Procedure
1. Display the topology layout that describes this deployment environment to determine the status of the topology.
2. Display the topology to determine the state of the various roles in the topology. Note the roles with unexpected states or warning for further investigation.
3. Locate the nodes that are causing the error state for each role.
4. Make sure all nodes are synchronized.

   On the Nodes page in the administrative console select any unsynchronized nodes and click **Synchronize**.
5. Make sure that the messaging engines associated with all the buses are running.

   If they are not running, stop and start the messaging engines.
6. Locate the logs associated with the nodes in error and view the logs for error messages.
7. Take any actions prescribed by the error messages to affect the correction.

8. Correct any errors and restart the affected nodes.

Results

The nodes previously in error start and the status of the topology becomes "running."

What to do next

Restart any affected applications

Troubleshooting the business rules manager

Some of the problems you may encounter using the business rules manager are login errors, login conflicts, and access conflicts.

You can take various steps to troubleshoot these problems.

Resolving login errors

A log in error occurs upon logging in.

About this task

The login error message is as follows:

`Unable to process login. Please check User ID and password and try again.`

Note: Login errors occur only when global security is enabled and either the user ID, password, or both, are incorrect.

To resolve login errors, perform the following steps.

Procedure
1. Click **OK** on the error message to return to the Login page.
2. Enter the valid **User ID** and **Password**.
   - If passwords are case sensitive, make sure that Caps Lock key is not on.
   - Make sure the user ID and password are spelled correctly.
   - Check with the system administrator to be sure that the user ID and password are correct.
3. Click **Login**.

What to do next

If you resolve the login error, you will now be able to log in to the business rules manager. If the error is not resolved, contact your system administrator.

Resolving login conflict errors

A login conflict error occurs when another user with the same user ID is already logged in to the application.

About this task

The login conflict message is as follows:

```
Another user is currently logged in with the same User ID. Select from the
following options:
```

Usually this error occurs when a user closed the browser without logging out.
When this condition occurs, the next attempted login before the session timeout
expires results in a login conflict.

Note: Login conflict errors occur only when global security is enabled.

To resolve login conflict errors, select from the following three options:
*   Return to the Login page.

    Use this option if you want to open the application with a different user ID.
*   Log out the other user with the same user ID.

    Use this option to log out the other user and start a new session.

    Note: Any unpublished local changes made in the other session will be lost.
*   Inherit the context of the other user with the same user ID and log out that user.

    Use this option to continue work already in progress. All unpublished local
    changes in the previous session that have been saved will not be lost. The
    business rules manager will open to the last page displayed in the previous
    session.

Resolving access conflict errors

An access conflict error occurs when a business rule is updated in the data source
by one user at the same time another user is updating the same rule.

Before you begin

This error is reported when you publish your local changes to the repository.

About this task

To correct access conflict errors, perform the following actions:
*   Find the source of the business rule that is causing the error and check if your
    changes on the local machine are still valid. Your change may no longer be
    required after the changes done by another user.
*   If you choose to continue working in the business rules manager, you must
    reload those business rule groups and rule schedules in error from the data
    source as your local changes of business rule groups and rule schedules in error
    are no longer usable. Reload a business rule group or rule schedule page, by
    clicking **Reload** in the Publish and Revert page of the rule for which the error
    was reported. You can still use local changes in other business rule groups and
    rule schedules that are not in error.

Troubleshooting bindings

Various error conditions can occur with bindings that are specific to the type of
binding.

About this task

The manner in which error conditions are handled depends upon the type of
binding concerned.

Troubleshooting JMS and WebSphere MQ JMS bindings

You can diagnose and fix problems with JMS and WebSphere MQ JMS bindings.

Implementation exceptions

About this task

In response to various error conditions, the JMS and MQ JMS import and export implementation can return one of two types of exceptions:

- ServiceBusinessException: this exception is returned if exception specified on the service business interface (WSDL port type or Java interface) occurred.
- ServiceRuntimeException: raised in all other cases. In most cases, the cause exception will contain the original exception. In the J2C case that would be ResourceException and in the JMS case, it would be JMSException.

When an XML schema has a type defined without a global element, the JMS bindings (JMSDataBindingImplXML and JMSDataBindingImplJava) cannot resolve the type to an element.

Schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.ibm.com" xmlns:tns="http://www.ibm.com">
 <!-- global element required but missing -->

<complexType name="Quote">
  <sequence>
   <element name="symbol" type="string"></element>
  <element name="price" type="float"></element>
  </sequence>
 </complexType>
</schema>
```

If you receive one of the following exceptions:

```
com.ibm.websphere.sca.ServiceRuntimeException:
caused by: java.lang.IllegalArgumentException:
{Quote}Quote is not corresponding to a global element.
```

Or

```
[8/25/06 10:20:40:938 PDT] 00000054 FFDC          Z
com.ibm.ws.sca.databinding.impl.DataBindingImplXML
com.ibm.ws.sca.databinding.impl.DataBindingImplXML#002 Exception:

org.eclipse.emf.ecore.xmi.FeatureNotFoundException:
Feature 'Quote' not found. (sca:/dataObject.xml, 2, 126)
```

This might mean you need to define a global element:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.ibm.com" xmlns:tns="http://www.ibm.com">

<element name="Quote" type="tns:Quote"></element> <!-- global element required -->

<complexType name="Quote">
  <sequence>
   <element name="symbol" type="string"></element>
   <element name="price" type="float"></element>
  </sequence>
 </complexType>

</schema>
```

JMS-based SCA messages not appearing in the failed event manager

About this task

If SCA messages originated through a JMS interaction fail, you would expect to find these messages in the failed event manager. If such messages are not appearing in the failed event manager, ensure that the underlying SIB destination of the JMS Destination has a maximum failed deliveries value greater than 1. Setting this value to 2 or more enables interaction with the failed event manager during SCA invocations for the JMS bindings.

WebSphere MQ JMS-based SCA messages not appearing in the failed event manager

About this task

If SCA messages originated through a WebSphere MQ JMS interaction fail, you would expect to find these messages in the failed event manager. If such messages are not appearing in the failed event manager, ensure that the value of the maximum retries property on the underlying listener port is equal to or greater than 1. Setting this value to 1 or more enables interaction with the failed event manager during SCA invocations for the MQ JMS bindings.

Faults

About this task

The argument that is expected to be passed to the JMSDataBinding and JMSObjectBinding depends on the interface operation and the input, output and fault types.

For faults, the outDataBindingType specified on the method binding is used. If none is specified, the binding level dataBindingType is used for all serialization and deserialization.

If the fault type is simple, a string is set on the JMS DataBinding representing the fault message. In addition IsBusinessException is set to true.

If the fault type is a data object, then a data object is set on the JMS DataBinding which represents the fault message. This scenario requires the use of JMSDataBinding.

Messages containing faults are handled by the JMS data bindings. A Boolean header property IsBusinessException is intercepted by the data binding. If the value is true, the data binding informs the runtime that the payload contains fault data.

If you are working with a custom data binding, you need to take the following steps to handle faults correctly. Default implementations handle faults without user intervention.

Procedure

1. For JMS exports, use the setBusinessException (boolean isBusinessException) method on the JMSDataBinding interface to indicate that data object or object specified on the data binding is a fault object and the message created by the binding should be constructed accordingly. The Data Binding is then responsible to specify the isBusinessException appropriately.

2. For JMS imports use the isBusinessException() method on the JMSDataBinding interface to indicate whether the message contains a fault.

The data binding gets the value of the header property indicating a fault defined in the payload. After the runtime passes the JMS message to the data binding, it invokes isBusinessException() on the data binding. If returned value is false, the message is processed normally, otherwise, the ServiceBusinessException is returned to the caller. The data object or object produced by the binding is set on a ServiceBusinessException and it is returned to the caller.

Misusage scenarios: comparison with WebSphere MQ bindings
About this task

The WebSphere MQ JMS binding is designed to interoperate with JMS applications deployed against WebSphere MQ, which exposes messages according to the JMS message model. The WebSphere MQ import and export, however, are principally designed to interoperate with native WebSphere MQ applications, and expose the full content of the WebSphere MQ message body to mediations.

The following scenarios should be built using the WebSphere MQ JMS binding, not the WebSphere MQ binding:

- Invoking a JMS message-driven bean (MDB) from an SCA module, where the MDB is deployed against the WebSphere MQ JMS provider. This would be implemented using a WebSphere MQ JMS import.
- Allowing the SCA module to be called from a J2EE component servlet or EJB via JMS. This would be implemented using a WebSphere MQ JMS export.
- Mediating the contents of a JMS MapMessage, in transit across WebSphere MQ. This would use WebSphere MQ JMS export and import in conjunction with the appropriate data binding.

There are situations in which the WebSphere MQ binding and WebSphere MQ JMS binding might be expected to interoperate. In particular, when you are bridging between J2EE and non-J2EE WebSphere MQ applications, use a WebSphere MQ export and WebSphere MQ JMS import (or vice versa) in conjunction with appropriate data bindings, mediation module, or both.

Troubleshooting generic JMS bindings

Certain failure conditions can occur with a generic JMS binding.

About this task

Various error conditions can occur with generic JMS bindings.

Troubleshooting generic JMS exceptions
In response to various error conditions, the generic JMS import and export implementation can return an exception.

About this task

In response to various error conditions, the generic JMS import and export implementation can return one of two types of exceptions:

- ServiceBusinessException – this exception is returned if the exception specified on the service business interface (WSDL port type or Java interface) occurred.

- ServiceRuntimeException – raised in all other cases. In most cases, the *cause* exception will contain the original exception. In the case of JMS, it would be JMSException.

Troubleshooting generic JMS message expiry

A request message by the JMS provider is subject to expiration.

About this task

Request expiry refers to the expiration of a request message by the JMS provider when the JMSExpiration time on the request message is reached. As with other JMS bindings, the generic JMS binding handles the request expiry by setting expiration on the callback message placed by the import to be the same as for the outgoing request. Notification of the expiration of the callback message will indicate that the request message has expired and the client should be notified by means of a business exception.

If the callback destination is moved to the third-party provider, however, this type of request expiry is not supported.

Response expiry refers to the expiration of a response message by the JMS provider when the JMSExpiration time on the response message is reached.

Response expiry for the generic JMS binding is not supported, because the exact expiry behavior of a third-party JMS provider is not defined. You can, however, check that the response is not expired if and when it is received.

For outbound request messages, the JMSExpiration value will be calculated from the time waited and from the requestExpiration values carried in the asyncHeader, if set.

Troubleshooting generic JMS connection factory errors

When you define certain types of Connection Factory in your generic JMS provider, you may receive an error message when you try to start an application. You can modify the external Connection Factory to avoid this problem.

About this task

When launching an application you may receive the following error message: "MDB Listener Port JMSConnectionFactory type does not match JMSDestination type"

This problem can arise when you are defining external connection factories. Specifically the exception can be thrown when you create a JMS 1.0.2 Topic Connection Factory, instead of a JMS 1.1 (unified) Connection Factory (i.e., one that is able to support both point-to-point and publish/subscribe communication).

To resolve this issue take the following steps:

Procedure

1. Access the generic JMS provider that you are using.
2. Replace the JMS 1.0.2 Topic Connection Factory that you defined with a JMS 1.1 (unified) Connection Factory.

Results

When you launch the application with the newly defined JMS 1.1 Connection Factory you should no longer receive an error message.

Troubleshooting WebSphere MQ bindings

You can diagnose and fix faults and failure conditions that occur with WebSphere MQ bindings.

About this task

The primary failure conditions of WebSphere MQ bindings are determined by transactional semantics, by WebSphere MQ configuration, or by reference to existing behavior in other components. The primary failure conditions include:

- Failure to connect to WebSphere MQ queue manager or queue: a failure to connect to WebSphere MQ, to receive messages, will result in the MDB ListenerPort failing to start. This condition will be logged in the WebSphere Application Server log. Persistent messages will remain on the WebSphere MQ queue until they are successfully retrieved (or expired by WebSphere MQ). A failure to connect to WebSphere MQ, to send outbound messages, will cause rollback of the transaction controlling the send.
- Failure to parse inbound message, or to construct outbound message: a failure in the data binding will cause rollback of the transaction controlling the work.
- Failure to send outbound message: a failure to send a message will cause rollback of the relevant transaction
- Multiple or unexpected response messages: the import expects only one response message for each request message; when a response has arrived, the record is deleted. If response messages arrive unexpectedly, they will be discarded as with the JMS import.

Misusage scenarios: comparison with WebSphere MQ JMS bindings
About this task

The WebSphere MQ import and export are principally designed to interoperate with native WebSphere MQ applications, and expose the full content of the WebSphere MQ message body to mediations. The WebSphere MQ JMS binding, however, is designed to interoperate with JMS applications deployed against WebSphere MQ, which exposes messages according to the JMS message model.

The following scenarios should be built using the WebSphere MQ JMS binding, not the WebSphere MQ binding:

- Invoking a JMS message-driven bean (MDB) from an SCA module, where the MDB is deployed against the WebSphere MQ JMS provider. This would be implemented using a WebSphere MQ JMS import.
- Allowing the SCA module to be called from a J2EE component servlet or EJB via JMS. This would be implemented using a WebSphere MQ JMS export.
- Mediating the contents of a JMS MapMessage, in transit across WebSphere MQ. This would use WebSphere MQ JMS export and import in conjunction with the appropriate data binding.

There are situations in which the WebSphere MQ binding and WebSphere MQ JMS binding might be expected to interoperate. In particular, when you are bridging between J2EE and non-J2EE WebSphere MQ applications, use a WebSphere MQ export and WebSphere MQ JMS import (or vice versa) in conjunction with appropriate data bindings and/or mediation module.

If WebSphere MQ cannot deliver a message to its intended destination, usually due to configuration errors, it will send messages instead to a nominated dead-letter queue. In doing so it adds a dead-letter header to the start of the message body; this header contains failure reasons, the original destination, and other information.

Profile-specific log files

There are log files detailing the characteristics and runtime activities of individual profiles. These log files are located within the profile directory for each profile.

There are a number of log files that are created for each profile. Some of these logs describe the parameters used for the creation of the profile. These types of log files generally remain unchanged once the profile is fully configured. Other profile-specific logs are continually updated to capture error, warning, and information messages emitted during runtime. Some of these log files are also used to capture a Common Base Event (that may include business object data) that is selected for monitoring.

The table below specifies the different types of profile-specific log files and the locations where you can find them within the product. Within the table, the variable *install_root* represents the installation directory of WebSphere Process Server. The variable *profile_root* represents the root location of a profile.

`i5/OS` **On i5/OS platforms:** The variable *user_data_root* represents the default user data directory.

For more information see Default installation directories for the product, profiles, and tools.

Table 36. Profile-specific log files updated during runtime

Log	Contents

First failure data capture (ffdc) log and exception files (common to all profile types) are found in these directories:
 	
 • `Linux` `UNIX` **On Linux and UNIX®** **platforms:** *profile_root*/logs/ffdc
 	
 • `Windows` **On Windows platforms:** *profile_root*\logs\ffdc
 	
 • `i5/OS` **On i5/OS platforms:** *profile_root*/logs/ffdc
	Contains the ffdc log and exception files for individual profiles. There are two types of ffdc logs: a single log file with a compilation of all the errors encountered during the profile runtime, and numerous text files with details such as stack traces and other information. The naming conventions for the different types of profiles are given for both files, as follows:
	
	 • **Deployment manager profile:**
	   – Log file — *deployment_manager_name*_exception.log.
	   – Text files — *deployment_manager_name_hex_id_date_time*.txt.
	
	 • **Custom profile:**
	 •
	   – Log file(s) — *node_agent_name*_exception.log and *server_name*_exception.log .
	   – Text files — *node_agent_name*(or)*server_name_hex_id_date_time*.txt.
	
	 • **Stand-alone profile:**
	   – Log file — *server_name*_exception.log.
	   – Text files — *server_name_hex_id_date_time*.txt.

Table 36. Profile-specific log files updated during runtime (continued)

Log	Contents
Deployment manager logs (deployment manager profiles only) are found in these directories: • `Linux` `UNIX` **On Linux and UNIX platforms:** *profile_root*/logs/*deployment_manager_name* • `Windows` **On Windows platforms:** *profile_root*\logs*deployment_manager_name* • `i5/OS` **On i5/OS platforms:** *profile_root*/logs/ *deployment_manager_name*	You will work primarily with four log files in this directory: • startServer.log — Contains the system parameters detected on the system and the messages emitted by the deployment manager during the start process • stopServer.log — Contains the system parameters detected on the system and the messages emitted when the deployment manager is shut down. • SystemErr.log — Contains error and exception messages generated by the deployment manager during runtime. Continually updated while server is running. • SystemOut.log — Contains all messages, including error, warning, and information messages generated by the deployment manager during runtime. Continually updated while server is running.
Node agent logs (custom profiles only) are found in these directories: • `Linux` `UNIX` **On Linux and UNIX platforms:** *profile_root*/logs/*node_agent_name* • `Windows` **On Windows platforms:** *profile_root*\logs*node_agent_name* • `i5/OS` **On i5/OS platforms:** *profile_root*/logs/ *node_agent_name*	You will work primarily with four log files in this directory: • startServer.log — Contains the system parameters detected on the system and the messages emitted by the node agent during the start process • stopServer.log — Contains the system parameters detected on the system and the messages emitted when the node agent is shut down. • SystemErr.log — Contains error and exception messages generated by the node agent during runtime. Continually updated while node agent is running. • SystemOut.log — Contains all messages, including error, warning, and information messages generated by the node agent during runtime. Continually updated while the node agent is running.

Table 36. Profile-specific log files updated during runtime (continued)

Log	Contents

Server logs (custom and stand-alone profiles only) are found in these directories:
 	
• `Linux` `UNIX` **On Linux and UNIX platforms:** *profile_root*/logs/*server_name*
 	
• `Windows` **On Windows platforms:** *profile_root*\logs*server_name*
 	
• `i5/OS` **On i5/OS platforms:** *profile_root*/logs/*server_name*
	You will work primarily with four log files in this directory:
	
	• startServer.log —
	
	  Contains the system parameters detected on the system and the messages emitted by the server during the start process
	
	• stopServer.log —
	
	  Contains the system parameters detected on the system and the messages emitted when the server is shut down.
	
	• SystemErr.log —
	
	  Contains error and exception messages generated by the server during runtime. Continually updated while server is running.
	
	• SystemOut.log —
	
	  Contains all messages, including error, warning, and information messages generated by the server during runtime. Also contains any events being monitoring that are emitted from the Common Event Infrastructure (CEI), in Common Base Event format. These events may also include the level of business object data (FINE, FINER, or FINEST) that is specified for the monitor. Continually updated while the server is running.

Node federation log files are found in these directories (only applies to non-deployment manager profiles):
 	
• `Linux` `UNIX` **On Linux and UNIX platforms:** *profile_root*/logs
 	
• `Windows` **On Windows platforms:** *profile_root*\logs
 	
• `i5/OS` **On i5/OS platforms:** *profile_root*/logs
	Two log files are generated when you attempt to federate a custom, augmented, or stand-alone profile to a deployment manager:
	
	• addNode.log —
	
	  contains the pertinent server environment information and messages generated when you attempt to federate the profile.
	
	• isFederated.log —
	
	  lists the commands used by the deployment manager to federate the profile.

The location of the Integrated Solutions Console application deployment log file is listed here (only for deployment manager and stand-alone profiles): • `Linux` `UNIX` **On Linux and UNIX platforms:** *profile_root*/logs/iscinstall.log • `Windows` **On Windows platforms:** *profile_root*\logs\iscinstall.log • `i5/OS` **On i5/OS platforms:** *profile_root*/logs//iscinstall.log	The iscinstall.log file contains information regarding the deployment of the administration console application in a deployment manager or stand-alone profile.

Table 36. Profile-specific log files updated during runtime (continued)

Log	Contents
The location of the Installation Verification Tool log file is listed here (only for deployment manager and stand-alone profiles): • Linux UNIX **On Linux and UNIX platforms:** *profile_root*/logs/ivtClient.log • Windows **On Windows platforms:** *profile_root*\logs\ivtClient.log • i5/OS **On i5/OS platforms:** *profile_root*/logs//ivtClient.log	This log file contains the output generated by the Installation Verification Tool. You can start this program from the First Steps console after you create a deployment manager or stand-alone profile. The log contains basic configuration information and the messages that are displayed when you run the tool.
The location of the log file detailing the commands generated for a profile creation is listed here: • Linux UNIX **On Linux and UNIX platforms:** *profile_root*/logs/updateserverpolicy.log • Windows **On Windows platforms:** *profile_root*\logs\updateserverpolicy.log • i5/OS **On i5/OS platforms:** *profile_root*/logs/updateserverpolicy.log	This file contains the sequence of commands used by the product to set server environment variables and create a profile. All profile types will contain this file.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
577 Airport Blvd., Suite 800
Burlingame, CA 94010
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows: (c) (your company name) (year). Portions of

this code are derived from IBM Corp. Sample Programs. (c) Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks and service marks

IBM, the IBM logo, DB2, i5/OS, WebSphere, and z/OS are registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft and Windows are registered trademark of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

This product includes software developed by the Eclipse Project (http://www.eclipse.org).

IBM WebSphere Process Server for Multiplatforms, Version 6.1.0

IBM ®

Printed in USA