WebSphere® Process Server for z/OS

IBM

Version 6.0.2

Administering WebSphere Process Server

Note

Before using this information, be sure to read the general information in “Notices” on page 277.

30 March 2007

This edition applies to version 6, release 0, modification 2 of WebSphere Process Server for z/OS (product number 5655-N53) and to all subsequent releases and modifications until otherwise indicated in new editions.

To send us your comments about this document, email doc-comments@us.ibm.com. We look forward to hearing from you.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Administering WebSphere Process Server

The topics in this section describe how to administer the WebSphere® Process Server runtime environment, including the applications and resources deployed in the environment.

WebSphere Process Server documentation (in PDF format)

Additional resources for learning about WebSphere Process Server administration are available on the IBM Education Assistant web site under the **WebSphere Business Process Management** module.

For information about administering business processes and human tasks, refer to one of the following locations in the WebSphere Process Server for z/OS version 6.0.2 information center

- **Administering WebSphere Process Server > Administering Business Process Choreographer**
- **Administering WebSphere Process Server > Administering applications and application services > Administering business processes and human tasks**
- The *Business Process Choreographer* PDF file

For information about administering the Common Event Infrastructure (CEI), refer to one of the following locations in the WebSphere Process Server for z/OS version 6.0.2 information center:

- **Administering WebSphere Process Server > Administering the Common Event Infrastructure**
- The *Common Event Infrastructure* PDF file

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.

The administrative interfaces

WebSphere Process Server offers several interfaces for administering the runtime environment:

The administrative console

The administrative console is a browser-based interface that enables users to monitor, update, stop, and start a wide variety of applications, services, and resources. The administrative console can also be used to work with relationships and to locate and resolve failed WebSphere Process Server events.

The administrative console extends to provide administration capabilities for WebSphere Application Server, as well as other customer-defined products.

For more information, see "The administrative console for WebSphere Process Server" on page 3.

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 can be installed after the initial installation of the server. You can use the business rules manager to open a copy of a business rule from the repository, browse and edit a business rule, and publish a business rule to the repository.

Business Process Choreographer Explorer

Business Process Choreographer Explorer is a stand-alone Web application that provides a basic set of administration functions for managing business process and human tasks. You can view information about process templates, process instances, task instances, and their associated objects. You can also act on these objects; for example, you can start new process instances, repair and restart failed activities, manage work items, and delete completed process instances and task instances.

Business Process Choreographer Observer

Business Process Choreographer Observer is a Web application that creates reports about events that occur during the execution of business processes and human tasks. You can then use these reports to evaluate the effectiveness and reliability of your processes and activities.

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 > Scripting interface** in this information center for details about the scripting tools.

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 > API documentation** in this information center for details about the command-line tools.

Administrative programs

A set of Java™ classes and methods under the Java Management Extensions (JMX) specification 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 > Programming interfaces** in this information center for details about the programming interfaces.

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 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 process 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, which include the Business Flow Manager, Human Task Manager, Staff Service, and Service Reference Service.
resources-bpc.xml	Defines operating environment resources for a Business Process Choreographer container, including configuration information for Staff Plugin Providers. This file is present at the cell, node, and server scopes.
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, Events 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 for z/OS Information Center for complete information about server configuration files.

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. It can be used to administer at a cell, node, or server scope, and is available from both stand-alone process servers and deployment managers that manage all servers in a cell in a networked environment.

Note: The WebSphere Process Server administrative console is part of the larger WebSphere Application Server administrative console. 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 for z/OS 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
- 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 13 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:

- **Guided Activities > Configure your Network Deployment environment –** Provides a guided activity for configuring and administering nodes, databases, clusters, and servers in a Network 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
  - – Application Scheduler
  - – Business rules
  - – Events service
  - – Extended Messaging Service
  - – Service Component Architecture configuration
  - – Selectors
  - – Staff service
  - – WebSphere Business Integration Adapter Service
  - – Web service reference service
- **Resources –** Provides access to the following:
  - – WebSphere Business Integration Adapters
  - – Common Event Infrastructure Provider
  - – Staff plug-in provider
  - – Extended Messaging Provider
- **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, application integration or enterprise service bus administration).

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.

Related concepts

"Getting started with the administrative console"

"Administrative console task filters overview" on page 9
Task filtering allows you to customize your view of the WebSphere administrative console by removing some of the pages that you do not use.

"Administrative console pages" on page 13

"Administrative console buttons" on page 14

Getting started with the administrative console

The following list of tasks can help you get started using the administrative console to manage and administer WebSphere Process Server resources.

- **Start the server for the administrative console.**

  In a stand-alone server environment, the startServer command reads the configuration file for the specified application server and starts the server. If you have a group of servers (a cell) managed by a deployment manager, use the startManager command instead.

  See the WebSphere Application Server for z/OS Information Center for details on using startServer and startManager.
- **Start the administrative console.**

  See "Starting and stopping the administrative console" on page 7 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 8.
- **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 for z/OS 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 9.
- **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 for z/OS 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.

To start the console and log in, use the following 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 `CWWSR0221I: 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:

      - *install_root*/profiles/*profile_name*/config/cells/*cell_name*/nodes/*node_name*/serverindex.xml
      - *install_root*/profiles/*profile_name*/config/cells/*cell_name*/virtualhosts.xml

      Change all occurrences of the port selected during profile creation (by default, 6090) to the port for the console. Alternatively, shut down the other application that uses the conflicting port before starting the administrative console.

   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.

   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.

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.

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 9 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:

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.

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.

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.

Administrative console task filters overview

Task filtering allows you to customize your view of the WebSphere administrative console by removing some of the pages that you do not use.

Task filtering allows you to group tasks in the WebSphere administrative console. You have three choices for task filtering:

All
Displays all administrative console pages. An administrator uses this option to manage all parts of WebSphere Process Server and the underlying WebSphere Application Server.

Application Integration
Displays pages focused on the following tasks:

- Configuring service integration buses.
- Configuring message engines and network topologies used to support deployment of mediation modules, business modules, and service applications.
- Creating resources (for example, JMS connection factories and Common Event Infrastructure profiles) needed by deployed service application, business modules, and mediation modules.
- Monitoring the operation of mediation modules and service applications.

An administrator uses this option to deploy and administer business and mediation modules. This option assumes the required bus and server exists.

Server and Bus

Displays pages focused on the following tasks:

- Defining service integration buses, servers, server clusters, messaging engines, and network topologies needed to support the deployment of mediation modules, business modules and service applications.
- Enabling and disabling infrastructure services.
- Installing applications, business modules, and mediation modules.
- Creating resources (for example, JMS connection factories and Common Event Infrastructure profiles) needed by deployed service application and mediation modules
- Monitoring the operation of the bus and server environment.

An administrator uses this option to manage the bus and server environment needed to support the deployment of service applications and mediation modules. Includes defining the network and bus topology, defining appropriate resources, monitoring the runtime system, and troubleshooting any runtime errors. Use this option to create the technical environment for the applications.

When you first open WebSphere administrative console, by default “All” is selected.

Using the task filter:

Customize your WebSphere administrative console by selecting a task filter.

You can customize your WebSphere administrative console by selecting a task filter to remove some of the pages in the administrative console. If you want to see a specific server configuration, for example, you can use the task filter to show only the pages related to the server. To apply a task filter, use the following procedure.

1. Click the **Welcome** link in the navigation tree.
2. Choose a task filter from the list of available task filters in the **Task filtering selector** located near the bottom of the page.
3. Click **Apply**.

After you click **Apply**, the task filter takes effect immediately. You do not have to shut down and restart the administrative console.

Related concepts

Task filter Overview
Task filtering allows you to customize your view of the WebSphere administrative console by removing some of the pages that you do not use.

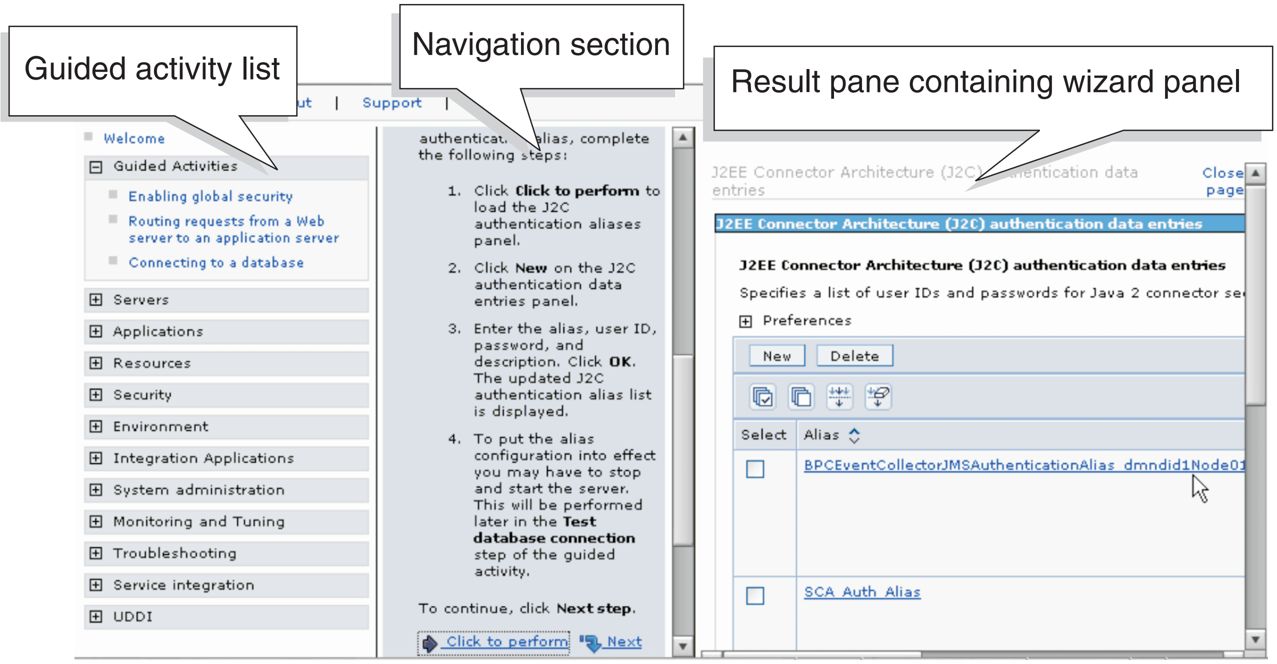
Administrative console guided activities and wizards

Guided activities are steps that help you perform specific tasks. To guide you through a specific portion of the task, a guided activity might start a wizard.

Configuring clusters or servers to support certain functions, such as network deployment, is intricate and can be difficult for those who are not familiar with the tasks. IBM® provides guided activities that explain the task and can start wizards that guide you through the configuration. Use guided activities until you become familiar with performing these configurations.

What is a guided activity?

A guided activity describes a logical series of tasks you must complete to achieve a specific goal. Each step in the guided activity corresponds to a task required to perform a part of the configuration, for example, configuring a cluster. To further simplify the configuration, some steps might offer a wizard to guide you through the configuration panels you must complete for that step. Figure 1 shows the administrative console displaying a guided activity.



authenticat alias, complete the following steps:

1. Click **Click to perform** to load the J2C authentication aliases panel.
2. Click **New** on the J2C authentication data entries panel.
3. Enter the alias, user ID, password, and description. Click **OK**. The updated J2C authentication alias list is displayed.
4. To put the alias configuration into effect you may have to stop and start the server. This will be performed later in the **Test database connection** step of the guided activity.

To continue, click **Next step**.

J2EE Connector Architecture (J2C) entication data entries

J2EE Connector Architecture (J2C) authentication data entries

J2EE Connector Architecture (J2C) authentication data entries

Specifies a list of user IDs and passwords for Java 2 connector se

Select	Alias
□	BPCEventCollectorJMSAuthenticationAlias_dmndid1Node01
□	SCA_Auth_Alias

Figure 1. A guided activity

What is a wizard?

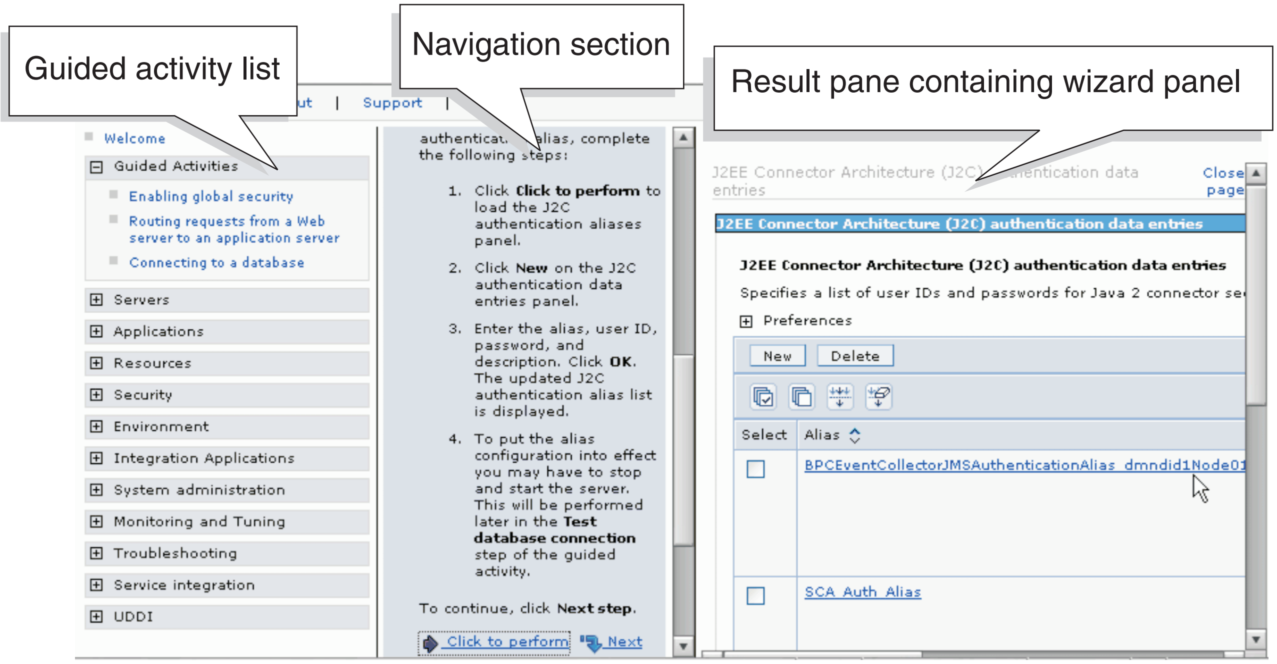
A wizard guides you through the configuration panels you must complete to achieve a specific configuration goal. The number of panels presented varies with the complexity of the task. After you save the changes, you can proceed to the next step in the guided activity. A guided activity starts a wizard when you select **Click to perform step** in the description.

Using guided activities and wizards:

Guided activities and wizards help you perform complex configuration tasks for servers and clusters. By using these activities, you become more familiar with the steps that comprise specific configuration tasks and help prevent configuration errors. Use them until you become familiar with the tasks and can perform them without guidance.

You must be at the administrative console and be logged in with a userid that has the correct authority to perform the tasks.

When you are unfamiliar with specific configuration tasks or perform them rarely, you might find it beneficial to use the available guided activities. These panels not only give you background information about the task you are performing, but also provide step-by-step directions to help you complete the task. Portions of the guided activity might launch wizards that perform the described task.



authentica alias, complete the following steps:

1. Click **Click to perform** to load the J2C authentication aliases panel.
2. Click **New** on the J2C authentication data entries panel.
3. Enter the alias, user ID, password, and description. Click **OK**. The updated J2C authentication alias list is displayed.
4. To put the alias configuration into effect you may have to stop and start the server. This will be performed later in the **Test database connection** step of the guided activity.

To continue, click **Next step**.

J2EE Connector Architecture (J2C) authentication data entries

Specifies a list of user IDs and passwords for Java 2 connector se

Figure 2. A guided activity

Tip: When using a guided activity for the first time, read through all of the steps before performing any of them. Reading through the activity allows you to get a feel for the things that you will do and the interactions between the different steps. During this initial read, do not launch any wizards by selecting **Click to perform this step**.

1. Expand the **Guided Activities** section of the administrative console.

   Click the plus sign (+) next to **Guided Activities**.
2. From the list of tasks displayed, choose the task to perform.

   Click on an item in the list. The console now displays two additional sections: a navigation section and a result pane. (See A guided activity.) The navigation section displays an introduction to the task and the result pane displays background information about the task.
3. To begin the guided activity, select **Start**.

   The navigation pane displays the next step of the task.
4. Read the description, perform the actions described, if needed, and select **Click to perform this step**, if present, to launch the associated configuration wizard.

   The result pane displays the first panel of the configuration task described in the navigation pane.
5. **Optional:** Adjust the panes so you can access the configuration panel easily.
   a. Click on the line separating the panes.
   b. Drag the line to the right or left to adjust the width of the pane.
6. Perform the actions required by the wizard.

   After completing the wizard that helps you achieve the step you are brought back to the step in the guided activity.

Tip: Read over the instructions again to verify that you have achieved the goal of the step and to reinforce the place these actions have in the overall flow of the activity.

7. Repeat steps 4 on page 12 through 6 on page 12 until you complete the task.

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” on page 14. 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 and wizards" on page 10.

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 9.
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 for z/OS documentation.

Setting up the WebSphere Process Server administration environment roadmap

Specific tasks are needed to ensure that the WebSphere Process Server administration environment is set up correctly.

After properly installing the WebSphere Process Server, you must set up your administrative and server environment.Table 2 lists the tasks involved in setting up this environment.

Table 2. Tasks involved in setting up the WebSphere Process Server administration environment

Task	Described in
Configure the product	Configuring the product after installation
Configure ports	Avoiding port conflicts and Port number settings
Set up the administrative architecture	Setting up the administrative architecture
Configure cell-wide settings	Configuring cell-wide settings
Modify the server configuration files	Working with server configuration files
Administer servers and the network deployment environment	Administering process servers

Table 2. Tasks involved in setting up the WebSphere Process Server administration environment (continued)

Task	Described in
Configure WebSphere Process Server resources	Configuring WebSphere Process Server resources
Deploy and manage applications	Deploying and administering applications and Overview of preparing and installing modules
Administer WebSphere Process Server services	Administering business processes and human tasks, Business rules, Administering enterprise applications, and Configuring application scheduler

Setting up the administrative architecture

How to set up the services, deployment managers, and other portions of the environment that control the workflow through the servers that comprise your processing environment.

This task assumes that you have already:

1. Unloaded the product from the tape, as described in *Loading the product code from the installation media onto z/OS* in the WebSphere Process Server for z/OS installation and configuration PDF.
2. Created the product definitions by running the installation script, as described in *Running the installation script* in the WebSphere Process Server for z/OS installation and configuration PDF.
3. Configured the software, as describe in *Configuring the software* in the WebSphere Process Server for z/OS installation and configuration PDF.
4. Configured ports as describe in *Avoiding port conflicts* and *Port number settings* in the WebSphere Process Server for z/OS installation and configuration PDF.

You may want to review the tasks described in Setting up administrative architecture in the WebSphere Application Server for z/OS information center before beginning.

After you install and set up WebSphere Process Server, you need to monitor and control incorporated nodes and the resources on those nodes by using the administrative console.

1. Configure administrative services using Administration service settings
2. Install an instance of the database product
3. Create a WebSphere Process Server database in that database instance.
4. Create the WebSphere Process Server for z/OS database. See for information on creating the databases that will hold the hold the database tables. See *Create the databases and storage groups* in the *Configuring the software* section of the WebSphere Process Server for z/OS Installation and configuration guide.

   You can use **DBUtility.sh** to create the databases and database tables. **DBUtility.sh** automatically sets up the tables for the supported database types. The DBUtility.sh is available in the augmented WebSphere Application Server for z/OS® configuration HFS. For example, */WebSphere/V6R0/AppServer/*bin/DBUtility.sh.

Through response file parameters, you decide if you want to create the database table definitions as part of the product configuration, or, you can decide to create the definitions later, by running the DDL and SQL manually. The name of the response file property that determines when to run the DDL and SQL is **dbDelayConfig**. For more information on how to set this property, see the *Sample response files* topic in the *Configuring the software* section of the WebSphere Process Server for z/OS Installation and configuration guide.

For information on Creating the database table definitions manually, after running the product configuration script, see "Configuring the database manually" in the WebSphere Process Server for z/OS Installation and configuration guide.

5. Configure cells
6. Configure deployment managers
7. On the administrative console select **Environment > WebSphere Variables > (the scope of node) > *DatabaseName*_JDBC_DRIVER_PATH** to define where the WebSphere Process Server database resides

   Using one of the following variables to define the database:
   - For DB2®: DB2UNIVERSAL_JDBC_DRIVER_PATH
   - For Cloudscape: not supported for distributed environments

   For more information, see Configuring WebSphere variables.
8. Manage nodes
9. Manage node agents
10. Manage node groups
11. Configure remote file services

Your WebSphere Process Server environment is now ready to process work.

You can now install applications as described in *Overview of preparing and installing modules* in the WebSphere Process Server Developing and deploying modules PDF..

Configuring cell-wide settings

To assist in handling requests among Web applications, Web containers, and application servers, you can configure cell-wide settings for virtual hosts and variables.

This task assumes you have already set up the administrative architecture, as described in *Setting up the administrative architecture*.

When you are establishing a network deployment environment or are planning to install applications in a cluster you also have to configure specific settings across the cell.

- Configure virtual hosts
- Configure variables

  The variable that defines the WebSphere Process Server database must be configured with a scope of node or cell. Use one of the following for the variable *DatabaseName*_JDBC_DRIVER_PATH:
  - For DB2: DB2UNIVERSAL_JDBC_DRIVER_PATH
  - For Cloudscape™: not supported for distributed environments

Administering WebSphere Process Server servers roadmap

WebSphere Process Server server configuration provides settings that control how a server provides services for running enterprise applications and their components. Administrators can create and configure servers in an existing application server environment.

WebSphere Process Server administrators can configure one or more servers and perform tasks such as the following:

Table 3. Sever administration tasks

Task	**Described in**
Create servers	See *Creating servers* in this PDF.
Manage application servers	See *Managing application servers roadmap* in this PDF.
Configure transport chains	Configuring transport chains
Develop custom services	Developing custom services
Define processes for the application server	Defining application server processes
Configure the Java Virtual Machine	Configuring the JVM

Creating servers

For WebSphere Process Server, you can create servers using either the wsadmin createApplicationServer command or the **Create New Application Server** wizard in the administrative console.

This task assumes that you have installed and configured the product.

Create servers to provide a processing environment for WebSphere Process Server applications.

Choose the method and complete the steps for that method

Method	Steps to complete
Create New Application Server wizard	1. In the administrative console **Application Servers** page, click **New** to open the **Create New Application Server** page 2. Define your server by doing the following: a. Select a node for the server. b. Type a name for the server. The name must be unique within the node. c. Select the **defaultProcessServer** template to create the server. You can also use an existing application server as a template. The new process server inherits all properties of the template server. 3. Select whether the new server will have unique ports for each HTTP transport. By default, this option is enabled. If you select this option, you might need to update the alias list for the virtual host that you plan to use with this server to contain these new port values. (This includes the host alias that you use for the Business Process Choreographer configuration in step 4. Business Process Choreographer is itself an application and requires host_alias management.) If you clear this option, ensure that the default port values do not conflict with other servers on the same workstation. **Note:** If you create the server using an existing application server as a model, do not select to map applications from the existing server to the new server. By default, this option is disabled. 4. **Optional:** Configure Business Process Choreographer, if the server will run applications that contain business processes or human tasks. See Configuring Business Process Choreographer for more information. 5. Install the SchedulerCalendars application on one or more servers as described in Configuring Business Process Choreographer
wsadmin createApplicationServer command	See the createApplicationServer command name in Commands for the AdminTask object

Managing application servers roadmap

You can use the **Application Servers** panel of the administrative console, or command line tools such as startServer and stopServer to manage servers. The administrative console interface is described.

To view information about an server, use the **Application Servers** panel on the administrative console.

Note: You can upgrade a portion of the nodes in a cell, while leaving others at the older release level. This means that, for a period of time, you may be managing servers that are at the current release and servers that are running the newer release in the same cell. In this mixed environment, there are some restrictions on what you can do with servers that are running the older release level. There are no restrictions on what you can do with the servers that are running on the newer release level. For details, see *Creating servers* in the Administering WebSphere Process Server servers roadmap section of this PDF.

Table 4 lists the tasks to manage an application server.

Table 4. Server management tasks available from the administrative console

Task	How to accomplish
Display the available servers	Click **Servers > Application Servers** in the console navigation tree. The Application servers page lists the servers and the cell and the nodes holding the servers. It also shows the status of each server. The status indicates whether a server is started, stopped, or unavailable.
Display information for a particular server	Click a specific server name under **Name**. This accesses the Application server settings page for that server.
Create a process server	*Creating servers* in the Administering WebSphere Process Server servers roadmap section of this PDF
Start servers	Starting an application server
Monitor server operation	Detecting and handling problems with runtime components
Stop servers	Stopping an application server
Delete servers	1. Click **Servers > Application Servers** in the console navigation tree to access the **Application Servers** page. 2. Select the check box beside a server to delete it. 3. Click **Delete**. 4. Click **OK** to confirm the deletion.

Configuring WebSphere Process Server resources roadmap

After installation, you must configure key resources for your WebSphere Process Server environment, such as application scheduler, business processes, Common Event Infrastructure, business rules, selectors and relationships.

An important step in completing your post-installation tasks for setting up the administrative environment is configuring your key resources. Use Table 5 to locate the information to configure the resources.

Table 5. Steps to configure resources

Resource	How to configure
Application scheduler	See *Configuring application scheduler* in the Administering applications and application services section of this PDF.
Business Process Choreographer	See *Configuring Business Process Choreographer* in the Business Process Choreographer PDF.
Common Event Infrastructure	See *Configuring the Common Event Infrastructure* in the Common Event Infrastructure PDF.
Business rules	See *Business rules manager* in the Overview of business rules and selectors section of this PDF.

Table 5. Steps to configure resources (continued)

Resource	How to configure
Selector components	See *Overview of selector components* in the Overview of business rules and selectors section of this PDF.
Relationship service	*Administering the relationship service* in the Configuring WebSphere Process Server settings roadmap section of this PDF.

Working with server configuration files

You can change the default locations of configuration files, as needed. Server configuration files define the available application servers, their configurations, and their contents.

To work with server configuration files in WebSphere Process Server, follow the instructions in the Working with server configuration files topic in the WebSphere Application Server for z/OS information center.

Installing and managing applications

After product installation, you must complete a number of post-installation tasks concerning the administration of applications and their environments.

After installation, you must administer applications and their environment, including customizing tasks, deploying applications and administering server environments.

- For more information, refer to Administering applications and their environment in the WebSphere Application Server for z/OS, Version 6.0.x information center.
- For more information on installing applications on WebSphere Process Server, refer to Overview of preparing and installing module.
- For information on stopping, starting and modifying applications, refer to Deploying and administering applications in the WebSphere Application Server for z/OS, Version 6.0.x information center.

Administering applications and application services

This section describes how to use the administrative interfaces to administer WebSphere Process Server applications and application services, including business processes and tasks, business rules, and schedules.

Configuring WebSphere Process Server settings roadmap

Lists services within WebSphere Process Server that you configure after installing the software.

Upon successful installation of WebSphere Process Server, you must complete configuration by specifying settings for the various services you are using. Table 6 on page 22 lists the services and the topics that describe how to configure the settings for each service.

Table 6. Services to configure

Service	**Described in**
Application scheduler	See *Configuring the Application Scheduler for a standalone server* in the *Application Scheduler* section of Administering applications and application services.
Events service	Managing Events Service
Extended messaging service	See *Enabling the extended messaging service* in the section *Administering extended messaging resources: Overview*.
WebSphere Business Integration Adapter service	See *Working with WebSphere Adapters* in the section *Administering adapters*.
Relationship service	See *Administering the relationship service*.

Managing Events Service

The Events Service configuration panel lists the properties set to ensure that information about WebSphere Process Server is automatically included in each event passed to the Events Infrastructure.

Start the administrative console. If you are unfamiliar with starting the administrative console or the steps to display the application servers, refer to the related topics on the administrative console. This task is performed at the administrative console.

When active, the events service ensures that any event that passes to the events infrastructure automatically includes information about a server. Use the **Events Service configuration panel** to set the events service configuration to on or off.

To configure the **Events Service Startup property** for an application server, complete the following steps:

1. Select the application server to configure.

   In the navigation panel, click **Servers > Application servers**. Select the application server from the list.

   The server properties display.
2. Display the Events Service configuration panel.

   From the Business Integration properties, select **Events service**.

   The **Events Service** configuration panel displays.
3. Select or deselect **Enable service** at startup field.

   The default value is to enable the service automatically at startup. If this option is deselected and subsequent applications require this service to run, the system administrator must manually start the service. Optionally, the system administrator can restart this service by selecting the option and restarting the server.
4. Verify that the Java Naming and Directory Interface (JNDI) name is correct.

   Review the **Events infrastructure emitter factory JNDI name** field. If you have not generated an alternative profile, it is recommended that you accept the default JNDI name.
5. Select any custom properties, if necessary.

   Click the **Custom properties** link and select any custom properties from the listing.
6. Save your configuration.

Click **Apply** to save your configuration and remain on the **Events Service configuration** panel. Click **OK** to save your configuration and return to the Business Integration panel.

7. Apply your changes to the server.

   Stop and restart the application server to apply your changes.

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.″

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.

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.

Security role required: To perform this task, you must be logged in as a configurator or an administrator. Any WebSphere security role can view this configuration.

Perform this task to set the data source and cache size hint (relationship instance count) properties for the relationship service.

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 **Cache Size Hint (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 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.

Security role required: Any WebSphere security role can view this configuration.

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.

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.

Security role required: Any WebSphere security role can view this configuration.

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**.

Creating a clustered environment

Creating a clustered environment enhances the failover and scaling characteristics of WebSphere Process Server applications. To establish a clustered environment, you and your team will perform the described steps.

There are a number of things you should do before creating a clustered environment:

- Ensure that you have adequate resources to implement clustering successfully.
- Analyze the service applications you will deploy into the clustered environment. Some of the optional steps you perform depend on the needs of the service applications.
- Ensure that the application logic is tolerant of a clustered environment, for example:
  - The application tolerates partitioned queues when you make provisions for orphaned requests or for requests processed out of order. An application failure may create one or both of these situations.
  - There are no system-wide values that you maintain locally.
- Familiarize yourself with network deployment and clustering as described in the WebSphere Application Server for z/OS information center.
- Familiarize yourself with these instructions before performing any of the steps. An overall idea of the steps can help you proceed efficiently.
- Make sure WebSphere Application Server for z/OS has been installed.

  See Installing the product and additional software in the WebSphere Application Server for z/OS information center.
- Configure the WebSphere Application Server for z/OS to support a WebSphere Process Server for z/OS clustered environment.

  This involves the following tasks

  - Creating a Network Deployment cell as described in the WebSphere Application Server for z/OS information center.
  - Creating an empty managed node as described in the WebSphere Application Server for z/OS information center.

  Note: Do not Federate into Deployment manager. This means do not run the job named BBOWMNAN as part of the process for creating the empty managed node.
- Run the WebSphere Process Server for z/OS install script to create the product definitions.

  See *Running the installation script* in the WebSphere Process Server for z/OS Installation and Configuration PDF for a description of this task.
- Create the databases and storage groups.

  WebSphere Process Server for z/OS includes a sample file that you can use to create the database and storage groups.

  See *Creating databases and storage groups* in the WebSphere Process Server for z/OS Installation and Configuration PDF for a description of this task.
- Edit the deployment manager response file.

  The deployment manager response file contains default property settings that you modify to suit your environment. Also, the values that you used when creating the database and storage groups must match their corresponding values in the response file. For more information, see *Working with response files* in the WebSphere Process Server for z/OS Installation and Configuration PDF.
- Run the WebSphere Process Server for z/OS configuration script with the deployment manager response file.

  This results in augmenting the default profile with WebSphere Process Server for z/OS network deployment configuration data.

  Also, as a result of running the WebSphere Process Server for z/OS script, the DDL for the WebSphere Process Server for z/OS databases only are created. The databases for WebSphere ESB are not created.

  See *Creating a stand-alone configuration* in the in the WebSphere Process Server for z/OS Installation and Configuration PDF for a description of this task.
- Run the WebSphere Process Server for z/OS install script for the managed node.

  See *Running the installation script* in the WebSphere Process Server for z/OS Installation and Configuration PDF for a description of this task.
- Run the WebSphere Process Server for z/OS configuration script with the managed node response file.

  This results in augmenting the default profile with WebSphere Process Server for z/OS managed node configuration data.

  See for a description of this task.

  See *Creating an empty managed node configuration* in the WebSphere Process Server for z/OS Installation and Configuration PDF for a description of this task.
- Federate the managed node into the Deployment manager.

  Run the job BBOWMNAN. This is the job you did not run when you created the empty managed node originally.

  See .

  See *Federating an empty managed node to a deployment manager* in the WebSphere Process Server for z/OS Installation and Configuration PDF for a description of this task.

Create a clustered environment when your application requires more capacity and availability than a single server provides. The clustered environment provides these additional benefits:

Workload balancing
By running application images on multiple servers, a cluster balances an application workload across the servers in the cluster.

Processing power for the application
You can add processing power to your application by configuring additional server hardware as cluster members supporting the application.

Application availability
When a server fails the application continues to process work on the other servers in the cluster thereby allowing recovery efforts to proceed without impacting application users.

Maintainability
You can stop a server for planned maintenance without stopping application processing.

Flexibility
You can add or remove capacity as needed by using the administrative console.

Note: A list of related tasks is included at the bottom of this page. The title of any related tasks for a particular step is contained as a parenthetical comment for that step. If you are unfamiliar with the task, see the related topic for additional information.

1. Design the clustered environment
   a. Lay out the topology of the cell. Determine what physical and logical resources are needed for the cell.
   b. Decide whether to use databases or schemas for the various components of the cell.
   c. Decide if there is a need for monitored events and which servers should host the monitored events.
2. Create other databases needed by the cell, such asdatabases and schema for Business Process Execution Language and Common Event Infrastructure:
   a. Create Environment variables Create the following:
      - DB2UNIVERSAL_JDBC_DRIVER_PATH = /db2810/jcc/classes
      - DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH = /db2810/jcc/lib
      - Add JDBC JCC Properties file to DMGR/Server Adjunct and servant Custom Property

        Application servers > server > Process Definition > Servant > JavaVirtualMachine >Custom Properties > New

        db2.jcc.propertiesFile, for example, `/shared/db2810/jcc/properties/DB2JccConfiguration.properties`

        Note: This property will be required for each server configured in the cluster.
3. **Optional:** Create the cluster that handles monitored events, if you need monitored events.
   a. Create the cluster with cluster members using the default WebSphere Process Server template.

   b. Using the cluster you are configuring as the target and the Common Event Infrastructure (CEI) database created in step 2 on page 27, install the CEI application, and install the message-driven bean (MDB) application. (Configuring the Common Event Infrastructure)

4. **Optional:** Create the cluster that handles administration applications, if the application contains business rules or selectors and the application requires modification to the rules or selectors after you deploy the application.

   Note: If you have determined that administration of business rules and selectors will be deployed to clusters that handle other WebSphere Process Server components, you can combine this step with either step 3 on page 27 or step 6.

   a. Create the cluster with cluster members using the default WebSphere Process Server Template.

   b. Use the Advanced Configuration panel to deploy the business rules manager server.

5. Configure messaging for the cluster.

   Note: If you have determined that administration messaging will be handled by clusters that handle other WebSphere Process Server components, you can combine this step with either step 3 on page 27, step 4, or step 6.

   Important: This step must be completed before installing the first service application in the cluster.

   a. Create the cluster using the default WebSphere Process Server template.

   b. Select **Default Destination Location** on the Advanced Configuration panel.

   c. Add the cluster as a member of the Event bus.

      Use the data source you created in step 2 on page 27. If in step 1b on page 27 you decided to use multiple schemas in a single database, select the correct schema when configuring the messaging engine data source.

   d. Add the cluster as a bus member of the Business Process Choreographer bus.

      Use the data source you created in step 2 on page 27. If in step1b on page 27 you decided to use multiple schemas in a single database, select the correct schema when configuring the messaging engine data source.

6. Create the cluster for service applications.

   a. Create the cluster with cluster members using the default WebSphere Process Server template

   b. Use the Advanced Configuration panel to select the messaging cluster.

      When combining this step with step 5, select the default messaging cluster as stated in that step. Otherwise, select the correct option and choose the cluster you created in 5.

   c. **Optional:** Configure Business Process Choreographer support using the databases or schemas created in step 2 on page 27. (Configuring Business Process Choreographer)

      Perform this step when an application contains Business Process Execution Language (BPEL) or business state machine components.

   d. **Optional:** Use the Advanced Configuration panel to select the Java Naming and Directory Interface (JNDI) name of the emitter factory profile to associate the cluster with the emitter factory profile you created in step 2 on page 27.

When you start the servers and any installed applications, they will start normally.

You can now install applications on the cluster.

Preparing a server or cluster to support Service Component Architecture applications

A server or cluster can be configured to support WebSphere Process Server Service Component Architecture (SCA) applications. Servers and clusters can host SCA applications, application destinations, or both.

An SCA application (also called a service application) is one that requires the use of one or more of the following buses: SCA.APPLICATION.*CellName*.Bus, SCA.SYSTEM.*CellName*.Bus, or BPC.*CellName*.Bus. Each application uses a set of Java Message Service queues, 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.

By default, new servers and clusters are not configured to host SCA applications and their destinations. To enable this support, use the Service Component Architecture page in the administrative console. There are three basic scenarios for configuring a server or cluster to support SCA applications:

- 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 destinations. See "Hosting Service Component Architecture applications without hosting destinations".
- The server or cluster hosts SCA applications and application destinations. In this scenario, the required messaging engines are configured on the server or cluster. Note that these messaging engines can also host destinations required by applications with other deployment targets. See "Hosting Service Component Architecture applications and application destinations" on page 30.
- The server or cluster hosts the destinations for applications deployed on a remote server or cluster. Even though the server or cluster does not host SCA applications, service bus members must be configured as described in "Hosting Service Component Architecture applications and application destinations" on page 30.

Hosting Service Component Architecture applications without hosting destinations

A server or cluster can be configured as a deployment target for Service Component Architecture (SCA) applications whose destinations are located on a remote server or cluster.

Ensure that you have a remote service integration bus member already configured with the messaging engines needed to host destinations. See "Hosting Service Component Architecture applications and application destinations" on page 30 for instructions on configuring bus members.

You must be logged in as administrator or configurator to perform the following task.

To configure a server or cluster to host SCA applications, use the following procedure.

1. From within the administrative console, click one of the following, depending on your scope:
   - To configure a server: **Servers > Application Servers >** *server_name*> **Service Component Architecture**
   - To configure a cluster: **Servers > Clusters >** *cluster_name*> **Service Component Architecture**
2. Click **Use a remote destination location**.
3. Use the associated drop-down menu to specify the name of the remote server or cluster that is going to host the destinations.
4. Click **OK**.

The server or cluster can now host SCA applications. In addition, the SchedulerCalendar application is now available to support application hosting.

- If you are planning to use business processes defined with the Business Process Execution Language (BPEL) on this cluster or server, you must also use the Business Process Container wizard to perform the necessary configuration.
- If you are planning to use applications that contain human tasks or if you plan to use the Business Process Choreographer Explorer in this cluster or server, you must also use the Human Task Container wizard to perform the necessary configuration.

Hosting Service Component Architecture applications and application destinations

A server or cluster can be configured to host Service Component Architecture (SCA) applications and application destinations. These destinations can be used to support SCA applications installed locally or on a remote server or cluster.

You must be logged in as administrator or configurator to perform the following task.

This task involves configuring the system and application buses. You can use existing data sources for one or both of the buses, or you can configure new data sources. Note the following considerations for data sources:

- If you are using a DB2 database with your messaging engines, you can use the same database instance for both the system and application bus data sources, but you must specify a unique schema name for each bus. (The schema contains tables for the data source; each messaging engine stores its resources in a single schema, and each database schema is used by one messaging engine only.)
- If you are using any other supported database, you must use a different database instance for the system and application bus data sources.

To configure a server or cluster to host SCA applications and destinations, use the following procedure.

1. From within the administrative console, click one of the following, depending on your scope:
   - To configure a server: **Servers > Application Servers >** *server_name*> **Service Component Architecture**
   - To configure a cluster: **Servers > Clusters >** *cluster_name*> **Service Component Architecture**
2. Click **Configure a destination location**.
3. Use the **JDBC provider** drop-down menu to specify the appropriate template for the JDBC provider. The JDBC provider you select determines the type of database that can be used for the system and application buses.

4. In the **User name** field, type the ID you are using to access the data sources for the system and application buses.
5. In the **Password** field, type the password associated with the user ID.
6. Specify the configuration for the SCA system bus, as follows.

Type of data source to use	Steps to perform
Existing data source **Note:** You must use a different data source for the system bus and the application bus.	1. Click **Use existing data source**. 2. Use the drop-down menu to specify the data source you want to use.
New data source	1. If the database you are using supports schema names, use the **Schema name** field to specify the schema that contains the system bus data source tables. 2. In the **Database properties** field, enter any custom properties required by the SCA system bus data source.

7. Specify the configuration for the SCA application bus, as follows.

Type of data source to use	Steps to perform
Existing data source **Note:** You must use a different data source for the system bus and the application bus.	1. Click **Use existing data source**. 2. Use the drop-down menu to specify the data source you want to use.
New data source	1. If the database you are using supports schema names, use the **Schema name** field to specify the schema that contains the system bus data source tables. 2. In the **Database properties** field, enter any custom properties required by the SCA application bus data source.

8. Click **OK**.

The server or cluster can host SCA applications and application destinations. In addition, the SchedulerCalendar application is now available to support application hosting.

- If you are planning to use business processes defined with the Business Process Execution Language (BPEL) on this cluster or server, you must also use the Business Process Container wizard to perform the necessary configuration.
- If you are planning to use applications that contain human tasks or if you plan to use the Business Process Choreographer Explorer in this cluster or server, you must also use the Human Task Container wizard to perform the necessary configuration.

Administering applications and application services

This section describes how to use the administrative interfaces to administer WebSphere Process Server applications and application services, including business processes and tasks, business rules, and schedules.

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 7. 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.

Table 7. Buttons for administering enterprise applications (continued)

Button	Resulting action
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 for z/OS documentation.

Removing unneeded Service Component Architecture destinations

Generally, when you uninstall a module, the server deletes all inactive Service Component Architecture (SCA) destinations that the module uses. If you start the Java Virtual Machine (JVM) and choose to reuse SCA destinations by setting the system variable **SCA_recycleDestinations** to `true`, you must manually delete the SCA destinations.

This task assumes that you set the value of the JVM custom variable SCA_recycleDestinations to `true` either through the administrative console or in the startServer.bat or startServer.sh file.

If you have changed the default processing for SCA destinations, you must manually remove any inactive SCA destinations after you uninstall a module that contains SCA destinations.

Notes:

Active destination
A destination that belongs to a currently deployed module.

Inactive destination
A destination that does not belong to any currently deployed module.

1. From the command line, enter the deleteSCADestinations.jacl command. To delete a destination associated with a specific module, even if the destination is active, use the **-force** option.
2. Display the SCA destinations to make sure you have deleted the correct destinations.

The destinations are removed from the server.

Changing WebSphere MQ destinations after deploying Service Component Architecture modules

As your business environment evolves, it may be necessary to change which WebSphere MQ destinations a Service Component Architecture (SCA) module communicates with. Use this task to effect those changes.

You must know which Service Component Architecture (SCA) modules use the WebSphere MQ 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 by setting them in the **Custom Properties** of the **WebSphere MQ messaging provider > WebSphere MQ connection factory objects** :
  - You must set the custom property **SENDEXIT** to have the value com.ibm.ws.sca.internal.mq.exit.MQInternalSendExitImpl.
  - You must set the custom property **RECEXIT** to have the value com.ibm.ws.sca.internal.mq.exit.MQInternalReceiveExitImpl.
  - You must set the optional custom property **SENDEXITINIT** to a text string of your choice. Setting this property helps you identify the exit in a trace.
  - You must set the optional custom property **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.

After deploying SCA modules, you may need to change the WebSphere MQ destinations to meet new business requirements.

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 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 on page 34 and 6 for each destination you are changing.
8. Save the configuration changes.

Restart the applications you stopped in step 1 on page 34.

Service Component Architecture modules and WebSphere MQ

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 in WebSphere Integration Developer (WID).

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 by setting them in the **Custom Properties** of the **WebSphere MQ messaging provider > WebSphere MQ connection factory objects** :
  - You must set the custom property **SENDEXIT** to have the value com.ibm.ws.sca.internal.mq.exit.MQInternalSendExitImpl.
  - You must set the custom property **RECEXIT** to have the value com.ibm.ws.sca.internal.mq.exit.MQInternalReceiveExitImpl.
  - You must set the optional custom property **SENDEXITINIT** to a text string of your choice. Setting this property helps you identify the exit in a trace.
  - You must set the optional custom property **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.

Related concepts

"Bindings" on page 74
Binding information determines how a service connects to and interacts with an application. Specifically bindings are the protocols and transports assigned to imports and exports.

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. Use the Application Scheduler panel in the administrative console to administer these migrated scheduler entries as well.

In an Network Deployment environment, the Application Scheduler is automatically installed for every managed server and cluster member created - no additional action is needed. See Planning for a Network Deployment cell in WebSphere Application Server for z/OS information center for instructions on creating new managed servers and cluster members.

In a standalone server environment, Application Scheduler is optional. While creating the standalone server profile, you select a check box to configure and install Application Scheduler on that server.

Configuring the Application Scheduler for a standalone server

To use Application Scheduler, you must make sure that it is installed. See *Working with response files* in the WebSphere Process Server for z/OS Installation and Configuration PDF for information on configuring the Application Scheduler.

You need to augment the profile first.

This is an optional component and you must configure Application Scheduler to migrate WebSphere InterChange Server schedule entries to WebSphere Process Server Complete these steps to install the application server for a standalone server.

See the values for Application Scheduler in the response file and set the property values required in order to use Application Scheduler. For information on properties that pertain to Application Scheduler, see *Sample response files* in the WebSphere Process Server for z/OS Installation and Configuration PDF.

The installation is complete.

Application Scheduler is ready to use.

Accessing the Application Scheduler

Access Application Scheduler either programmatically using the Application Scheduler Mbean interface, or through the Application Scheduler panel of the administrative console.

For more information on accessing Application Scheduler see:

- "Accessing Application Scheduler using Application Scheduler MBean interface" on page 37
- "Displaying scheduler entries using the administrative console" on page 38

Accessing Application Scheduler using Application Scheduler MBean interface

Use the command line to invoke the Application Scheduler MBean

Perform the following to invoke Application Scheduler MBean.

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 = "WBISchedulerMB1";
   protected static final String TYPE_QUERY_CONSTANT = ",type=";
   protected static final String WBI_SCHED_MB_TYPE = "WBIScheduler";
   protected static final String SERVER_QUERY_CONSTANT = ",process=";
   protected static final String NODE_QUERY_CONSTANT = ",node=";

   serverName = "server1";
   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.

Then invoke the MBean's createScheduleEntry method:

```
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.

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

To display this panel and view existing scheduler events, follow these steps.

1. Select **Servers > Application Servers > ServerName**.
2. Select **Application Scheduler** from **Business Integration**.
3. Select the **scope** (cell, node, server) of the entries to display. The existing scheduled events for that scope are listed.

You can now edit the scheduler event, create a new scheduler event, or delete an existing event.

Creating a scheduled event

Administrative console provides a panel for creating new scheduled events.

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" for more information.

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.

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
- 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.

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.

Status	Description
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.

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 38 for more information.

There are times that you need to modify an event to fit a need. To modify an event, follow these steps.

1. Click the **Event** 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.

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.

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 38 for more information.

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.

1. Select the event to be deleted from the **Select** column.
2. Click **Delete**.
3. Click **OK** at the prompt.

The event is deleted.

Administering WebSphere Process Server resources

The administrative interfaces enable you to administer the resources associated with WebSphere Process Server, including selectors, target components, adapters, and the Extended Messaging Service.

Administering adapters

WebSphere Adapters, version 6.0, and WebSphere Business Integration Adapters (based on WebSphere Business Integration Framework, version 2.6) provide an approach to enterprise information systems (EIS) integration which is application module oriented.

WebSphere Adapters (sometimes referred to as resource adapters, JCA Adapters, or J2C Adapters) are compliant with J2EE Connector Architecture (JCA 1.5). JCA is the J2EE standard for EIS connectivity. EIS Import and EIS Export provide SCA application modules with a consistent view of those services outside the module. This allows components to communicate with the variety of EIS systems using the consistent SCA programming model.

WebSphere Adapters are assembled in WebSphere Integration Developer from imported RAR files and then exported as an Enterprise Application Archive (EAR) file and deployed on WebSphere Process Server.

Importing and exporting capabilities within the Service Component Architecture define the external interfaces or access points that a service module has in WebSphere Process Server. Imports and exports can be either to other modules within the same application, or to other applications on EISs. Imports identify services outside a module, making them callable from within the module. Exports allow components in a module to provide their services to external clients. A module level import or export permits modules to access other modules. A system level import or export permits your applications to access applications on EISs as if they were local components, which allows your applications to work with WebSphere Adapters and 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. For more information about working with these adapters and WebSphere Process Server, see the WebSphere Business Integration Adapter information center.

Working with WebSphere Adapters

WebSphere Adapters can be installed and administered using the administrative console.

To perform this task you must have access to the administrative console, and have security permission to alter server settings within the console.

This task should be performed if you have a WebSphere Adapter, that may or may not be embedded within an application. Use the administrative console to work with this adapter.

1. Install the WebSphere Adapter

   The process of installing WebSphere Adapters depends on whether they are embedded within an application:

Option	Description
Standalone adapters	Use the administrative console to install the adapter. **Note:** Standalone WebSphere Adapters are not supported in WebSphere Process Server v6.0.
Embedded adapters	The adapter will be installed as part of the application installation.

2. Administer the adapter.
3. Configure J2EE connection factories.

Overview of adapters

There are two types of adapter that can be used in WebSphere Process Server: WebSphere Business Integration Adapters and WebSphere Adapters. An overview of the functionality of these adapters is provided.

Adapters provide connectivity to access data, technologies and protocols that enhance integration infrastructure. They extract data and transaction information from cross-industry and industry-specific packaged applications and connect them to a central server.

Two types of adapter are supported in WebSphere Process Server:

- **WebSphere Business Integration Adapters** consist of a collection of software, application program interfaces (APIs) and tools that enable applications to exchange business data through an integration broker.
- **WebSphere Adapters** (sometimes referred to as Resource Adapters) are the preferred technology. They enable managed, bidirectional connectivity between enterprise information systems (EISs) and J2EE components supported by WebSphere Process Server.

Each business application requires its own application-specific adapter to participate in the business integration process. Adapters allow communication from various Enterprise Applications, such as Seibel or PeopleSoft to WebSphere Process Server.

WebSphere Business Integration Adapters allow multiple applications to communicate to fully integrate your system. Users are encouraged to use WebSphere Adapters which are fully compliant with the Java 2 Enterprise Edition (J2EE) Connector Architecture (JCA) version 1.5. The WebSphere Business Integration Adapters are not JCA-compliant, and run outside the application server.

The WebSphere Business Integration Adapters are the appropriate choice if a WebSphere Adapter does not yet exist for your application. If you have purchased WebSphere Business Integration Adapter licenses you may continue to use them, but you are encouraged to migrate to WebSphere Adapters as these will provide an updated way to connect to popular packaged software.

For information on developing applications using adapters see the WebSphere Integration Developer Information Center.

For information on installing, deploying and configuring WebSphere Adapters see the WebSphere Adapters Information Center.

For information on developing WebSphere Adapters for WebSphere Process Server, see (DeveloperWorks with the WebSphere Adapter Toolkit Guide).

Introduction to WebSphere Adapters:

WebSphere Adapters provide an integrated mechanism for communication between WebSphere Process Server applications and enterprise information systems. Your applications act upon the data stored in the enterprise information systems.

WebSphere Adapters implement the Java 2 Enterprise Edition (J2EE) Connector architecture (JCA) version 1.5. They are referred to as WebSphere Adapters or Resource Adapters. They manage bidirectional connectivity between enterprise information systems (EISs) and J2EE components supported by WebSphere Process Server.

JCA is designed to facilitate data sharing and to integrate new J2EE applications with legacy and other EISs. JCA stipulates how to develop a WebSphere Adapter that can:

- Plug into any J2EE-compliant application server.
- Connect an application running on that server with an EIS.
- Enable data exchange between the J2EE application and the EIS.

The JCA standard accomplishes this by defining a series of contracts that govern interactions between an EIS and J2EE components within an application server. Fully compliant with the JCA standard, WebSphere Adapters have been developed to run on WebSphere Process Server. Compliance with JCA has several advantages:

- JCA is an open standard,
- JCA is the J2EE standard for EIS connectivity,
- JCA provides a managed framework.

Each WebSphere Adapter is made up of the following:

- **Foundation classes** These implement a generic set of contracts that WebSphere Process Server uses to manage interactions between J2EE applications and all WebSphere Adapters. These quality of service and life cycle management contracts, also known as system contracts, define the service provider interface (SPI). For example, system contracts specify security credential management, connection pooling and transaction management parameters.

- **EIS subclasses** These generic and EIS-specific subclasses define the Common Client Interface (CCI) and EIS API contracts. For example, Activation and Connection Specs allow WebSphere Process Server to manage incoming and outgoing events for the WebSphere Adapter.
- **Enterprise Metadata Discovery** This utility introspects the EIS to generate service data objects (SDOs) and other artifacts that are compiled in a standard Enterprise Application Archive (EAR) file.

A simplified version of the operation of a WebSphere Adapter is shown schematically in figure 1:

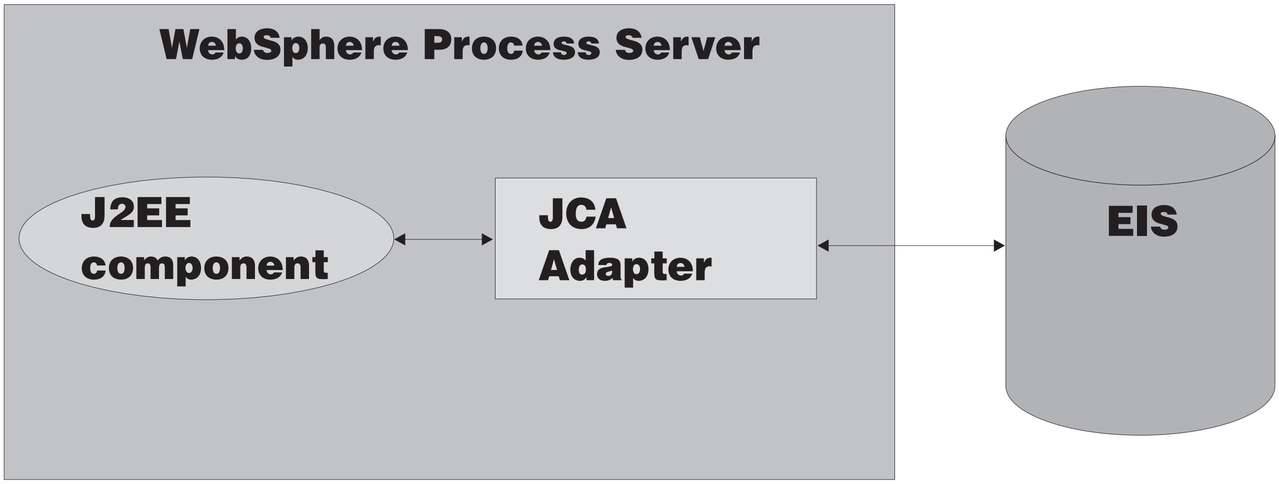


Figure 3. Simplified schematic of a WebSphere Adapter.

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 WebSphere Process Server and the EIS for the two types of adapter.

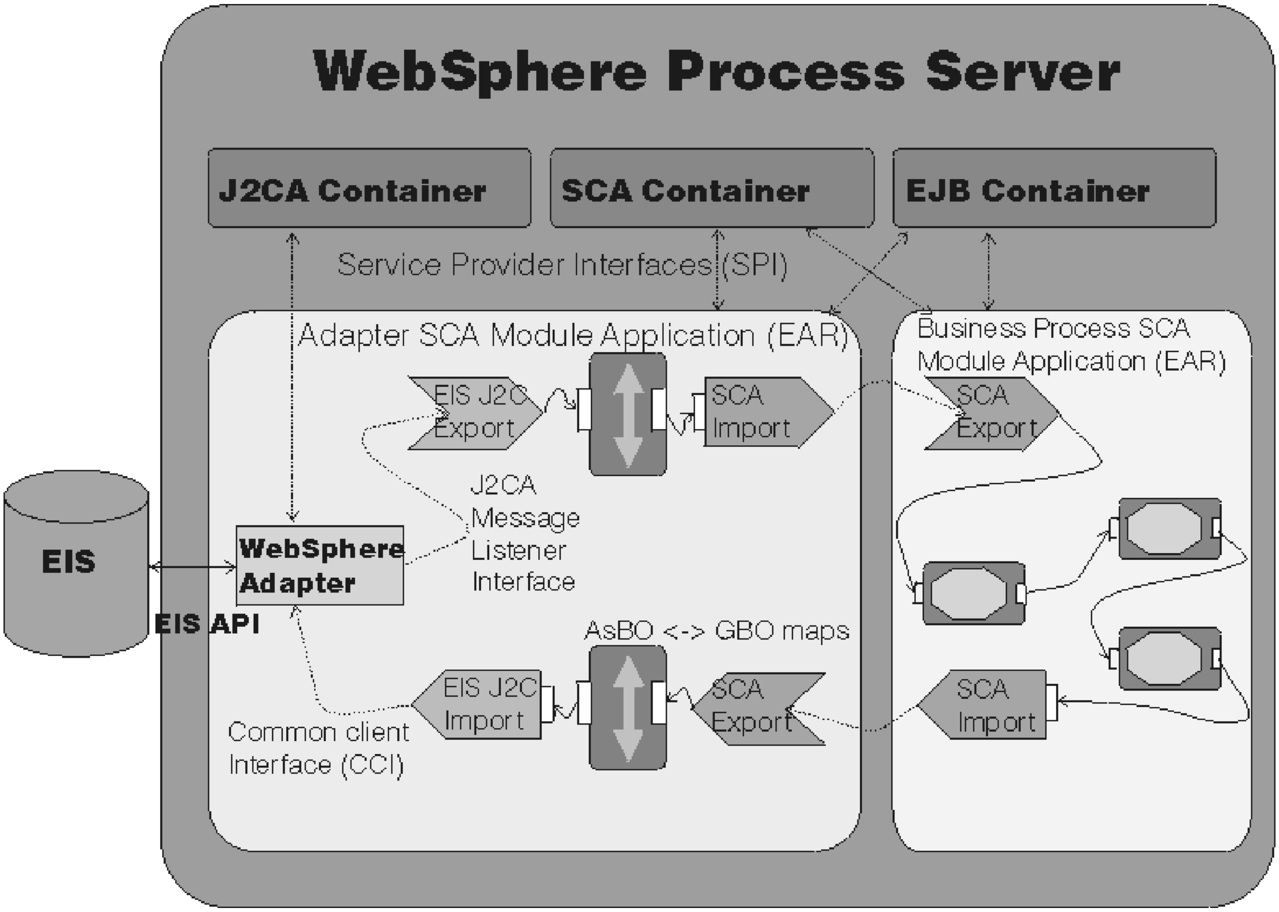


Figure 4. Detailed schematic of a WebSphere Adapter.

Figure 1 depicts a WebSphere Adapter managing the connectivity between a J2EE component supported by WebSphere Process Server and the EIS. The WebSphere Adapter resides inside WebSphere Process Server.

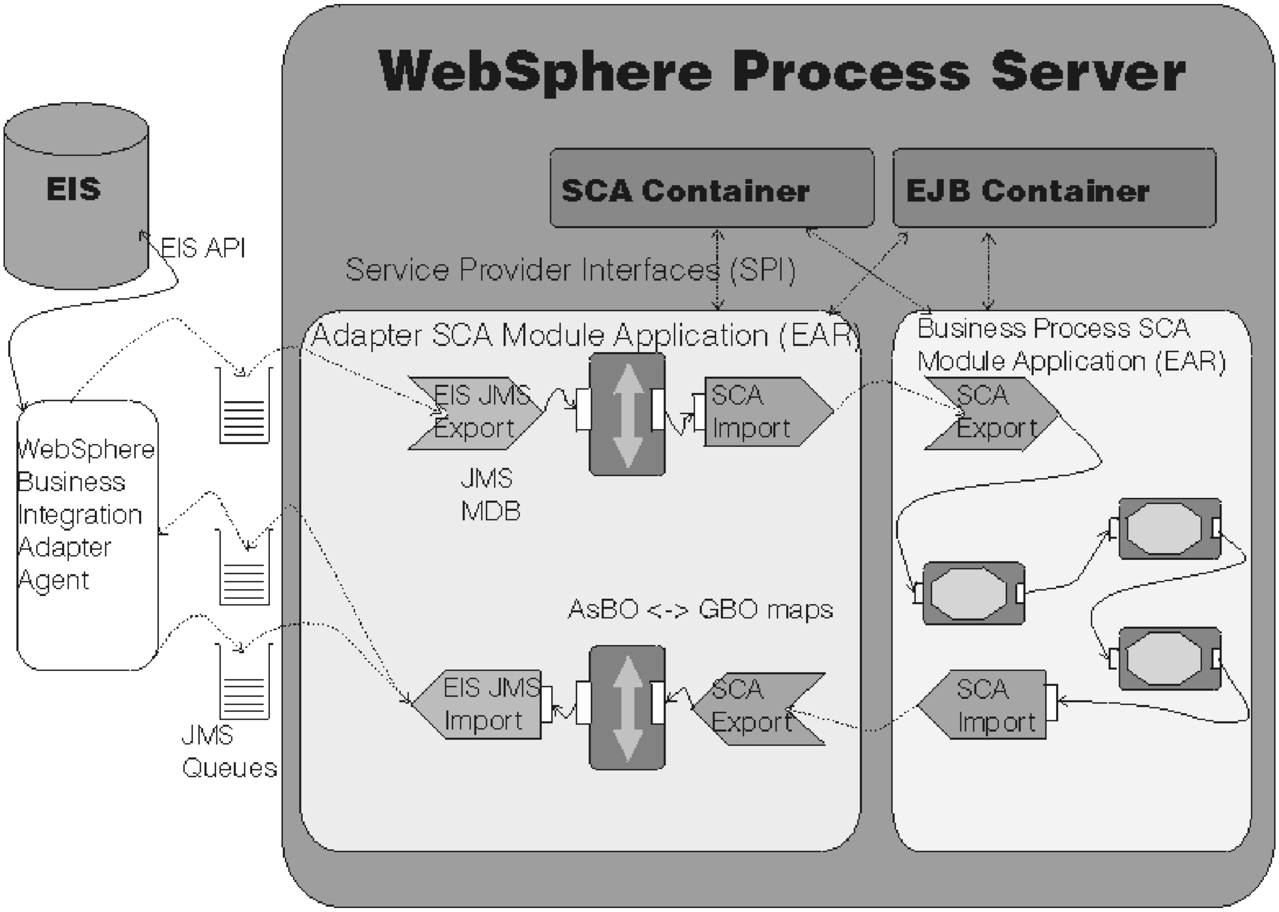


Figure 5. Detailed schematic of a WebSphere Business Integration Adapter.

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 via a Java Messaging Service (JMS) transport layer.

Differences between the two types of adapter include:

Table 8. 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.
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.

Table 8. Differences between WebSphere Adapters and WebSphere Business Integration Adapters. (continued)

Feature	WebSphere Adapters	WebSphere Business Integration Adapters
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 WebSphere Process Server.	Reside outside WebSphere Process Server. The server or integration broker communicates with the adapter via a Java Message Service (JMS) transport layer.

WebSphere Adapters are the recommended tool.

Advantages of using WebSphere Adapters:

WebSphere Adapters are the preferred choice in WebSphere Process Server. There are several key areas in which WebSphere Adapters have an advantage over WebSphere Business Integration Adapters.

WebSphere Adapters provide several advantages over WebSphere Business Integration Adapters. These advantages are summarized here:

- Integration - WebSphere Adapters are integrated into WebSphere Process Server, WebSphere Business Integration Adapters are outside the application server.
- JCA compliance - only WebSphere Adapters are fully compliant with JCA version 1.5.
- Request processing - WebSphere Adapters do not rely on the WebSphere Adapter Framework nor an integration server for request initiation with Enterprise Information Systems.
- Connectivity - WebSphere Adapters do not rely on the WebSphere Adapter Framework for connectivity, but use JCA contracts to manage life cycle tasks.
- Data models - WebSphere Adapters use an enterprise service discovery wizard to parse an EIS and develop Service Data Objects (SDOs). The enterprise service discovery wizard is part of the WebSphere Adapter implementation. WebSphere Business Integration Adapters use a separate object.
- Event notification - WebSphere Adapters use a subclass of EventStore to retrieve events from an EIS, whereas WebSphere Business Integration Adapters use a pollFor Events method.

There are a limited set of WebSphere Adapters available but it is recommended to use them wherever possible.

Installing applications with embedded WebSphere Adapters

If an application is developed with a WebSphere Adapter embedded, the adapter is deployed with the application. You do not need to install the adapter separately. The steps to install an application with an embedded adapter are described.

This task should only be performed if the application is developed with an embedded WebSphere Adapter.

1. Assemble an application with resource adapter archive (RAR) modules in it. See Assembling applications.
2. Install the application following the steps in Installing a new application. In the Map modules to servers step, specify target servers or clusters for each RAR file. Be sure to map all other modules that use the resource adapters defined in the RAR modules to the same targets. Also, specify the Web servers as targets that serve as routers for requests to this application. The plug-in configuration file (plugin-cfg.xml) for each Web server is generated based on the applications that are routed through it.

   Note: When installing a RAR file onto a server, WebSphere Process Server looks for the manifest (MANIFEST.MF) for the connector module. It looks first in the connectorModule.jar file for the RAR file and loads the manifest from the connectorModule.jar file. If the class path entry is in the manifest from the connectorModule.jar file, then the RAR uses that class path. To ensure that the installed connector module finds the classes and resources that it needs, check the Class path setting for the RAR using the console. For more information, see Resource Adapter settings and WebSphere relational resource adapter settings.
3. Save the changes. Click **Finish > Save**.
4. Create connection factories for the newly installed application.
   a. Open the administrative console.
   b. Select the newly installed application Click **Applications > Enterprise Applications >** *application name*.
   c. Click **Connector Modules** in the Related Items section of the page.
   d. Select the RAR file. Click on ***filename.rar***.
   e. Click **Resource adapter** in the Additional Properties section of the page.
   f. Click **J2C Connection Factories** in the Additional Properties section of the page.
   g. Click on an **existing connection factory** to update it, or **New** to create a new one.

      Note: If the WebSphere Adapter was configured using an EIS Import or EIS Export a ConnectionFactory or ActivationSpec will exist and can be updated.

   Linux and UNIX: If you install an adapter that includes native path elements, consider the following: If you have more than one native path element, and one of the native libraries (native library A) is dependent on another library (native library B), then you must copy native library B to a system directory. Because of limitations on most UNIX® systems, an attempt to load a native library does not look in the current directory.

   After you create and save the connection factories, you can modify the resource references defined in various modules of the application and specify the Java Naming and Directory Interface (JNDI) names of the connection factories wherever appropriate.

   Note: A given native library can only be loaded one time for each instance of the Java virtual machine (JVM). Because each application has its own classloader, separate applications with embedded RAR files cannot both use the same native library. The second application receives an exception when it tries to load the library.

If any application deployed on the application server uses an embedded RAR file that includes native path elements, then you must always ensure that you shut down the application server cleanly, with no outstanding transactions. If the application server does not shut down cleanly it performs recovery upon server restart and loads any required RAR files and native libraries. On completion of recovery, do not attempt any application-related work. Shut down the server and restart it. No further recovery is attempted by the application server on this restart, and normal application processing can proceed.

WebSphere Adapter:

A WebSphere Adapter (or JCA Adapter, or J2C Adapter) is a system-level software driver that a Java application uses to connect to an enterprise information system (EIS). WebSphere Adapters conform to version 1.5 of the JCA specification.

A WebSphere Adapter plugs into an application server and provides connectivity between the EIS, the application server, and the enterprise application.

An application server vendor extends its system once to support the J2EE Connector Architecture (JCA) and is then assured of seamless connectivity to multiple EISs. Likewise, an EIS vendor provides one standard WebSphere Adapter with the capability to plug into any application server that supports the connector architecture.

WebSphere Process Server provides the WebSphere Relational Resource Adapter (RRA) implementation. This WebSphere Adapter provides data access through JDBC calls to access the database dynamically. The connection management is based on the JCA connection management architecture. It provides connection pooling, transaction, and security support. WebSphere Process Server version 6.0 supports JCA version 1.5.

Data access for container-managed persistence (CMP) beans is managed by the WebSphere Persistence Manager indirectly. The JCA specification supports persistence manager delegation of the data access to the WebSphere Adapter without specific knowledge of the back-end store. For the relational database access, the persistence manager uses the relational resource adapter to access the data from the database. You can find the supported database platforms for the JDBC API at the WebSphere Process Server prerequisite Web site.

IBM supplies resource adapters for many enterprise systems separately from the WebSphere Process Server package, including (but not limited to): the Customer Information Control System (CICS®), Host On-Demand (HOD), Information Management System (IMS™), and Systems, Applications, and Products (SAP) R/3.

In WebSphere Process Server, EIS Imports and EIS Exports are used to interface with WebSphere Adapters. As an alternative, applications with WebSphere Adapters can be written by developing EJB session beans or services with tools such as Rational® Application Developer. The session bean uses the javax.resource.cci interfaces to communicate with an enterprise information system through the WebSphere Adapter.

WebSphere Adapter deployment considerations:

The deployment of WebSphere Adapters requires specific options regarding scope.

You can deploy a WebSphere Adapter in two ways, using the administrative console:

- Standalone - the adapter is installed at the node level and is not associated with a specific application.

  Note: Deployment of standalone WebSphere Adapters is not supported in WebSphere Process Server v6.0.
- Embedded - the adapter is part of an application, deploying the application also deploys the adapter.

For embedded WebSphere Adapters:

- the RAR file can be application-scoped within an SCA module (with EIS imports or exports).
- the RAR file can be application-scoped within a non-SCA module. The application itself, containing the EIS imports and exports, is a separate SCA module.

You should not install standalone WebSphere Adapters.

Note: The administrative console does not preclude the installation of standalone WebSphere Adapters, but this should not be done. WebSphere Adapters should be embedded in applications.

Only embedded WebSphere Adapters are appropriate for deployment in WebSphere Process Server. Furthermore, deployment of an embedded WebSphere Adapter is only supported for RAR files that are application-scoped within an SCA module; deployment in a non-SCA module is not supported.

Installing Standalone WebSphere Adapters:

If you intend to use a standalone WebSphere Adapter you should install it, as described here. You can alternatively use an embedded adapter, which is installed automatically as part of the installation of the associated application.

Note: WebSphere Adapters should be embedded in applications. Standalone WebSphere Adapters are not support in WebSphere Process Server v6.0. These instructions are for reference only.

You should configure the database before installing the adapter.

You must have access to, and be part of the necessary security role for, the administrative console to perform this task.

1. Open the Install RAR file dialog window. On the administrative console:
   a. Expand **Resources**
   b. Click **Resource Adapters**
   c. Select the scope at which you want to define this resource adapter. (This scope becomes the scope of your connection factory). You can choose cell, node, cluster, or server.
   d. Click **Install RAR**

   A window opens in which you can install a JCA connector and create, for it, a WebSphere Adapter. You can also use the **New** button, but the **New** button creates only a new resource adapter (the JCA connector must already be installed on the system).

Note: When installing a RAR file using this dialog, the scope you define on the Resource Adapters page has no effect on where the RAR file is installed. You can install RAR files only at the node level. The node on which the file is installed is determined by the scope on the Install RAR page. (The scope you set on the Resource Adapters page determines the scope of the new resource adapters, which you can install at the server, node, or cell level.)

2. Install the RAR file

   From the dialog, install the WebSphere Adapter in the following manner:

   a. Browse to the location of the JCA connector. If the RAR file is on the local workstation select **Local Path** and browse to find the file. If the RAR file is on a network server, select **Server path** and specify the fully qualified path to the file.
   b. Click **Next**
   c. Enter the resource adapter name and any other properties needed under General Properties. If you install a J2C Resource Adapter that includes native path elements, consider the following: If you have more than one native path element, and one of the native libraries (native library A) is dependent on another library (native library B), then you must copy native library B to a system directory. Because of limitations on most UNIX systems, an attempt to load a native library does not look in the current directory.
   d. Click **OK**.

WebSphere Adapter applications as members of clusters:

WebSphere Adapter module applications can be cloned as members of a cluster under certain conditions.

WebSphere Adapter module applications can be one of three types, depending on the flow of information through the adapter:

- A WebSphere Adapter application with only EIS exports - only inbound traffic.
- A WebSphere Adapter application with only EIS imports - only outbound traffic.
- A WebSphere Adapter application with both EIS imports and exports - bidirectional traffic.

Clusters are used to provide scalability and availability to your applications in a network deployment environment.

WebSphere Adapter module applications that have either inbound or bidirectional traffic, cannot be cloned as members of a cluster. An application with purely outbound traffic can be cloned as a member of a cluster.

An application that has an inbound or bidirectional WebSphere Adapter (that is, including EIS exports) can still be given availability in a network deployment by use of an external Operating System High Availability (HA) management software package, such as HACMP™, Veritas or Microsoft® Cluster Server.

WebSphere Business Integration Adapter applications as members of clusters:

WebSphere Business Integration Adapter module applications can be cloned as members of a cluster under certain conditions.

WebSphere Business Integration Adapter module applications can be one of three types, depending on the flow of information through the adapter:

- A WebSphere Business Integration Adapter application with only EIS exports - only inbound traffic.
- A WebSphere Business Integration Adapter application with only EIS imports - only outbound traffic.
- A WebSphere Business Integration Adapter application with both EIS imports and exports - bidirectional traffic.

Clusters are used to provide scalability and availability to your applications in a network deployment environment.

WebSphere Business Integration Adapter module applications that have either inbound or bidirectional traffic, cannot be cloned as members of a cluster. An application with purely outbound traffic can be cloned as a member of a cluster.

An application which has inbound or bidirectional WebSphere Business Integration Adapter (i.e., including EIS exports) can still be given availability in a network deployment by use of an external Operating System High Availability (HA) management software package, such as HACMP, Veritas or Microsoft Cluster Server.

Administering a WebSphere Adapter using the administrative console

Administer and configure your WebSphere Adapters using the WebSphere Process Server administrative console.

In order to perform this task, you must have security permission to change settings in the administrative console.

When you have installed a WebSphere Adapter, you can administer it using the administrative console.

1. Open the resource adapter console for the adapter you wish to administer. On the administrative console:
   a. Expand **Resources**
   b. Click **Resource Adapters**
   c. Choose the WebSphere Adapter to administer.
2. Alter desired properties under the General Properties or Additional Properties headings.

   When you have finished making changes click the **Apply** button.

Changes are only applied to the local configuration until you save them to the master configuration.

Note: The server may need to be restarted in order for changes to become effective.

WebSphere Adapter administrative console settings:

The administrative console settings for WebSphere Adapters and their default values are described here.

Purpose

Use the administrative console to edit the settings for your WebSphere Adapters.

To view the administrative console:
- Expand **Resources**
- Click **Resource Adapters**
- Choose the resource adapter from the list.

The various fields are described here:

Scope Specifies the level to which this resource definition is visible. For general information, see Administrative console scope settings in the Related Reference section. The Scope field is a read only string field that shows where the particular definition for a resource adapter is located. This is set either when the resource adapter is installed (which can only be at the node level) or when a new resource adapter definition is added.

Name Specifies the name of the resource adapter definition. This property is required. A string with no spaces meant to be a meaningful text identifier for the resource adapter.

Property	**Value**
Data type	String

Description Specifies a text description of the resource adapter. A free-form text string to describe the resource adapter and its purpose.

Property	**Value**
Data type	String

Archive path Specifies the path to the RAR file containing the module for this resource adapter. This property is required.

Property	**Value**
Data type	String

Class path Specifies a list of paths or JAR file names which together form the location for the resource adapter classes. This includes any additional libraries needed by the resource adapter. The resource adapter code base itself is automatically added to the class path, but if anything outside the RAR is needed it can be specified here.

Property	**Value**
Data type	String

Native path Specifies a list of paths which forms the location for the resource adapter native libraries. The resource adapter code base itself is automatically added to the class path, but if anything outside the RAR is needed it can be specified here.

Property	**Value**
Data type	String

ThreadPool Alias Specifies the name of a thread pool that is configured in the server that is used by the resource adapter's Work Manager. If there is no thread pool configured in the server with this name, the default configured thread pool instance, named Default, is used. This property is only necessary if this resource adapter uses Work Manager.

Property	**Value**
Data type	String

Administering Connection Factories

A connection factory is used in communication between an application and enterprise information system (EIS).

An application component uses a connection factory to access a connection instance, which the component then uses to connect to the underlying EIS.

Examples of connections include database connections, Java Message Service connections, and SAP R/3 connections.

Configuring J2EE Connector connection factories in the administrative console:

Connection factories are used in mediating communication between an application and an enterprise information system (EIS). The administrative console can be used to administer connection factories.

This task should be performed if you have a standalone resource adapter. Embedded WebSphere Adapters require slightly different treatment. You must also have security permission to edit the administrative console.

If your application requires access to a non-relational database, you need to configure a resource adapter and a connection factory rather than a JDBC provider and a data source.

1. Select the resource adapter to administer. From the top level of the administrative console, perform the following steps:
   a. Expand **Resources**
   b. Select **Resource Adapters**
   c. Select the WebSphere Adapter that you want to administer.
2. Create a new connection factory for this adapter.

   Create the new connection factory by:
   a. Select JCA Connection Factories under the Additional Properties.
   b. Click **New**.
   c. Specify general properties.
   d. Specify authentication preference.
   e. Select aliases for component-managed authentication, container-managed authentication, or both.

      If no aliases are available, or you want to define a different one,
      1) Click **Apply**.
      2) Click **J2C Authentication Data Entries** under Related Items.
      3) Click **New**.
      4) Specify general properties.

        5) Click **OK**.
    f. Click **OK**.
3. Modify the connection pool properties of the newly created connection factory to optimize the behavior of the connection pool manager.

   Change connection pool values by:
    a. Select the new connection factory.
    b. Under the Additional Properties heading select **Connection pool properties**.
    c. Change any desired values by clicking the property name.
    d. Click **OK**.
    e. Under Additional Properties click **Custom properties**.
    f. Click any property name to change its value.
    g. Click **OK**.

   Related tasks

   "Configuring connection factories for WebSphere Adapters within applications" on page 62
   WebSphere Adapters can be standalone or embedded within applications. The process of configuring the connection factories depends whether the adapter is within the application. When a WebSphere Adapter is embedded in the application use the following instructions to configure your connection factory.

Setting general properties for connection pools:

You can assign general settings to a connection pool using the administrative console.

To change general settings for a connection pool you must have security permission to alter values on the administrative console.

In order to assign general properties of a connection pool, you must first create a connection factory for your WebSphere Adapter.

1. Open the Connection Pool Properties panel in the administrative console.

   From the top level of the administrative console:
    a. Expand **Resources**
    b. Click **Resource Adapter**
    c. Click on the WebSphere Adapter which has the connection factory you wish to administer.
    d. Click **J2C Connection Factories** under the Additional Properties heading.
    e. Select the connection factory to administer.
    f. Click **Connection Pool Properties** under the Additional Properties heading.
2. General connection pool properties such as timeouts, maximum and minimum connections and purge policy can be altered on this panel. Default values are supplied.
3. After changing the desired properties, click **OK**.

Connection pool settings:

You can change the values of various properties of a connection pool on the Connection Pool Properties panel of the administrative console.

Purpose

The Connection Pool Settings panel is used to assign general property values for connection pools. You can edit properties such as timeouts, purge policies and connection limits.

Configuration Tab

Connection timeout

Specifies the interval, in seconds, after which a connection request times out and a ConnectionWaitTimeoutException is thrown.

This value indicates the number of seconds a request for a connection waits when there are no connections available in the free pool and no new connections can be created, usually because the maximum value of connections in the particular connection pool has been reached. For example, if Connection Timeout is set to 300, and the maximum number of connections are all in use, the pool manager waits for 300 seconds for a physical connection to become available. If a physical connection is not available within this time, the pool manager initiates a ConnectionWaitTimeout exception. It usually does not make sense to retry the getConnection() method; if a longer wait time is required you should increase the Connection Timeout setting value. If a ConnectionWaitTimeout exception is caught by the application, the administrator should review the expected connection pool usage of the application and tune the connection pool and database accordingly.

If the Connection Timeout is set to 0, the pool manager waits as long as necessary until a connection becomes available. This happens when the application completes a transaction and returns a connection to the pool, or when the number of connections falls below the value of Maximum Connections, allowing a new physical connection to be created.

If Maximum Connections is set to 0, which enables an infinite number of physical connections, then the Connection Timeout value is ignored.

Property	Value
Data type	Integer
Units	Seconds
Default	180
Range	0 to max int

Maximum connections

Specifies the maximum number of physical connections that you can create in this pool.

These are the physical connections to the back-end resource. Once this number is reached, no new physical connections are created and the requester waits until a physical connection that is currently in use returns to the pool, or a ConnectionWaitTimeout exception is issued.

For example, if the Maximum Connections value is set to 5, and there are five physical connections in use, the pool manager waits for the amount of time specified in Connection Timeout for a physical connection to become free.

If Maximum Connections is set to 0, the connection pool is allowed to grow infinitely. This also has the side effect of causing the Connection Timeout value to be ignored.

If multiple standalone application servers use the same data source, there is one pool for each application server. If clones are used, one data pool exists for each clone. Knowing the number of data pools is important when configuring the database maximum connections.

You can use the Tivoli® Performance Viewer to find the optimal number of connections in a pool. If the number of concurrent waiters is greater than 0, but the CPU load is not close to 100%, consider increasing the connection pool size. If the Percent Used value is consistently low under normal workload, consider decreasing the number of connections in the pool.

Property	Value
Data type	Integer
Default	10
Range	0 to max int

Minimum connections

Specifies the minimum number of physical connections to maintain.

If the size of the connection pool is at or below the minimum connection pool size, the Unused Timeout thread does not discard physical connections. However, the pool does not create connections solely to ensure that the minimum connection pool size is maintained. Also, if you set a value for Aged Timeout, connections with an expired age are discarded, regardless of the minimum pool size setting.

For example if the Minimum Connections value is set to 3, and one physical connection is created, the Unused Timeout thread does not discard that connection. By the same token, the thread does not automatically create two additional physical connections to reach the Minimum Connections setting.

Property	Value
Data type	Integer
Default	1
Range	0 to max int

Reap time

Specifies the interval, in seconds, between runs of the pool maintenance thread.

For example, if Reap Time is set to 60, the pool maintenance thread runs every 60 seconds. The Reap Time interval affects the accuracy of the Unused Timeout and Aged Timeout settings. The smaller the interval, the greater the accuracy. If the pool maintenance thread is enabled, set the Reap Time value less than the values of Unused Timeout and Aged Timeout. When the pool maintenance thread runs, it discards any connections remaining unused for longer than the time value specified in Unused Timeout, until it reaches the number of connections specified in Minimum Connections. The pool maintenance thread also discards any connections that remain active longer than the time value specified in Aged Timeout.

The Reap Time interval also affects performance. Smaller intervals mean that the pool maintenance thread runs more often and degrades performance.

To disable the pool maintenance thread set Reap Time to 0, or set both Unused Timeout and Aged Timeout to 0. The recommended way to disable the pool maintenance thread is to set Reap Time to 0, in which case Unused Timeout and Aged Timeout are ignored. However, if Unused Timeout and Aged Timeout are set to 0, the pool maintenance thread runs, but only physical connections which timeout due to non-zero timeout values are discarded.

Property	Value
Data type	Integer
Units	Seconds
Default	180
Range	0 to max int

Unused timeout

Specifies the interval in seconds after which an unused or idle connection is discarded.

Set the Unused Timeout value higher than the Reap Timeout value for optimal performance. Unused physical connections are only discarded if the current number of connections exceeds the Minimum Connections setting. For example, if the unused timeout value is set to 120, and the pool maintenance thread is enabled (Reap Time is not 0), any physical connection that remains unused for two minutes is discarded. Note that accuracy of this timeout, as well as performance, is affected by the Reap Time value. See Reap Time for more information.

Property	Value
Data type	Integer
Units	Seconds
Default	1800
Range	0 to max int

Aged timeout

Specifies the interval in seconds before a physical connection is discarded.

Setting Aged Timeout to 0 supports active physical connections remaining in the pool indefinitely. Set the Aged Timeout value higher than the Reap Timeout value for optimal performance. For example, if the Aged Timeout value is set to 1200, and the Reap Time value is not 0, any physical connection that remains in existence for 1200 seconds (20 minutes) is discarded from the pool. Note that accuracy of this timeout, as well as performance, are affected by the Reap Time value. See Reap Time for more information.

Property	Value
Data type	Integer
Units	Seconds
Default	0
Range	0 to max int

Purge policy

Specifies how to purge connections when a stale connection or fatal connection error is detected.

Valid values are:

- EntirePool: All connections in the pool are marked stale. Any connection not in use is immediately closed. A connection in use is closed and issues a stale connection exception during the next operation on that connection. Subsequent getConnection() requests from the application result in new connections to the database opening. When using this purge policy, there is a possibility that some connections in the pool are closed unnecessarily when they are not stale. However, this is a rare occurrence. In most cases, a purge policy of EntirePool is the best choice.

- FailingConnectionOnly: Only the connection that caused the stale connection exception is closed. Although this setting eliminates the possibility that valid connections are closed unnecessarily, it makes recovery from an application perspective more complicated. Only the currently failing connection is closed, so there is a good possibility that the next getConnection() request from the application will return a connection from the pool that is also stale, resulting in more stale connection exceptions. The connection pretest function attempts to insulate an application from pooled connections that are not valid. When a back-end resource, such as a database, goes down, pooled connections that are not valid might exist in the free pool. This is especially true when the purge policy is failingConnectionOnly; in this case, the failing connection is removed from the pool. Depending on the failure, the remaining connections in the pool might not be valid.

Property	**Value**
Data type	String
Default	EntirePool

Setting advanced properties for connection pools:

You can assign advanced settings to a connection pool using the administrative console.

To change advanced settings for a connection pool you must have security permission to alter values on the administrative console.

In order to assign advanced properties of a connection pool, you must first create a connection factory for your WebSphere Adapter.

1. Open the Advanced Connection Pool Properties panel in the administrative console.

   From the top level of the administrative console:

   a. Expand **Resources**

   b. Click **Resource Adapter**

   c. Click on the WebSphere Adapter which has the connection factory you wish to administer.

   d. Click **J2C Connection Factories** under the Additional Properties heading.

   e. Select the connection factory to administer.

   f. Click **Connection Pool Properties** under the Additional Properties heading.

   g. Click **Advanced Connection Pool Properties** under the Additional Properties heading.

2. Advanced connection pool settings such as partitioning, protection from connection overloading and dealing with connections that are not-responding. Default values are supplied, but tuning these properties is likely to provide improved performance and is recommended.

3. After changing any desired properties click **OK**.

Connection pool advanced settings:

You can change advanced settings for the connection pool on the Advanced Connection Pool Settings panel of the administrative console.

Purpose

The Advanced Connection Pool Settings panel is used to assign values for connection pools. On this panel you can alter properties such as partitioning, avoiding connection overloading and dealing with connections that are not responding.

Configuration Tab

Number of shared partitions

Specifies the number of partitions that are created in each of the shared pools.

Property	Value
Data type	Integer
Default value	0
Range	0 to max int

Number of free pool partitions

Specifies the number of partitions that are created in each of the free pools.

Property	Value
Data type	Integer
Default value	0
Range	0 to max int

Free pool distribution table size

The free pool distribution table size is used for better distribution of the Subject and CRI hash values within a hash table to minimize collisions for faster retrieval of a matching free connection.

If there are many incoming requests with varying credentials, this value can help with the distribution of finding a free pool for a connection for that user. Larger values are more common for installations that have many different credentials accessing the resource. Smaller values (1) should be used if the same credentials apply to all incoming requests for the resource.

Property	Value
Data type	Integer
Default value	0
Range	0 to max int

Surge Threshold

Specifies the number of connections created before surge protection is activated.

Surge protection is designed to prevent overloading of a data source when too many connections are created at the same time. Surge protection is controlled by two properties, surge threshold and surge creation interval.

The surge threshold property specifies the number of connections created before surge protection is activated. After you reach the specified number of connections, you enter surge mode.

The surge creation interval property specifies the amount of time, in seconds, between the creation of connections when in surge mode.

For example, assume the follow settings:

- maxConnections = 50
- surgeThreshold = 10
- surgeCreationInterval = 30 seconds

If the connection pool receives 15 connection requests, 10 connections are created at about the same time. The 11th connection is created 30 seconds after the first 10 connections. The 12th connection is created 30 seconds after the 11th connection. Connections continue to be created every 30 seconds until there are no more new connections needed or you reach the maxConnections value.

Surge connection support starts if the surge threshold is > -1 and the surge creation interval is > 0. The surge threshold property has a default value of -1, which indicates that it is turned off.

wsadmin example

```
$AdminControl getAttribute $objectname surgeCreationInterval
$AdminControl setAttribute $objectname surgeCreationInterval 30
$AdminControl getAttribute $objectname surgeThreshold
$AdminControl setAttribute $objectname surgeThreshold 15
```

Property	Value
Data type	Integer
Default value	-1
Range	-1 to max int

Surge creation interval

Specifies the amount of time between connection creates when you are in surge protection mode.

If the number of connections specified in the surge threshold property have been made, each request for a new connection must wait to be created on the surge creation interval. This property has a default value of 20, which indicates that at least 20 seconds should pass between connections being created. Valid values for this property are any positive integer.

Property	Value
Data type	Integer
Default value	20
Range	0 to max int

Stuck timer time

A stuck connection is an active connection that is not responding or returning to the connection pool. If the pool appears to be stuck (you have reached the stuck threshold), a resource exception is given to all new connection requests until the pool is unstuck. The stuck timer time property is the interval for the timer. This is how often the connection pool checks for stuck connections. The default value is 5 seconds.

If an attempt to change the stuck time, stuck timer time, or stuck threshold properties using the **wsadmin** scripting tool fails, an IllegalState exception occurs. The pool cannot have any active requests or active connections during this request. For the stuck connection support to start, all three stuck property values must be greater than 0 and maximum connections must be greater than 0.

Also, the stuck timer time, if it is set, must be less than the stuck time value. In fact, it is suggested that the stuck timer time should be one-quarter to one-sixth the value of stuck time so that the connection pool checks for stuck connections 4 to 6 times before a connection is declared stuck. This reduces the likelihood of false positives.

wsadmin example

```
$AdminControl getAttribute $objectname stuckTime
$AdminControl setAttribute $objectname stuckTime 30
$AdminControl getAttribute $objectname stuckTimerTime
$AdminControl setAttribute $objectname stuckTimerTime 15
$AdminControl getAttribute $objectname stuckThreshold
$AdminControl setAttribute $objectname stuckThreshold 10
```

Property	**Value**
Data type	Integer
Default value	5
Range	0 to max int

Stuck time

A stuck connection is an active connection that is not responding or returning to the connection pool. If the pool appears to be stuck (you have reached the stuck threshold), a resource exception is given to all new connection requests until the pool is unstuck. The stuck time property is the interval, in seconds, allowed for a single active connection to be in use to the back-end resource before it is considered to be stuck.

Property	**Value**
Data type	Integer
Default value	0
Range	0 to max int

Stuck Threshold

A stuck connection is an active connection that is not responding or returning to the connection pool. If the pool appears to be stuck (you have reached the stuck threshold), a resource exception is given to all new connection requests until the pool is unstuck. An application can explicitly catch this exception and continue processing. The pool will continue to periodically check for stuck connections when the number of stuck connections is past the threshold. If the number of stuck connections drops below the stuck threshold, the pool will detect this during its periodic checks and enable the pool to begin servicing requests again. The stuck threshold is the number of connections that need to be considered stuck for the pool to be in stuck mode.

Property	**Value**
Data type	Integer
Default value	0
Range	0 to max int

Configuring connection factories for WebSphere Adapters within applications:

WebSphere Adapters can be standalone or embedded within applications. The process of configuring the connection factories depends whether the adapter is within the application. When a WebSphere Adapter is embedded in the application use the following instructions to configure your connection factory.

You must have the necessary permissions to edit the settings on the administrative console in order to perform this task.

1. Indicate the EAR file which contains the RAR file that you wish to upload and install.

   From the top level of the administrative console follow these steps:

   a. Expand **Applications**.

   b. Select **Install New Application**.

   c. Browse to the EAR file on the local or remote system.

      Select the radio button associated with the remote or local file system, and then specify a path to the EAR file.

   d. Click **Next**.

2. Install the application indicating that you wish to map to a J2C connection factory.

   Select resource ref mapping to a J2C connection factory and click **Next**.

3. After the application installs, create and configure a connection factory for the WebSphere Adapter embedded in the newly installed application.

   From the top level of the administrative console follow these steps:

   a. Expand **Applications**.

   b. Select **Enterprise Applications**.

   c. Select the newly installed application.

   d. Click **Connector Modules** under Related Items.

   e. Select the RAR file.

   f. Under Additional Properties, select **Resource Adapters**.

   g. Click **J2C Connection Factories** under Additional Properties.

   h. Click **New**.

   i. Specify any necessary general properties.

   j. **Optional:** Specify an Authentication alias for XA recovery, or use a component-managed authentication alias. This field is only displayed for resources that support XA transactions.

   k. **Optional:** Select a Component-managed authentication alias.

      If a desired alias is not available, or you want to define a different one,

      1) Click **Apply**.

      2) Click **J2C Authentication Data Entries** under Related Items.

      3) Click **New**.

      4) Specify general properties.

      5) Click **OK** to return to the J2C Connection Factories Settings panel.

      Select or define an alias, if any components of your application obtain connections from this connection factory with the empty-argument getConnection() method and with Application or Per connection factory authentication specified in the resource reference.

   l. Complete the creation of your new connection factory. Click **OK**.

4. **Optional:** Change any connection pool properties of your new connection factory.

   From the J2C Connection Factory Collection panel:

   a. Select the connection factory that you just created

   b. Open the Connection Pool settings panel Click **Connection pool properties** under the Additional Properties heading.

c. Change the values of any properties by clicking on the property name.
d. Confirm these changes to the connection pool settings. Click **OK**.

5. **Optional:** Change any custom properties

   From the Connection Factory Settings panel of the new connection factory:

   a. Click **Custom Properties** under the Additional Properties heading.
   b. Click any property name to change its value.

      Note: UserName and Password, if present, are overridden by the component-managed authentication alias you specified in a previous step.

   c. Click **Save**.

   Related tasks

   "Configuring J2EE Connector connection factories in the administrative console" on page 54
   Connection factories are used in mediating communication between an application and an enterprise information system (EIS). The administrative console can be used to administer connection factories.

J2C connection factory collection:

The J2C connection factory collection panel provides a selectable list of connection factories.

Purpose

Use this panel to select a connection factory, which represents one set of connection configuration values.

Application components such as enterprise beans have resource reference descriptors that refer to the connection factory, not the WebSphere Adapter. The connection factory is really a configuration properties list holder. In addition to the arbitrary set of configuration properties defined by the vendor of the WebSphere Adapter, there are several standard configuration properties that apply to the connection factory. These standard properties are used by the Java 2 Connectors connection pool manager in the application server run time and are not known by the vendor-supplied WebSphere Adapter code.

Connection Factory collection panel

Name

Specifies a list of the connection factory display names.

Property	Value
Data type	String

JNDI Name

Specifies the Java Naming and Directory Interface (JNDI) name of this connection factory.

Property	Value
Data type	String

Description

Specifies a text description of this connection factory.

Property	Value
Data type	String

Category

Specifies a string that you can use to classify or group this connection factory.

Property	Value
Data type	String

J2C connection factory settings:

You specify settings for various properties of a connection factory on the J2C Connection Factory Settings panel of the administrative console.

Purpose

The J2C Connection Factory Settings panel is used to assign general property values for the selected connection factory. You can edit properties such as time-outs, purge policies and connection limits.

Configuration tab

Connection Factory Interface

Specifies the fully qualified name of the Connection Factory Interfaces supported by the resource adapter.

This is a required property. For new objects, the list of available classes is provided by the resource adapter in a drop-down list. After you create the connection factory, the field is a read only text field.

Property	Value
Data type	Drop-down list or text

Authentication Alias for XA recovery

This optional field is used to specify the authentication alias that should be used during XA recovery processing.

If the WebSphere Adapter does not support XA transactions, then this field will not be displayed. The default value will come from the selected alias for application authentication (if specified).

Use Component-managed Authentication Alias

Selecting this radio button specifies that the alias set for component-managed authentication is used at XA recovery time.

Property	Value
Data type	Radio button

Specify:

Selecting this radio button enables you to choose an authentication alias from a drop-down list of configured aliases.

Property	Value
Data type	Radio button

Name

Specifies a list of connection factory display names.

This is a required property.

Property	Value
Data type	String

JNDI Name

Specifies the JNDI name of this connection factory.

For example, the name could be *eis/myECIConnection*.

After you set this value, save it and restart the server. You can see this string when you run the dumpNameSpace tool. This is a required property. If you do not specify a JNDI name, it is filled in by default using the Name field.

Property	Value
Data type	String
Default	eis/*display name*

Description

Specifies a text description of this connection factory.

Property	Value
Data type	String

Category

Specifies a string that you can use to classify or group this connection factory.

Property	Value
Data type	String

Component-managed Authentication Alias

Specifies authentication data for component-managed signon to the resource.

Choose from aliases defined under Security>JAAS Configuration> J2C Authentication Data.

To define a new alias not already appearing in the pick list:

- Click **Apply** to expose Related Items.
- Click **J2C Authentication Data Entries**.
- Define an alias.
- Click the connection factory name at the top of the J2C Authentication Data Entries page to return to the connection factory page.
- Select the alias.

Property	Value
Data type	Pick-list

Container-managed Authentication Alias

Specifies authentication data (a string path converted to userid and password) for container-managed signon to the resource.

Note: The container-managed authentication alias is superseded by the specification of a login configuration on the resource-reference mapping at deployment time, for components with res-auth=Container.

Choose from aliases defined under Security>JAAS Configuration> J2C Authentication Data.

To define a new alias not yet included in the list:

- Click **Apply** to expose Related Items.
- Click **J2C Authentication Data Entries**.
- Define an alias.
- Click the connection factory name at the top of the J2C Authentication Data Entries page to return to the connection factory page.
- Select the alias.

Property	Value
Data type	Pick-list

Authentication Preference

Specifies the authentication mechanisms defined for this connection factory.

Note: The authentication preference is superseded by the combination of the <res-auth> application component deployment descriptor setting and the specification of a login configuration on the resource-reference mapping at deployment time.

This setting specifies which of the authentication mechanisms defined for the corresponding resource adapter applies to this connection factory. Common values, depending on the capabilities of the resource adapter, are: KERBEROS, BASIC_PASSWORD, and None. If None is chosen, the application component is expected to manage authentication (<res-auth>Application</res-auth>). In this case, the user ID and password are taken from one of the following:

- The component-managed authentication alias
- UserName, Password Custom Properties
- Strings passed on the getConnection method

For example, if two authentication mechanism entries are defined for a resource adapter in the *ra.xml* document:

- <authentication-mechanism-type>BasicPassword</authentication-mechanism-type>
- <authentication-mechanism-type>Kerbv5</authentication-mechanism-type>

the authentication preference specifies the mechanism to use for container-managed authentication. An exception is issued during server startup if a mechanism that is not supported by the WebSphere Adapter is selected.

Property	Value
Data type	Pick-list
Default value	BASIC_PASSWORD

Mapping-Configuration Alias

Allows users to select from the Security > JAAS Configuration > Application Logins Configuration list.

Note: The Mapping-Configuration Alias is superseded by the specification of a login configuration on the resource-reference mapping at deployment time, for components with res-auth=Container.

The DefaultPrincipalMapping JAAS configuration maps the authentication alias to the userid and password. You may define and use other mapping configurations.

Property	**Value**
Data type	Pick-list

J2C connection factory advanced settings:

You can change advanced settings for the J2C connection factory on the J2C Connection Factory Advanced Settings panel of the administrative console.

Purpose

The J2C Connection Factory Advanced Settings panel is used to assign values for advanced properties of the connection factory. On this panel you can alter properties pertaining to the management of cached handles and logging of missing transactions.

Configuration tab

Manage cached handles

If checked, cached handles (handles held in inst vars in a bean) are tracked by the container.

Property	**Value**
Data type	Check box

Log missing transaction contexts

If checked, the container logs that there is a missing transaction context when a connection is obtained.

Property	**Value**
Data type	Check box

Configuring WebSphere Business Integration Adapters

The process of configuring and using the WebSphere Business Integration Adapters is three-fold:

1. Install the application EAR file.

   Installation of the application EAR file is accompanied by the creation of all necessary artifacts for the WebSphere Business Integration Adapter to work.
2. Set up the administration of the WebSphere Business Integration Adapter.

   In order to administer a WebSphere Business Integration Adapter you must:
   1. Create a connection queue factory
   2. Create a WebSphere Business Integration Adapter resource
   3. Enable the WebSphere Business Integration Adapter service.

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.

3. Manage the WebSphere Business Integration Adapter

   Use the administrative console to manage the WebSphere Business Integration Adapters.

Create the artifacts necessary for the WebSphere Business Integration Adapter to work:

To create the artifacts necessary for the WebSphere Business Integration Adapter to work, install the application EAR file.

In order to create the artifacts that you need to use a WebSphere Business Integration Adapter in WebSphere Process Server, you should follow 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. The required steps are described here.

You must install the application EAR file to create the artifacts required for the WebSphere Business Integration Adapter before you perform this task.

In order to have administrative control over a WebSphere Business Integration Adapter you must first perform the following administrative functions.

1. Create a Queue Connection Factory.

   From the top level of the administrative console follow these steps:

   a. Expand **Resources**.

   b. Expand **JMS Providers**.

   c. Select **Default Messaging**.

   d. Select **JMS queue connection factory**.

      Under the JMS subheading select **JMS queue connection factory**.

   e. Create a new JMS queue connection factory.

      Click **New**.

   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 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:

When a WebSphere Business Integration Adapter is running, it can be managed using the Manage the WebSphere Business Integration Adapter resources panel on the administrative console.

The WebSphere Business Integration Adapter must be running in order to be managed.

You can manage a WebSphere Business Integration Adapter from the administrative console. The Manage the WebSphere Business Integration Adapter resources panel allows you to choose one or more resources to manage and perform various administrative actions upon these resources.

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.
Resume	Changes the status of the selected resources from paused to active.
Shutdown	Changes the status of the selected resources from active to unavailable.

Generating service component definitions and the MQClientLink configuration file

Before using WebSphere Business Integration Adapters, it is necessary to generate service component definitions (SCA artifacts) and the MQClientLink configuration file (.wbia file). This is achieved with the WebSphere Business Integration Adapter Artifact Importer in the WebSphere Integration Developer environment.

The WebSphere Integration Developer uses the WebSphere Business Integration Adapter Artifact Importer to discover and import the WebSphere Business Integration Adapter Connector Configuration File and WebSphere Business Integration Adapter Business Objects directory and generate the desired service component definitions supporting the specified interaction-styles for the WebSphere Business Integration Adapter.

Note: This task is performed in WebSphere Integration Developer, and is described here only for reference. See the WebSphere Integration Developer information center for more details.

1. Obtain the necessary configuration files and Business Objects.
2. Use the WebSphere Business Integration Adapter Artifact Importer to generate the necessary service component definitions (SCA artifacts).

The service component definitions (SCA artifacts) and the WebSphere Process Server MQClientLink configuration file (.wbia file) are generated.

When an application is deployed to WebSphere Process Server, the service component definitions and the MQClientLink configuration (.wbia) file are handled automatically by the deployment tools. It is recommended that the configuration file be left in its default state, but it is possible to manually edit the file if required.

Installing EIS applications

An EIS application module can be deployed to a J2EE platform. The deployment results in an application, packaged as an EAR file deployed to the server. All the J2EE artifacts and resources are created, the application is configured and ready to be run.

The deployment to the J2EE platform creates the following J2EE artifacts and resources:

Table 9. Mapping from bindings to J2EE artifacts

Binding in the SCA module	Generated J2EE artifacts	Created J2EE resources
EIS Import	Resource References generated on the module Session EJB.	ConnectionFactory
EIS Export	Message Driven Bean, generated or deployed, depending on the listener interface supported by the Resource Adapter.	ActivationSpec
JMS Import	Message Driven Bean (MDB) provided by the runtime is deployed, resource references generated on the module Session EJB. Note that the MDB is only created if the import has a receive destination.	• ConnectionFactory • ActivationSpec • Destinations
JMS Export	Message Driven Bean provided by the runtime is deployed, resource references generated on the module Session EJB	• ActivationSpec • ConnectionFactory • Destinations

When the import or export defines a resource like a ConnectionFactory, the resource reference is generated into the deployment descriptor of the module Stateless Session EJB. Also, the appropriate binding is generated into the EJB binding file. The name, to which resource reference is bound, is either the value of the target attribute, if one is present, or default JNDI lookup name given to the resource, based on the module name and import name.

Upon deployment, the implementation locates the module session bean and uses it to lookup the resources.

During deployment of the application to the server, the EIS installation task will check for the existence of the element resource to which it is bound. If it does not exist, and the SCDL file specifies at least one property, the resource will be created and configured by the EIS installation task. If the resource does not exist, no action is taken, it is assumed that resource will be created before execution of the application.

When the JMS Import is deployed with a receive destination, a Message Driver Bean (MDB) is deployed. It listens for replies to requests that have been sent out. The MDB is associated (listens on) the Destination sent with the request in the JMSreplyTo header field of the JMS message. When the reply message arrives, the MDB uses its correlation ID to retrieve the callback information stored in the callback Destination and then invokes the callback object.

The installation task creates the ConnectionFactory and three destinations from the information in the import file. In addition, it creates the ActivationSpec to enable the runtime MDB to listen for replies on the receive Destination. The properties of the ActivationSpec are derived from the Destination/ConnectionFactory properties. If the JMS provider is a SIBus Resource Adapter, the SIBus Destinations corresponding to the JMS Destination are created.

When the JMS Export is deployed, a Message Driven Bean (MDB) (not the same MDB as the one deployed for JMS Import) is deployed. It listens for the incoming requests on the receive Destination and then dispatches the requests to be processed by the SCA. The installation task creates the set of resources similar to the one for JMS Import, an ActivationSpec, ConnectionFactory used for sending a reply and two Destinations. All the properties of these resources are specified in the export file. If the JMS provider is an SIBus Resource Adapter, the SIBus Destinations corresponding to JMS Destination are created.

Remote deployment of a WebSphere Adapter application

In a network deployment environment it may be convenient to remotely install WebSphere Adapter applications on nodes which reside closer to the EIS (e.g., on the same machine as the EIS) than the node which contains the consuming application (e.g., an application which contains a BPEL process). The best practice for such remote deployment is described herein.

To perform this task you must have a WebSphere Adapter embedded in an application. You must also be working in a network deployment environment with an operating Deployment Manager, and have sufficient privileges to install the application on a remote machine.

This task is appropriate when you have a consuming application, such as a BPEL process application, which requires communication with an EIS mediated by a WebSphere Adapter application in a network deployment environment.

1. Install the consuming application on any node of the cell.
2. Remotely install the WebSphere Adapter application on a node situated in close proximity to the EIS with which it interacts.

The consuming process can access the information in the EIS via the WebSphere Adapter application, without the need for installation of the adapter on every node in the cell.

A BPEL process application requiring communication with an EIS, mediated by a WebSphere Adapter application is a situation where remote deployment of the WebSphere Adapter application is ideal. The WebSphere Adapter application can reside on a node in the cell which is conveniently close to the EIS with which it interacts. The BPEL process application need not reside on the same node. A WebSphere Process Server Network Deployment environment following best practice guidelines would have an architecture similar to that shown in figure 1.

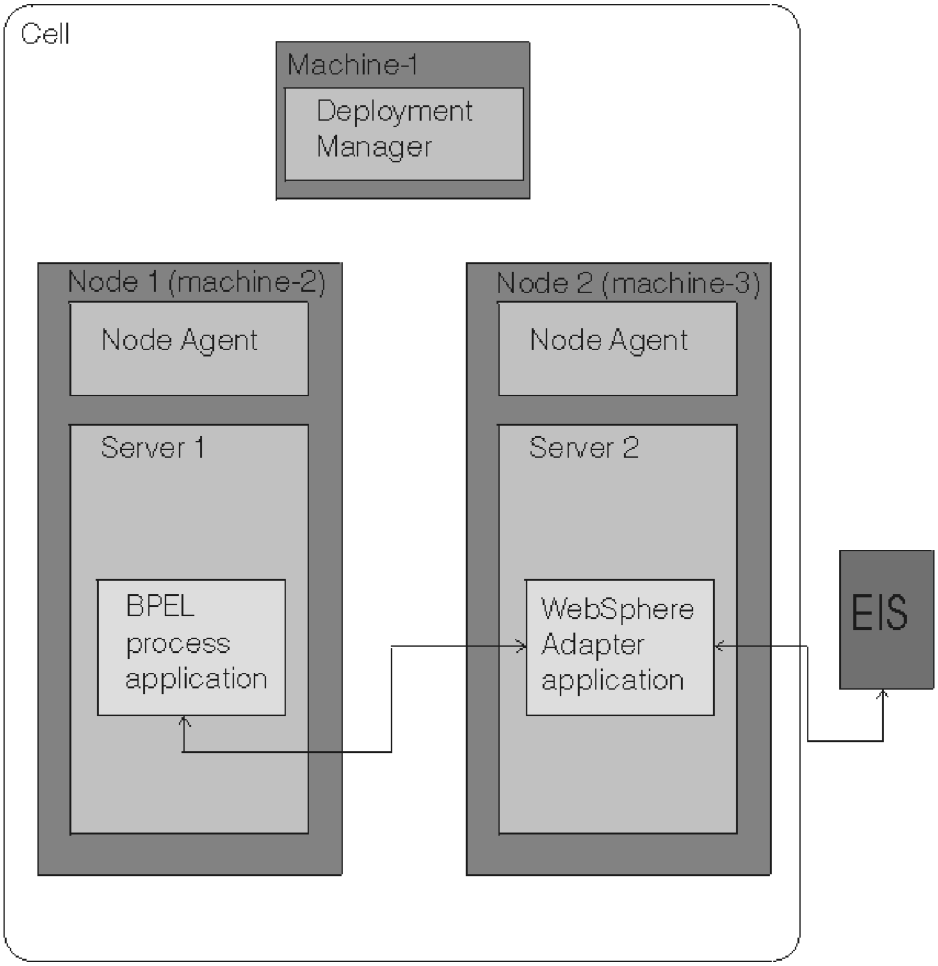


Figure 6. The cell architecture of a remote deployment of a WebSphere Adapter application in a network deployment environment.

Bindings

Binding information determines how a service connects to and interacts with an application. Specifically bindings are the protocols and transports assigned to imports and exports.

Your applications connect to services through a binding.

Bindings are a critical part of your import and export components. Bindings determine specifically how your import and export components interact with clients outside the module which contains the import and export components. The bindings specify the message format and protocol details for a particular interface.

Use JMS bindings 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.

IBM Web Services bindings can also by 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.

EIS and JMS bindings

The key bindings when it comes to EIS systems are JMS bindings and EIS bindings.

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. You cannot do this with an EIS binding. Since JMS is used for asynchronous communication, you can also use JMS as the binding between an external messaging source and a component in a module.

EIS bindings are 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. Essentially, EIS bindings are used to access external EIS systems or for external EIS systems to access applications deployed in WebSphere Process Server.

Related concepts

"Service Component Architecture modules and WebSphere MQ" on page 35

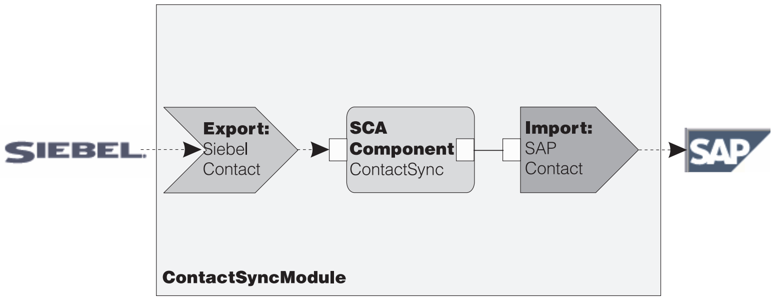
SCA EIS Binding

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 may require data to be transferred to or form an external EIS. When you create SCA modules requiring such connectivity, in addition to your SCA component you will include EIS imports and/or EIS exports to communicate with specific external EISs. Details of how to develop EIS imports and exports are provided in the WebSphere Integration Developer Information Center.

Example

A typical SCA module named ContactSyncModule, synchronizes contact information between a Siebel system and a SAP system. The SCA component named ContactSync listens via an EIS Service Export named Siebel Contact for changes to Siebel contacts. The ContactSync SCA component itself makes use of an SAP service via an EIS Service 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 81
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. The JMS binding is a specialization of the EIS import and export bindings.

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.

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 service 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 WebSphere Process Server contains 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 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.

EIS bindings support

An EIS module can be deployed to the J2EE platforms.

The EIS module, a SCA module that follows the EIS module pattern, can be deployed to the J2EE platform.

The deployment of EIS module to the J2EE platform results in an application which is ready to execute, packaged as an EAR file and deployed to the server. All the J2EE artifacts and resources are created; the application is configured and ready to be run.

Key features of an EIS import binding

An EIS import is a Service Component Architecture (SCA) import that allows components in the SCA module to use EIS services defined outside the SCA module. An EIS import, is used to transfer data from the SCA component to an external EIS.

Purpose

The role of the EIS import is to bridge the gap between SCA components and external EIS systems. External services can be treated as an EIS export. In this case, the external service sends its data in the form of a periodic notification.

The EIS import provides SCA components with a uniform view of the services 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 Service, 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, Interface Binding, using 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 service. This reflects the fact that the set of services, 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, 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.

The adapter interaction style allows for the EIS Export binding to invoke the target component either asynchronously or synchronously. The default is asynchronously.

Related concepts

"Data bindings" on page 104
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.

Administering EIS import bindings:

EIS import bindings are installed in WebSphere Process Server as part of your SCA applications. Administer your imports from the administrative console.

You must have permission to make changes to the master configuration in order to perform this task.

You have an installed application which includes an EIS import module.

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 Imports and expand the import 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 import binding. Expand Bindings and click on the binding that you want to administer. You can change the port or the name of the imported service.
4. Save changes to the configuration in order for you changes to take effect.

Key features of an EIS export binding

An EIS export is a Service Component Architecture (SCA) export, it allows services to be available to clients outside the SCA module. An EIS export, is used to transfer data from an external EIS into the SCA module. An EIS export can be thought of as a subscription service listening to an external request from an EIS. The SCA component that utilizes the EIS export views it as a local service.

Purpose

The role of the EIS export is to bridge the gap between SCA components and external EISs. External services can be treated as an EIS export. In this case, the external service sends its data in the form of periodic notifications.

The EIS export provides SCA components with a uniform view of the services 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 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.

Administering EIS export bindings:

EIS export bindings are installed in WebSphere Process Server as part of your SCA applications. Administer the export from the administrative console.

You must have permission to make changes to the master configuration in order to perform this task.

You have an installed application which includes an EIS export module.

1. View the export 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 Exports and expand the export that you want to administer. You can choose to administer the interfaces or the bindings.
2. **Optional:** View the WSDL of the export 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 export binding. Expand Bindings and click on the binding that you want to administer. You can change the port or the name of the exported service.
4. Save changes to the configuration in order for you changes to take effect.

External clients with EIS bindings

WebSphere Process 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 WebSphere Process Server or needs to be invoked by a component from within WebSphere Process Server.

The client invokes the EIS Import as any other Service, 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, 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 Services. This interface can be thought of as the interface that external service like SAP or PeopleSoft will invoke (through the implementation of the EIS Export Service runtime). The export uses EISExportBinding to bind exported Services to the external EIS Service, it allows you to subscribe a Service 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. The JMS binding is a specialization of the EIS import and export bindings.

The services provided through a JMS binding allows a Service Component Architecture (SCA) module to make calls or receive messages from external systems. These systems can be an external JMS system or a WebSphere MQ based system.

The JMS service 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.

A user would choose to use the JMS provider for JCA 1.5 resource adapter when they want to integrate with an external system, other than WebSphere MQ. Using the SIB JMS Resource Adapter external services can receive messages and send messages to integrate with an SCA Component.

Related concepts

"SCA EIS Binding" on page 75
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.

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:

- A 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 service.
- 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 binding provide outbound connectivity from Service Component Architecture (SCA) applications to JMS providers.

The connection that is part of a JMS Import is made up of a connection factory. A connection factory is the object a client uses to create a connection to a provider. A connection factory encapsulates a set of connection configuration parameters that has been defined by an administrator. Each connection factory is an instance of the ConnectionFactory, QueueConnectionFactory, or TopicConnectionFactory interface.

The destination element of a JMS Import is a send message which initiates the destination.

Two types of JMS Import binding are supported:

- One-way - a message is sent to the destination specified as *send* in the import file. Nothing is sent to the *JMSReplyTo* field of the JMS header.
- Two-way, (or request-response)- a message is sent to the Destination specified as *send* in the import file. The Destination specified as *receive* is sent in the *JMSReplyTo* header property. 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.

JMS Export bindings

JMS Export binding provides inbound connectivity from JMS to the SCA System.

The connection that is part of a JMS Export is an activation specification.

A JMS Export has send, receive and callback 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 *JMSReplyTo* 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 service provides a reply. The Destination specified in the *JMSReplyTo* field of the incoming message overwrites the Destination specified in the *send*. The Destination specified to in the *callback* field of the export is used to store callback information. This is required for two-way or request/response operations on the interface for the JMS Service.

JMS message delivery and the failed event manager

JMS bindings are configured by default to have a maximum failed delivery of 5. JMS bindings require a value for the maximum failed delivery of at least 2 for the underlying SIB destination of the JMS destination. This is required for the receive destinations on both Import and Exports to enable the failed event manager to store failed messages. If you set a value for the maximum failed delivery of less than 2, messages to the target component that fail will not be available in the failed event manager. A failure in the target component constitutes a runtime exception. Failed messages that do not reach the failed event manager are routed to the exception destination as specified by the underlying SIB destination.

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 system other than WebSphere MQ. External services 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.

Related concepts

"MQ JMS bindings support" on page 96
The WebSphere MQ JMS service provides integration with WebSphere MQ based JMS providers.

Key features of a JMS Import binding

There are a number of key features of a JMS Import. These features include headers and created J2EE resources.

The JMS Import service 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 (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 (listens on) the destination sent with the request in the JMSReplyTo header field of the JMS message.

Special headers

Special header properties are used in an import to indicate to the target export binding information on how the message should be handled.

Parameters

A mapping from the native method to the operation method is provided through a special header.

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.

J2EE resources

A number of J2EE resources are created when a JMS Import is deployed to a J2EE environment.

Parameters

ConnectionFactory
Used by clients to create a connection to the JMS provider.

ActivationSpec
Used for receiving the response to a request.

Destinations

- Send destination - Where the request or outgoing message is sent.
- Receive destination - Where the response or incoming message should be placed.
- Callback destination - SCA JMS System destination used for correlation information. Do not read/write to this destination.

The installation task creates the ConnectionFactory and three destinations from the information in the import file. In addition, it creates the ActivationSpec to enable the runtime MDB to listen for replies on the receive destination. The properties of the ActivationSpec are derived from the destination/ConnectionFactory properties.

Configuring JMS Import bindings:

The JMS Import binding may 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, your JMS Import bindings are created in WebSphere Integration Developer. At development time you have an option to create the connections and destinations required for the JMS Import at install time, i.e., when the component is installed on your WebSphere Process Server. The alternative is to specify the JNDI name of the resources on the WebSphere Process Server that you intend your JMS Import to utilize.

Configuring the JMS Import 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 table.

Table 10. The names and JNDI names of resources created at the installation of a JMS Import binding on WebSphere Process 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

Note: The resources are created at the server scope. The default scope in the administrative console is node. You must change the scope in order to locate and administer the resources.

In the alternative case, where the JMS Import is expecting to find resources on the server that it will use, you must have these resources installed and the Import file must contain their JNDI names. The association between the JMS Import and the resources will then be made.

Administering JMS Import bindings:

You can administer JMS Imports bindings to tune, or set special features of the resource. The administrative tasks are performed using the WebSphere administrative console.

You must have permission to make and save changes to the profile on the administrative console.

The JMS Import must be installed as part of a Service Component Architecture (SCA) application in your WebSphere Process Server profile.

1. Open the default messaging provider settings panel in the administrative console.

   Expand JMS Providers, click Default Messaging (Resources > JMS Providers > Default Messaging).
2. Set the scope to Server. The resources your JMS Import uses are created at the Server level. Set the Scope to Server in order to view and administer these resources.

   Note: If you created the resources for the JMS Import, you might have done so at a different scope level, set the scope accordingly.
3. **Optional:** Administer JMS connection factories (Resources > JMS Providers > Default Messaging > JMS connection factory).

Click on JMS connection factory, in the list of connection factories. This panel shows a list of JMS connection factories with a summary of their configuration properties. Click on the JMS 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 JMS connection factory for use with the default messaging JMS provider. These configuration properties control how connections are created to associated JMS queues and topics.

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.

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 default messaging provider.

4. **Optional:** Administer JMS queue connection factories (Resources > JMS Providers > Default Messaging > JMS queue connection factory).

   Click on JMS queue connection factory, in the list of connection factories. This panel shows a list of JMS queue connection factories with a summary of their configuration properties. To administer a queue connection factory click on it, or click New to create a new one.

   Use the subsequent panel to browse or change the configuration properties of the selected JMS queue connection factory for use with the default messaging JMS provider. These configuration properties control how connections are created to associated JMS 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.

   A JMS queue connection factory is used to create connections to the associated JMS provider of JMS queues, for point-to-point messaging. Use queue connection factory administrative objects to manage JMS queue connection factories for the default messaging provider.

5. **Optional:** Administer JMS topic connection factories (Resources > JMS Providers > Default Messaging > JMS topic connection factory).

   Click on JMS topic connection factory, in the list of connection factories. This panel shows a list of JMS topic connection factories with a summary of their configuration properties. Click on the JMS topic 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 JMS topic connection factory for use with the default messaging JMS provider. These configuration properties control how connections are created to associated JMS topics.

   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.

   A JMS topic connection factory is used to create connections to the associated JMS provider of JMS topics, for publish/subscribe messaging. Use topic connection factory administrative objects to manage JMS topic connection factories for the default messaging provider.

6. **Optional:** Administer JMS queues (Resources > JMS Providers > Default Messaging > JMS queue).

   Click on JMS queue, in the list of destinations. This panel shows a list of JMS queues with a summary of their configuration properties. Click on the JMS queue that you want to administer, or click New to create a new JMS queue.

   Use the subsequent panel to change or enter settings for the 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 default messaging provider.
7. **Optional:** Administer JMS topics (Resources > JMS Providers > Default Messaging > JMS topic).

   Click on JMS topic, in the list of destinations. This panel shows a list of JMS topics with a summary of their configuration properties. Click on the JMS topic that you want to administer, or click New to create a new JMS topic.

   Use the subsequent panel to change or enter settings for the 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 default messaging provider.
8. **Optional:** Administer JMS activation specifications (Resources > JMS Providers > Default Messaging > JMS activation specification).

   Click on JMS activation specification, in the list of activation specifications. The panel shows a list of JMS activation specifications with a summary of their configuration properties.

   A JMS activation specification is associated with one or more message-driven beans (MDB) and provides the configuration necessary for them to receive messages. A JMS activation specification is used to provide configuration information when a JMS endpoint (message-driven bean application) is deployed and activated against the default messaging provider.

   Click on the JMS activation specification that you wish to administer or click New to create a new JMS activation specification.

Save your changes to the master profile and if necessary restart WebSphere Process Server.

Key features of a JMS Export binding

There are a number of key features of a JMS Export binding. These features include headers and created J2EE resources.

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 an export to indicate to the target export binding information on how the message should be handled.

Parameters

A special header is provided for the export binding to inform the export how to handle messages.

TargetFunctionName
This header is used on the export binding to map from native method to operation method. 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 a JMS Export is deployed to a J2EE environment.

Parameters

ConnectionFactory
Used by clients to create a connection to the JMS provider.

ActivationSpec
Configures the message endpoints that represent MDBs in their interactions with the messaging system.

Destinations

- Receive destination - The destination where the incoming or request message should be placed.
- Send destination - The destination where the response message will be sent, if not superseded by the JMSReplyTo header field in the incoming message.
- Callback destination - SCA JMS System destination used to store correlation information. Do not read from or write to this destination.

The installation task creates the set of resources similar to the one for JMS Import, an ActivationSpec, ConnectionFactory used for sending a reply and three destinations - send, receive and callback. All the properties of these resources are specified in the export file.

Configuring JMS export bindings:

The JMS export binding may 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, your JMS 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 export at install time, i.e., when the component is installed on your WebSphere Process Server. The alternative is to specify the JNDI name of the resources on the WebSphere Process Server that you intend your JMS export to utilize.

Configuring the JMS export 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 table.

Table 11. The names and JNDI names of resources created at the installation of a JMS Export binding on WebSphere Process 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 default scope in the administrative console is node. You must change the scope in order to locate and administer the resources.

In the alternative case, where the JMS export is expecting to find resources on the server that it will use, you must have these resources installed and the Export file must contain their JNDI names. The association between the JMS export and the resources will then be made.

Related tasks

"Working with external clients" on page 92
An external client (i.e., outside of WebSphere Process Server) may need to interact with an application installed in WebSphere Process Server.

"Enabling event sequencing for a JMS Export binding" on page 108
JMS Export bindings support event sequencing. In order for events to be handled in the order in which they are received you need to configure properties of the binding.

Administering JMS export bindings:

You can administer JMS Export bindings to tune, or set special features of the resource. The administrative tasks are performed using the WebSphere administrative console.

You must have permission to make and save changes to the profile on the administrative console.

The JMS export must be installed as part of a Service Component Architecture (SCA) application in your WebSphere Process Server profile.

1. Open the default messaging provider settings panel in the administrative console.

   Expand JMS Providers, click Default Messaging (Resources > JMS Providers > Default Messaging).
2. Set the scope to Server. The resources your JMS Export uses are created at the Server level. Set the Scope to Server in order to view and administer these resources.

   Note: If you created the resources for the JMS Export, you might have done so at a different scope level, set the scope accordingly.

3. **Optional:** Administer JMS connection factories (Resources > JMS Providers > Default Messaging > JMS connection factory).

   Click on JMS connection factory, in the list of connection factories. This panel shows a list of JMS connection factories with a summary of their configuration properties. Click on the JMS 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 JMS connection factory for use with the default messaging JMS provider. These configuration properties control how connections are created to associated JMS queues and topics.

   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.

   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 default messaging provider.
4. **Optional:** Administer JMS queue connection factories (Resources > JMS Providers > Default Messaging > JMS queue connection factory).

   Click on JMS queue connection factory, in the list of connection factories. This panel shows a list of JMS queue connection factories with a summary of their configuration properties. To administer a queue connection factory click on it, or click New to create a new one.

   Use the subsequent panel to browse or change the configuration properties of the selected JMS queue connection factory for use with the default messaging JMS provider. These configuration properties control how connections are created to associated JMS 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.

   A JMS queue connection factory is used to create connections to the associated JMS provider of JMS queues, for point-to-point messaging. Use queue connection factory administrative objects to manage JMS queue connection factories for the default messaging provider.
5. **Optional:** Administer JMS topic connection factories (Resources > JMS Providers > Default Messaging > JMS topic connection factory).

   Click on JMS topic connection factory, in the list of connection factories. This panel shows a list of JMS topic connection factories with a summary of their configuration properties. Click on the JMS topic 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 JMS topic connection factory for use with the default messaging JMS provider. These configuration properties control how connections are created to associated JMS topics.

   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.

   A JMS topic connection factory is used to create connections to the associated JMS provider of JMS topics, for publish/subscribe messaging. Use topic connection factory administrative objects to manage JMS topic connection factories for the default messaging provider.

6. **Optional:** Administer JMS queues (Resources > JMS Providers > Default Messaging > JMS queue).

   Click on JMS queue, in the list of destinations. This panel shows a list of JMS queues with a summary of their configuration properties. Click on the JMS queue that you want to administer, or click New to create a new JMS queue.

   Use the subsequent panel to change or enter settings for the 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 default messaging provider.

7. **Optional:** Administer JMS topics (Resources > JMS Providers > Default Messaging > JMS topic).

   Click on JMS topic, in the list of destinations. This panel shows a list of JMS topics with a summary of their configuration properties. Click on the JMS topic that you want to administer, or click New to create a new JMS topic.

   Use the subsequent panel to change or enter settings for the 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 default messaging provider.

8. **Optional:** Administer JMS activation specifications (Resources > JMS Providers > Default Messaging > JMS activation specification).

   Click on JMS activation specification, in the list of activation specifications. The panel shows a list of JMS activation specifications with a summary of their configuration properties.

   A JMS activation specification is associated with one or more message-driven beans (MDB) and provides the configuration necessary for them to receive messages. A JMS activation specification is used to provide configuration information when a JMS endpoint (message-driven bean application) is deployed and activated against the default messaging provider.

   Click on the JMS activation specification that you wish to administer or click New to create a new JMS activation specification.

Save your changes to the master profile and if necessary restart WebSphere Process Server.

External clients

WebSphere Process Server can send messages to, or receive messages from external clients using JMS bindings.

An external client, for example a web portal or an EIS needs to send a message to an SCA module in WebSphere Process Server or needs to be invoked by a component from within WebSphere Process 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 service 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 may or may not demand a response from the external client.

Working with external clients:

An external client (i.e., outside of WebSphere Process Server) may need to interact with an application installed in WebSphere Process Server.

To illustrate working with external clients we consider a very simple scenario in which an external client wishes to interact with a generic application on a WebSphere Process Server. The figure depicts a typical simple scenario.

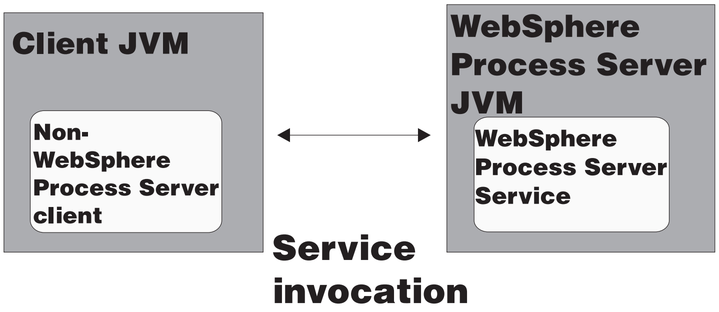

Client JVM
Non-WebSphere Process Server client
WebSphere Process Server JVM
WebSphere Process Server Service
Service invocation

Figure 7. A simple use-case scenario. An external client interacts with a WebSphere Process Server application.

The service is exported using a JMS binding, this makes the service available to external clients.

When you have an external client in a Java virtual machine (JVM) separate from WebSphere Process 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 look up the resources through JNDI. The client then using JMS 1.1 spec client to access the destinations and send and receive messages on the destinations. The default JNDI names of the resources created automatically by the runtime are listed in table xyz. If you have pre-created resources then use those JNDI names.

1. You must configure JMS destinations and connection factory to send the message.
2. You must have the correct JNDI context and port for the SIB Resource Adapter and the messaging bootstrapping port.

   WebSphere Process Server uses some default ports but if there are more servers installed in that machine, alternative ports are created at installation time to avoid conflicts with other server instances. You can determine which ports WebSphere Process Server is employing with the administrative console. Go to Servers > Application Servers > *your server name* > Configuration and click Ports under Communication. You can edit the port which is being used by clicking on Ports or details.
3. The client obtains an Initial Context with the correct values then look 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.

   The default JNDI names of the resources created automatically by the runtime are described in related topics.

   If you have pre-created resources then use the JNDI names of the resources that you created.

Example: request-response operation for an order processing application

The client source code for a request-response operation is shown below.

```
ConnectionFactory conFact;
Connection conn;
Session sess;
Destination requestQueue, responseQueue;
MessageProducer requestProducer;
MessageConsumer replyConsumer;
String jmsRequestQName = "jms/ExportIn";
String jmsReplyQName = "jms/ExportOut";
String jmsFactoryName = "jms/MyCF";
// the initial context factory to be used to get the JNDI initial context
String jndiFactoryName = "com.ibm.websphere.naming.WsnInitialContextFactory";
// URL for bootstrapping, containing ip address of WPS host machine
// and bootstrapping port that by default is 2809
// you should change it to the values of your host:port
String jmsContextURL = "iiop://9.26.237.246:2809/";
TextMessage jmsRequestMessage, jmsReplyMessage;
// this is the serialized business object "order",
// we have it already serialized as a String in order to avoid
// the extra complexity of serialization to this sample.
String message = " " String message = "<order:Order xsi:type=\"order:Order\"
xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
xmlns:order=\"http://OrderProcessing\"> "
+ "<orderId>123</orderId>"
+ "<customerId>456</customerId>"
+ "<orderDate>2002-01-01Z</orderDate>"
+ "<orderStatus>Reject</orderStatus>"
+ "<totalPrice>1000.0</totalPrice>"
+ "</order:Order>";

System.out.println("JMS Factory name:" + jmsFactoryName);
System.out.println("Queue Factory name:" + jmsRequestQName);
System.out.println("Initial Context Provider URL:" + jmsContextURL);
System.out.println();

try {
  Hashtable jndiHash = new Hashtable();
jndiHash.put(javax.naming.Context.INITIAL_CONTEXT_FACTORY,
jndiFactoryName);
 jndiHash.put(javax.naming.Context.PROVIDER_URL, jmsContextURL);
 javax.naming.Context ctx = new InitialContext(jndiHash);
 System.out.println("Got Initial Context from provider: " +
jmsContextURL);

 conFact = (ConnectionFactory) ctx.lookup(jmsFactoryName);
 if (conFact != null) {
  System.out.println("Got Connection Factory: " + jmsFactoryName);
  System.out.println(conFact.toString());
 }

 conn = conFact.createConnection();
 conn.start();
 System.out.println("Created Queue Connection...");
 System.out.println(conn.toString());

 sess = conn.createSession(true, Session.AUTO_ACKNOWLEDGE);
 System.out.println("created Session...");
 System.out.println(sess.toString());
 System.out.println(conn.toString());

 requestQueue = (Queue) ctx.lookup(jmsRequestQName);
 if (requestQueue != null) {
System.out.println("Found JMS request Destination " + jmsRequestQName);
  System.out.println(requestQueue.toString());
 }
 responseQueue = (Queue) ctx.lookup(jmsReplyQName);
 if (requestQueue != null) {
  System.out.println("Found JMS reply-to Queue " + jmsReplyQName);
 }

 requestProducer = sess.createProducer(requestQueue);
 System.out.println("Created Sender...");

 jmsRequestMessage = sess.createTextMessage();
 System.out.println("Creating Message...");
 jmsRequestMessage.setText(message);
 jmsRequestMessage.setJMSReplyTo(responseQueue);
 jmsRequestMessage.setStringProperty("TargetFunctionName",
"submitOrderRequestResponse");

 requestProducer.send(jmsRequestMessage);
 sess.commit();

 System.out.println("MESSAGE SENT!");
 System.out.println();
 System.out.println("JMS Correlation ID: ");
 String corrID = jmsRequestMessage.getJMSCorrelationID();
 if (corrID != null) {
  System.out.println(jmsRequestMessage.getJMSCorrelationID());
 }
 else {
  System.out.println("NULL");
 }
 System.out.println();
 System.out.println("JMS Message: ");
 System.out.println(jmsRequestMessage.toString());

 System.out.println();
 System.out.println("WAITING FOR THE RESPONSE...");
 System.out.println();

 responseQueue = (Queue) ctx.lookup(jmsReplyQName);
 if (responseQueue != null) {
  System.out.println("Found JMS response Destination " +
jmsReplyQName);
  System.out.println(responseQueue.toString());
 }
 else {
  System.out.println("JMS response Destination not found: " +
jmsReplyQName);
 }

 replyConsumer = sess.createConsumer(responseQueue);
 jmsReplyMessage= (TextMessage)replyConsumer.receive(10000);
 if (jmsReplyMessage != null) {
  System.out.println("MESSAGE RECEIVED!!");
  System.out.println();
  System.out.println("JMS Message: ");
  System.out.println(jmsRequestMessage.toString());
 }
 else {
  System.out.println("DIDN'T FIND MESSAGE IN THE REMOTE QUEUE: " +
jmsReplyQName);
 }
 sess.commit();
requestProducer.close();
 replyConsumer.close();
 sess.close();
 conn.close();

} catch (Exception j) {
 System.out.println("There was an exception, (for whole stack trace look
in SystemErr.log): " + j.getMessage());
 j.printStackTrace();

}
```

Related concepts

"Configuring JMS export bindings" on page 88
The JMS export binding may 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.

WebSphere MQ JMS bindings

The WebSphere MQ JMS service provides integration with WebSphere MQ based JMS providers.

Use the WebSphere MQ JMS Provider services when you want to integrate directly with a WebSphere MQ based system from your WebSphere Process Server environment. This eliminates the need to use MQ Link or Client Link features of the Service Integration Bus.

Importing services 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 services 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 serves the purpose of providing a mapping to the operation on the target component to be invoked.

WebSphere MQ JMS bindings: a general perspective

The WebSphere MQ JMS service provides integration with WebSphere MQ based JMS providers.

WebSphere MQ administrative tasks

The WebSphere MQ Administrator is expected to create the underlying Queue Manager and Queues in WebSphere MQ 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 based JMS providers. Please ensure that you are 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 JMS Import binding 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 JMS header.
- Two-way (or request-response) - a message is sent to the destination specified as *send* in the import file. The destination specified as *receive* is sent in the JMSReplyTo header field. The 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 *JMSCorrelationId* set to the sent message *JMSMessageID* 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.

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 service provides a reply. The destination specified in the JMSReplyTo field of the response message overwrites the destination specified in the send field.

WebSphere MQ JMS message delivery and the failed event manager

WebSphere MQ JMS bindings are configured by default to have a maximum retries value of 1. The WebSphere MQ JMS bindings require a value for the maximum retries of at least 1 for the underlying listener ports used for an incoming message. This is required for the receive destinations on both Import and Exports to enable the failed event manager to store failed messages. If you set the maximum retries value to less than 1, messages to the target component that fail will not be available in the failed event manager. A failure in the target component constitutes a runtime exception.

MQ JMS bindings support

The WebSphere MQ JMS service provides integration with WebSphere MQ based JMS providers.

Use the WebSphere MQ JMS Provider services when you want to integrate directly with a WebSphere 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 services 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 services 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 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 WebSphere Process 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

"JMS bindings support" on page 83
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.

Key features of a MQ JMS Import binding

There are a number of key features of a MQ JMS Import. These features include headers, J2EE artifacts and created J2EE resources.

Headers

A JMS message header contains a number of predefined fields that contain values that both clients and providers use to identify and to route messages.

Parameters

JMSCorrelationID
Link 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 Import binding provides various correlation schemes that are used to determine how to correlate request messages with response messages.

Parameters

The response message correlation schemes are:

RequestMsgIDToCorrelID
The response service is expected to copy the request JMSMessageID to the JMSCorrelationID field of the response message. This is the default setting.

RequestCorrelIDToCorrelID
The response service is expected to copy the request JMSCorrelationID to the JMSCorrelationID field of the response message.

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 - Where the request or outgoing message is sent.
- Receive destination - Where the response or incoming message should be placed.

Configuring MQ JMS Import bindings:

The MQ JMS Import binding may 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, your MQ JMS Import 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 Import at install time, i.e., when the component is installed on your WebSphere Process Server. The alternative is to specify the JNDI name of the resources on the WebSphere Process Server that you intend your MQ JMS import to utilize.

Configuring the MQ JMS Import 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 table.

Table 12. The names and JNDI names of resources created at the installation of a MQ JMS Import binding on WebSphere Process 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

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.

Note: 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 alternative case, where the MQ JMS import is expecting to find resources on the server 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.

Administering MQ JMS Import bindings:

You can administer MQ JMS Imports to tune, or set special features of the resource. The administrative tasks are performed using the WebSphere administrative console.

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.

The MQ JMS Import must be installed as part of a Service Component Architecture (SCA) application in your WebSphere Process Server profile.

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 WebSphere 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 JMS connection factory for use with WebSphere MQ as a JMS provider. These configuration properties control how connections are created to associated JMS queues and topics.

   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.

   The unified JMS connection factories configured in the WebSphere MQ messaging provider, allow for both point-to-point and publish/subscribe messaging.
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 JMS 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 JMS 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 JMS queue. Connections to the queue are created by the associated JMS queue connection factory for WebSphere MQ as a messaging provider.
5. **Optional:** Administer WebSphere MQ topic connection factories.

   Click on WebSphere MQ topic connection factories, in the list of additional properties. This panel shows a list of the WebSphere MQ topic connection factories with a summary of their configuration properties. Click on the WebSphere MQ topic connection factory that you want to administer, or click New to create a new connection factory.

   Use the subsequent panel to view or change the configuration properties of the selected topic connection factory for use with the WebSphere MQ as a JMS provider. These configuration properties control how connections are created to the associated JMS topic destination.

   Use WebSphere MQ topic connection factory administrative objects to manage JMS topic connection factories for the WebSphere MQ JMS provider.
6. **Optional:** Administer WebSphere MQ topic destinations.

   Click on WebSphere MQ topic destinations, in the list of additional properties. This panel shows a list of the WebSphere MQ topic connection factories with a summary of their configuration properties. Click on the topic destination that you want to administer, or click New to create a new topic destination.

   Use the subsequent panel to browse or change the configuration properties of the selected JMS topic destination for publish/subscribe messaging with WebSphere MQ as a messaging provider.

   A WebSphere MQ topic destination is used to configure the properties of a JMS topic for WebSphere MQ as a messaging provider. Connections to the topic are created by the associated topic connection factory.
7. Save your changes to the master profile and if necessary restart WebSphere Process Server.

Key features of a MQ JMS Export binding

There are a number of key features of a MQ JMS Export. These features include headers, J2EE artifacts and created J2EE resources.

Correlation schemes

The WebSphere MQ JMS Export binding provides various correlation schemes that are used to determine how to correlate request messages with response messages.

Parameters

The response message correlation schemes are:

RequestMsgIDToCorrelID
The WebSphere MQ JMS service will copy the request JMSMessageID to the JMSCorrelationID field of the response message to be sent. This is the default setting.

RequestCorrelIDToCorrelID
The WebSphere MQ JMS service will copy the request JMSCorrelationID to the JMSCorrelationID field of the response message.

J2EE resources

A number of J2EE resources are created when a MQ JMS Export 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

- Receive destination - The destination where the incoming or request message should be placed.
- Send destination - The destination where the response message will be sent, if not superseded by the JMSReplyTo header field in the incoming message.

Parameters

Configuring MQ JMS export bindings:

The MQ JMS export binding may 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, your MQ JMS export 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 export at install time, i.e., when the component is installed on your WebSphere Process Server. The alternative is to specify the JNDI name of the resources on the WebSphere Process Server that you intend your MQ JMS export to utilize.

Configuring the MQ JMS export 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 table.

Table 13. The names and JNDI names of resources created at the installation of a MQ JMS Export binding on WebSphere Process 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.

Note: 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 alternative case, where the MQ JMS export is expecting to find resources on the server 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 export and the resources will then be made.

Administering MQ JMS export bindings:

You can administer MQ JMS exports to tune, or set special features of the resource. The administrative tasks are performed using the WebSphere administrative console.

You must have permission to make and save changes to the profile on the administrative console.

The MQ JMS export must be installed as part of a Service Component Architecture (SCA) application in your WebSphere Process Server profile.

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 WebSphere 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 JMS connection factory for use with WebSphere MQ as a JMS provider. These configuration properties control how connections are created to associated JMS queues and topics.

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.

The unified JMS connection factories configured in the WebSphere MQ messaging provider, allow for both point-to-point and publish/subscribe messaging.

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 JMS 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 JMS 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 JMS queue. Connections to the queue are created by the associated JMS queue connection factory for WebSphere MQ as a messaging provider.

5. **Optional:** Administer WebSphere MQ topic connection factories.

   Click on WebSphere MQ topic connection factories, in the list of additional properties. This panel shows a list of the WebSphere MQ topic connection factories with a summary of their configuration properties. Click on the WebSphere MQ topic connection factory that you want to administer, or click New to create a new connection factory.

   Use the subsequent panel to view or change the configuration properties of the selected topic connection factory for use with the WebSphere MQ as a JMS provider. These configuration properties control how connections are created to the associated JMS topic destination.

Use WebSphere MQ topic connection factory administrative objects to manage JMS topic connection factories for the WebSphere MQ JMS provider.

6. **Optional:** Administer WebSphere MQ topic destinations.

   Click on WebSphere MQ topic destinations, in the list of additional properties. This panel shows a list of the WebSphere MQ topic connection factories with a summary of their configuration properties. Click on the topic destination that you want to administer, or click New to create a new topic destination.

   Use the subsequent panel to browse or change the configuration properties of the selected JMS topic destination for publish/subscribe messaging with WebSphere MQ as a messaging provider.

   A WebSphere MQ topic destination is used to configure the properties of a JMS topic for WebSphere MQ as a messaging provider. Connections to the topic are created by the associated topic connection factory.

Save your changes to the master profile and if necessary restart WebSphere Process Server.

External clients

WebSphere Process 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 WebSphere Process Server or needs to be invoked by a component from within WebSphere Process 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 service 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.

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.

Related reference

"Key features of an EIS import binding" on page 77
An EIS import is a Service Component Architecture (SCA) import that allows components in the SCA module to use EIS services defined outside the SCA module. An EIS import, is used to transfer data from the SCA component to an external EIS.

Data bindings for EIS service

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 DataObject, and is applicable to both, J2C and JMS.

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 record0based 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.

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 the JMS and MQ JMS Service

The data binding defined for the JMS and WebSphere MQ JMS Service has to implement the JMSDataBinding or JMSObjectBinding interface.

The JMSDataBinding interface for the JMS Service allows for the conversion of data between DataObject or Business Object and JMS Message.

Serialization:

The JMS and MQ JMS Service and Export implementation is able to provide default implementations for data bindings that provide serialization of the DataObjects to and from JMS or MQ JMS formats.

The JMS Service 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 may 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>
```

Working with data bindings in a runtime environment:

When an application is assembled, the Data Binding Generator generates DataBinding implementations for all types of all operations and stores their names in the import or export files, in the inDataBindingType or outDataBindingType attributes.

Data Bindings allow EIS Import or EIS Export implementation to convert argument data to the native format expected by the J2C or JMS implementation. The data binding classes implement various interfaces presented below.

The DataBinding is the root data binding interface. It allows for setting of a DataObject, and is applicable to both, J2C and JMS.

You can configure your own binding using pre-configured JMS resources. Pre define the connection factory and destination and then just point to them.

Export bindings and event sequencing

You can stipulate that your Exports process and deliver messages in the order in which they are received. For event sequencing to be possible you must configure your Export bindings appropriately.

Event sequencing provides the ability to sequence incoming events to an SCA component. EIS and JMS Exports 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 that those messages are received.

The specific settings required to permit event sequencing depend on the type of Export binding you are using.

In addition to EIS and JMS Exports, native WebSphere MQ Export bindings also support event sequencing. You must configure the native WebSphere MQ Export binding for event sequencing in WebSphere Integration Developer.

Related concepts

"Using event sequencing with Export bindings" on page 255
Event sequencing is supported with EIS Export bindings, JMS Export bindings, native 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.

Related information

WebSphere Integration Developer

Enabling event sequencing for an EIS Export binding

Event sequencing is supported for EIS Export bindings. The activation specification of the binding must be appropriately configured.

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 the specific provider documentation will need to be consulted for details of how to configure the adapter to enable the ordered or sequencing of events.

For more information on configuring the activation specification properties of WebSphere Adapters, see the topics on activation specification in the WebSphere Adapters information center.

For details of how to configure a clustered environment and whether event sequencing is supported for a particular resource adapter in a network deployment environment see the WebSphere Adapters information center.

Related concepts

WebSphere Adapters

"Using event sequencing with Export bindings" on page 255
Event sequencing is supported with EIS Export bindings, JMS Export bindings, native 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.

Enabling event sequencing for a JMS Export binding

JMS Export bindings support event sequencing. In order for events to be handled in the order in which they are received you need to configure properties of the binding.

Event sequencing for JMS Export bindings is not supported in a clustered environment.

To enable event sequencing for JMS Export bindings you must limit the number of concurrent message-driven beans (MDBs) that are processing incoming messages.

When using the SIB JMS Resource Adapter, set the maxConcurrency custom property on the ActivationSpec to a value of 1.

1. In the navigation pane of the administrative console, expand Resources and click Resource Adapters.
2. In the content pane, click SIB JMS Resource Adapter.
3. Under Additional properties, click J2C activation specification and select the Export activation spec. The name of the activation specification is defined when the Export binding is installed. See related topics for the naming convention.
4. Under Additional properties, click J2C activation specification custom properties
5. Set the maxConcurrency to 1. Click the value field and enter a value of 1, then click OK.
6. Save the changes to your master configuration.

Related concepts

"Configuring JMS export bindings" on page 88
The JMS export binding may 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.

"Using event sequencing with Export bindings" on page 255
Event sequencing is supported with EIS Export bindings, JMS Export bindings, native 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.

Enabling event sequencing for a WebSphere MQ JMS Export binding

WebSphere MQ JMS Export bindings support event sequencing. In order for events to be handled in the order in which they are received you need to configure properties of the binding.

Event sequencing for WebSphere MQ JMS Export bindings is not supported in a clustered environment.

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 (MDB).

1. Set the maxSessions property of your WebSphere MQ JMS Export binding to 1.
2. Save the changes to your master configuration.

   Related concepts

   "Using event sequencing with Export bindings" on page 255
   Event sequencing is supported with EIS Export bindings, JMS Export bindings, native 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.

JMS Headers

A JMS Header is a service data object (SDO) that contains all the properties of the 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 JMS Headers are provided using the system programming interface, consequently you set or get the JMS Headers by accessing the service component architecture Message. You cannot set or get the JMS header from a client program. You can however use an ESB Mediation to retrieve the JMS Header values or set them on an outgoing request.

JMS Header and JMS 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.

Handling faults in JMS and MQ JMS bindings

Faults can occur with JMS and MQ JMS bindings. Mechanisms are provided to handle these faults.

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 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.

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.

Troubleshooting JMS and MQ JMS bindings

You can diagnose and fix problems with your JMS or MQ JMS bindings.

In response to various error conditions, the JMS and MQ JMS Import and Export implementation can return one of the two types of exceptions:

- SCABusinessException – this exception is returned if exception specified on the service business interface (WSDL port type or Java interface) occurred.
- SCARuntimeException – 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 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 may 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

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

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.

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.

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

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.

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**.

Perform your administrative tasks.

Viewing display states

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

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

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.

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**.

Perform your administrative tasks.

Overview of 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.

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.

You must be at the administrative console to perform 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.

1. Locate the **Business Rules and Selectors Auditing** page. Navigate to this page by: **Servers > Application Severs >** *serverName* .
2. Select **Business Rules and Selectors Auditing** under **Business Integration - Additional Properties**.

3. Do one of the following depending on the type of change you want.

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

The audit log will now take the attributes you specified.

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.

These commands must be run from a command line environment for the server.

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 procedures shows 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. (**Servers > Applications servers >** *servername* **> Business Integration > Business Rules > Business Rules and Selectors Auditing > Runtime** or **Servers > Applications servers >** *servername* **> Business Integration > Selectors > Business Rules and Selector Auditing > Runtime**)

Note: The following procedure configures server *server1*. All of the steps beginning at step 2 could be placed in a jacl script and run that way.

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 and 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 SystemOut.log file. *Boolean* can either be true or false.

$AdminControl invoke $mbean getSeparateAuditLogFileName
Use to determine the filename 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.

If you want to save these changes, go to the administrative console (**Servers > Applications servers >** ***servername*** **> Business Integration > Business Rules > Business Rules and Selectors Auditing > Runtime** or **Servers > Applications servers >** ***servername*** **> Business Integration > Selectors > Business Rules and Selector Auditing > Runtime**) or enter $AdminConfig save.

Installing the business rules dynamic repository for a stand-alone server

When you run the installation and configuration script, the dynamic repository is installed in a stand-alone server..

If you select a database that does not support business rules, the system uses the default database, Cloudscape.

DB2 for z/OS version 7 limits the size of the primary key to 255 bytes. The dynamic artifact repository for business rule and selector artifacts uses the target name space/name/type to form the primary key. If you configured your system to use DB2 for z/OS version, you must limit the names as follows:

- target name space = 170 bytes
- maximum name = 75 bytes
- maximum type = 10 bytes (this is set by the runtime system)

Note: DB2 for z/OS version 8 does not share this limitation.

Installing the business rules dynamic repository for network deployment

Before installing any applications containing business rules, you must install the dynamic repository for business rules.

The dynamic repository supports a centralized configuration that allows you to configure all servers to use the same repository thus allowing all applications to use the same data. This is important for those users who will be updating this data dynamically using the business rule manager tool. The central repository allows:

- changes to be made once
- changes to take effect across all server installations

DB2 for z/OS version 7 limits the size of the primary key to 256 bytes. The dynamic artifact repository for business rule and selector artifacts uses the target name space/name/type to form the primary key. If you configured your system to use DB2 for z/OS version, you must limit the names as follows:

- target name space = 171 bytes
- maximum name = 75 bytes
- maximum type = 10 bytes (this is set by the runtime system)

Note: DB2 for z/OS version 8 does not share this limitation.

Considerations for modules containing business rules and selectors

This topic contains 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.

Considerations for database configuration

The dynamic artifact repository for business rule and selector artifacts uses the target name space/name/type to form the primary key. DB2 for z/OS version 7 limits the size of the primary key to 255 bytes.

If you configured your system to use DB2 for z/OS version, you must limit the names as follows:

- target name space = 170 bytes
- maximum name = 75 bytes
- maximum type = 10 bytes

Note: DB2 for z/OS version 8 does not share this limitation.

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. This task removes unneeded business rule and selector artifacts from the repository.

Make sure that you uninstall all copies of applications that use the business rules or selectors to be removed from all servers.

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 delete the unused artifacts from the database, you must do so manually after you uninstall all applications that use them. Remove artifacts using the tools supplied by the database platform of your repository.

1. Locate the database.

   Locating the database depends on the database platform.

Database platform	Location
Cloudscape	*WASHOME*\cloudscape\databases\RepositoryDB
Other databases	Depends on the location configured during installation and configuration of the server. For example, if you configured the server automatically and selected the default database name, the name of the database is WPSDB.

2. 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 installed applications that contain business rule and selector artifacts

3. Delete the artifacts for an application.

   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 the rows found. The primary key columns for the BYTESTORE table are **ARTIFACTTNS**, **ARTIFACTNAME**, and **ARTIFACTTYPE**.
c. Delete the rows found in step 3a on page 118 from the BYTESTORE table.
d. For each set of primary key values recorded in step 3b, find 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 rows found in step 3d from the BYTESTOREOVERFLOW table.
f. Delete the row in the APPTIMESTAMP table where the **APPNAME** column equals the name of the application.

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. (For more information about building and deploying business rules, see the WebSphere Integration Developer information center.)

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:
- 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

Create and modify business rules using an eclipse-based WebSphere Integration Developer tool. Manage and modify business rule values using the Web-based business rules manager tool.

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.

You must be at the administrative console for WebSphere Process Server to perform this task.

Perform this task to determine what business rule groups exist in your server.

1. From the administrative console, select **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.

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:

Exporting business rule components creates a file that you import into your development environment, thereby keeping the development artifacts synchronized with the actual production system artifacts.

Before starting this task, you should already have displayed your business rule components as described in "Displaying buinsess rule components."

Export business rule components when you have made changes to the business rule tables to synchronize your development environment with your production environment. This task begins with the business rule component display.

Tip: You can also export business rules using the command line. See "exportBusinessRuleArtifacts.jacl command."

1. Choose the business rule groups to export.

   Click the check boxes next to the business rule groups and then click **Export**. The browser displays a list of HTML links to the business rule groups you chose. (This is the Business rules to export panel.) Each business rule group has a file extension of `.zip`.
2. Download the files.

   Click on each filename and the system prompts you to save the file. When prompted, click **OK** to place the file in your file system.

   Note: If you choose to, you can rename the file as you download it.
3. Return to the business rules display panel.

   Click **Back** to return to the list of business rule groups.

The system saves file where you specified. You can then copy it to your test system.

You must import this file into your WebSphere Integration Developer environment. See the information center for WebSphere Integration Developer for more information.

Importing business rules using the administrative console:

Importing business rules allows you to update installed business rules without reinstalling an application.

You must be at the administrative console and have the location of a compressed file created by the export facility.

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.

Tip: You can also import business rules using the command line. See "importBusinessRuleArtifacts.jacl command."

1. Display the business rules on the server to which you are importing the business rules.
2. Click **Import**.
3. Specify the path to the file on the **Preparing for importing business rules** panel.

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 install after the initial installation of the server.

Business rules are designed and developed in WebSphere Integration Developer using if/then rule sets and decision tables to implement their operations. 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. These comprise rule books. 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

- Open a copy of a business rule from the repository
- Browse and edit a business rule
- Publish a business rule to the repository

Figure 1 shows how the business rules manager calls and publishes rules.

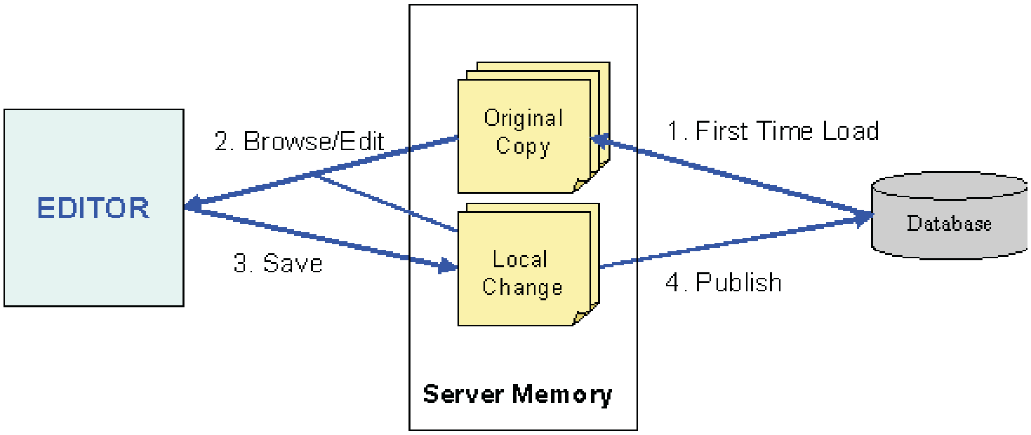


Figure 8. Business rules manager sequence of events

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 a rule from the database and stores it in the server memory as an original copy.
2. The rule is available for browsing or editing.
3. You save the business rule as a copy in the local server memory.
4. You publish the local copy back to the database. Alternatively, you can cancel the changes with no updates to the server being performed.

Installing the business rules manager:

The business rules manager is installed as an enterprise application for WebSphere Process Server, so that you can manage business rules during run time. You can install it using the administrative console or optionally, you can use a `.jacl` script for Windows, Linux, iSeries, or z/OS.

Security role required: You must be logged in as administrator or configurator to install the business rules manager.

1. To install the business rules manager from the administrative console, select the **Install business rules manager** check box on the **Business Rules Manager Configuration** page.
   a. From within the administrative console, expand **Servers** and click **Application Server**.
   b. Click the name of your server.
   c. On the **Configuration** page, under **Business Integration**, expand **Business Rules** and click **Business Rules Manager Configuration**.
   d. Under **General Properties**, select the **Install business rules manager** check box.
   e. In the **Context root** field, either accept the default context root of br/webclient or type a custom context root for the business rules manager URL.
   f. Click **OK**.
2. **Optional:** If you want to install the business rules manager manually using the command line (instead of using the administrative console), you must first make sure that WebSphere Process Server is started. Go to the*WPS_HOME*/bin

directory open the shell or command prompt for your operating system and run the specific installation command for your operating system.

- For Windows, run: `wsadmin.bat -f installBRManager.jacl [-s` *`servername`* `-n` *`nodename`* `| -cl` *`clustername`*`] -c` *`cellname`* `-r` *`rootname`* `-a` *`applicationname`*.
- For Linux, run: `./wsadmin.sh -f ./installBRManager.jacl [-s` *`servername`* `-n` *`nodename`* `| -cl` *`clustername`*`] -c` *`cellname`* `-r` *`rootname`* `-a` *`applicationname`*.
- For iSeries, run: `wsadmin -f ./installBRManager.jacl [-s` *`servername`* `-n` *`nodename`* `| -cl` *`clustername`*`] -c` *`cellname`* `-r` *`rootname`* `-a` *`applicationname`*.
- For z/OS, run: `wsadmin.sh -f ./installBRManager.jacl [-s` *`servername`* `-n` *`nodename`* `| -cl` *`clustername`*`] -c` *`cellname`* `-r` *`rootname`* `-a` *`applicationname`*.

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.

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 ″-c 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/webclient.″

applicationname
    The name of the application.

    The pair of arguments ″-a applicationname″ is optional. If missing, the default value of *applicationname* is ″BusinessRulesManager″.

Note: 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*

The other argument pairs are optional.

Note: Missing arguments will take the following default values:
- ″server1″ = *servername*
- ″br/webclient″ = *rootname*
- ″BusinessRulesManager″ = *applicationname*

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, the business rules database is used and shared between all servers in the same cell. The business rules database is a centrally governed storage for business rules information. When you access the business rules database, 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.

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.

1. Set security on user IDs.

   You must set global security when creating user IDs. Create user IDs and map them to the role `BusinessRuleUser` or `business analyst`. You must assign a role to each ID.
2. Set the session tracking mechanism to use cookies to track sessions.
3. At a minimum, set an appropriate session timeout value.
4. **Optional:** Set security on your server.

   If you have different roles or user IDs, you must set global security when configuring your server.

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.

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.

Make sure that both the server and client are configured correctly.

The default URLs for accessing the business rules manager are as follows. The URLs may vary according to the environment.

- https://hostname:9443/br/webclient (if security is enabled)
- http://hostname:9080/br/webclient (if security is not enabled)

where `hostname` is the name of the host.

Note: The Login page opens only when you have enabled global security on the server. If global security is not enabled, the Rule Books page will open when the business rules manager is accessed.

If global security is enabled, perform these steps to log in.

1. At the Login page, type your **User ID**.
2. Type your **Password**.
3. Click **Login**.

The initial page of the business rules manager will open with the existing rule books listed in the navigation pane.

You can now browse and edit business rule operations and templatize business rules.

Rule Books page and the business rules manager page layout:

When the business rules manager opens, the Rule Books page displays. It allows you to browse the business rule groups and their defined operations.

The Rule Books page is the first level of navigation. Its page layout includes many elements generic to other business rules manager pages.

Toolbar

The toolbar has the following content:

Welcome
Shows the user's name.

User identification
Provides the current user's name preceded with **Welcome User Name**.

Logout
Opens the Login page if global security is enabled.

Important: If you log out without publishing, a pop-up dialog appears to ask for confirmation.

Help Provides access to business rules topics in the WebSphere Process Server Information Center.

Navigation pane

The navigation pane is the left pane of the page. It provides access to the Publish and Revert page and the available rule books. 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.

Publish and Revert
Opens the Publish and Revert page where you can publish rule changes to the database or revert a rule to the original copy that was on the database.

Rule Books
Opens the Rule Book page, which is the top level of browsing. The rule books are listed in a navigation tree. You can expand or collapse a rule book by clicking either the plus (+) or minus (-) next to the rule book name to show all rules. When you select a rule book in the left pane navigation tree, all the child Rule 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 right pane of the page is the content area and is the main viewing and editing area. It includes 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 Rule Book page, Rule page, Rule Set page, or Decision Table page.

Title section

The title section includes the following information:

Path information

Provides the path to the page, such as the name of the rule book and the Rule page in the following format.

`BusinessRuleGroup01 > Table1_operation1`

Example: `CalculateDiscountBRG > CalculateDiscount`

Rule title

Includes the resource name and type of business rule in the following format:

`Ruleset112 - Ruleset`

Examples: `calculateDiscount-Rule Page`, `CalculateDiscountRS - Ruleset`

Function buttons

The function buttons that display on a page depend on the function of the particular page. The following table lists the possible function buttons for a page.

Table 14. Button list

Button Name	Function
Back	Returns to the previous page.
Edit	Enables editing of the Rule page, decision table, or rule set.
Save	Saves the changes to the local copy and goes back to the previous page. Note that the running server's state has not been changed. See "Publish" for how to change the server's state.
Cancel	Discards and 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 rule and then modify the values in order to make a new rule.
Publish	Publishes the Rule page to the server.
Revert	Cancels all changes to the rule that have been saved locally and reverts the rule to the original copy that resides in the database. Rules cannot be reverted after publishing.

Message 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

This general information section contains the following information.

Note: The Rule Books page does not include this section.

Last Published
Shows the last published date of the Rule page, rule set, or decision table.

Status Shows whether the Rule page, rule set, or decision table is in edit mode or has been published.

Description
Provides a brief description of the Rule page, rule set, or decision table.

Restriction: Do not use CDATA tags when editing the description fields for business rule group components and business rules in business rules manager as they make business rules 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 Rule Book page, Rule page, Rule Set page, or Decision Table page. For specific information for each of these pages, see the individual topics.

For the Rule Books page, the section includes the following information:

Business Rule Resources
Lists the names of the rule pages, rule sets, and decision tables.

Description
Provides either a brief description or name of the Rule page, rule set, and decision table.

Action
Shows the available actions for the corresponding business rule listing. It is initially empty; but when the group is expanded, an **Edit** button appears beside each rule.

Publish and Revert page:

The Publish and Revert page is for publishing locally saved changes for a rule to the database. It is also for reverting a rule back to the original copy that was on the database before it was saved locally.

The page-specific information section of the content area includes the following elements.

Changed Business Rules Resources section

Provides a list of business rules, rule sets, and decision tables available for publishing or reverting.

Business Rule Resources
Lists the names of the Rule pages, rule sets, and decision tables. Resources that are ready for publishing have a check box beside them.

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** push button in the **Action** column of the rule.

Rule Book page:

The Rule Book page lists all the business rule resources in the WebSphere Process Server system.

From the Rule Book page you can browse the business rule resources.

The page-specific information section of the content area includes the following elements.

Rule Book section

Provides a list of Rule pages, rule sets, and decision tables associated with the business rule group.

Business Rule Resources
Lists the names of the Rule pages, rule sets, and decision tables.

Description
Provides either a brief description or name of the Rule page, rule set, and decision table.

Action
Shows the available actions for the corresponding business rule listing. It is initially empty; but when the group is expanded, an **Edit** button appears beside each rule.

Rule page:

The Rule page provides an interface for modifying the values of a business rule in the rule logic selection record. The information is displayed in table format.

From the Rule 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.

Rule Logic Selection Records section

Provides a list of effective business rules that are the building blocks of that rule and enables working with business rule records, such as adding and sorting selection records.

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 business rule records.

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 selection records.

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.

Available Rule Logics

Provides a list of rule sets or decision tables that can apply to a particular business rule, with their associated descriptions, and actions 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 description, save the rule set as a working copy, or delete a rule.

The page-specific information section of the content area includes the following elements.

Rules section

Provides a list of associated rules with the following information:

Name Provides the name of the rule.

Rule Lists the variables, constraints, range, and enumeration that defines the rule.

Action
Enables reordering and deleting rules.

Templates section

Facilitates creating a new rule using an existing template. 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 the name of the rule.

Rule Provides a text area for specifying the variables, constraints, range, and enumeration that defines the rule.

Action
Enables adding the rule to the template.

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 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 descriptions, and save a decision table as a working copy.

The page-specific information section of the content area includes the following elements.

Decision Table 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.

Initialization Rule
Shows the initialization rule of this decision table. 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. The initialization rule displays only if the business rule definition was designed in WebSphere Integration Developer with an initialization action.

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 the decision table definition was designed in WebSphere Integration Developer with an *otherwise* condition specified. You cannot add or remove the *otherwise* condition column from a decision table dynamically from the business rules manager.

Templates section

Facilitates creating a new rule using an existing template.

Working with business rule records:

A business rule record identifies information for a rule, such as its effective dates and the if/then rule set or decision table associated with the rule.

Use the business rules manager to create, modify or delete business rule records.

Creating a business rule record:

You create business rule records from existing records.

You need to be in the **Edit** mode for the rule you want to create.

To create a new business rule record, perform the following steps:

1. On the Rule page click **Add Selection Record**.

   A new business rule record 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 a default business rule has been specified, it is used during the gap. As a best practice, you should always specify a default rule.
4. Select the **Effective Rule Logic** from the drop-down list.
5. Click **Save** .

A message displays in the message field indicating that the record has been temporarily saved and that you can publish the changes from the Publish and Revert page.

Related tasks

"Deleting a business rule record" on page 140
You can delete existing business rule selection records from the rule selection logic table. When a selection record is deleted, the associated rule set or decision table definition remains with the rule group and is listed in the Available Rule Logics section of the page. The rule can be added back either as the default rule or with a specific date and time.

Modifying a business rule record:

You can modify the date and time values of existing business rule selection records.

You need to be in the **Edit** mode for the rule you want to modify.

To modify a business rule, perform the following steps:

1. On the Rule page edit the **Start Date/Time** of the selection record:
   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 selection record:
   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 a default business rule has been specified, it is used during the gap. As a best practice, you should always specify a default rule.
3. Click **Save**.

   Note: If the **Date/Time** fields are invalid, the fields will turn **red** and a message will display in the message field indicating that the dates/time field values are invalid.

The record is saved locally and is ready to be published to the database. For more information, see “Publishing and reverting business rules” on page 141.

For more information on setting business rule dates, see “Splitting dates in business rules” on page 134.

Related tasks

“Deleting a business rule record” on page 140
You can delete existing business rule selection records from the rule selection logic table. When a selection record is deleted, the associated rule set or decision table definition remains with the rule group and is listed in the Available Rule Logics section of the page. The rule can be added back either as the default rule 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 business rule implementation per operation can be effective at any point in time. When no other business rule is found to be in effect for any point in time, the default business rule 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 selection record. The continuous date selection is only available on the **End Date/Time** field.

Note: The continuous selection is used when date ranges of two business rule selection records are contiguous. A continuous attribute is set to the end date of the first selection record. When this attribute is set, the start date of the second selection record is set to the end date of the first selection record 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.

You need to be in the **Edit** mode for the rule you want to modify.

To split a business rule record, perform the following steps:

1. Click **Split** next to the record.

   A new record is created with a start date of Jan 1; and its fields are in red. A message displays in the **Message** field indicating that the date/time field values are invalid.
2. Select the start date/time for the new record.

   The end date/time for the original record changes from *continuous* to the start date/time of the new record, and the end date/time of the new record changes to the end date/time of the previous record.
3. Modify the date/times of the new record.
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 a rule set record:

You create a new rule set record by copying an existing rule set record and modifying its values.

To create a new rule set record, perform the following steps:

1. Click **Copy** next to the rule logic record for the selected rule set.

   The **Edit** page opens for the new record, with a title Edit Mode:Copy_of_TableName-Ruleset.
2. In the **Name** field enter a name for the new rule set record.
3. In the **Description field** enter a short description of the new record.
4. 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.
5. Click the up or down arrow to place the rule in the correct sequence.
6. Click **Save**.

A message displays in the message field indicating that the record has been temporarily saved and that you can publish the changes from the Publish and Revert page.

Related tasks

"Deleting a business rule record" on page 140
You can delete existing business rule selection records from the rule selection logic table. When a selection record is deleted, the associated rule set or decision table definition remains with the rule group and is listed in the Available Rule Logics section of the page. The rule can be added back either as the default rule or with a specific date and time.

Creating a rule within a rule set from a template:

You create a new rule within a rule set using the rule templates associated with that rule set.

You need to be in the edit mode for the rule set.

To create a new rule from a template, perform the following steps:

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. Specify the values for the rule in the input fields or select the variables from the drop-down lists.
   c. Click **Add**.
3. 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.
4. Click **Save**.

The rule set is ready for publishing. For more information, see "Publishing and reverting business rules" on page 141.

Related tasks

"Deleting a business rule record" on page 140
You can delete existing business rule selection records from the rule selection logic table. When a selection record is deleted, the associated rule set or decision table definition remains with the rule group and is listed in the Available Rule Logics section of the page. The rule can be added back either as the default rule or with a specific date and time.

Modifying a rule within a rule set using a template:

You modify a rule in a rule set using templates associated with that rule set.

You need to be in the edit mode for the rule set.

To modify a rule using an existing template, perform the following steps:

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**.

The modified rule set is ready for publishing. For more information, see "Publishing and reverting business rules" on page 141

Related tasks

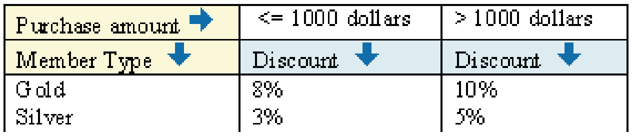
"Deleting a business rule record" on page 140
You can delete existing business rule selection records from the rule selection logic table. When a selection record is deleted, the associated rule set or decision table definition remains with the rule group and is listed in the Available Rule Logics section of the page. The rule can be added back either as the default rule or with a specific date and time.

Decision tables:

A decision table is a business rule record, 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.

The following figure shows an example of a decision table with the *conditions* Purchase amount along the top row and Member Type along the left column. The *action* Discount is in the middle and right columns. The orientation of conditions and actions is shown by arrows. The example shows that if a gold customer spends more than $1000, then the customer gets a 10% discount. The action is the intersection of the purchase amount condition and the member type condition.



Purchase amount →	<= 1000 dollars	> 1000 dollars
Member Type ↓	Discount ↓	Discount ↓
Gold	8%	10%
Silver	3%	5%

Figure 9. Decision table

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 templatize actions associated with the *otherwise* condition. The *otherwise* condition can be used zero or one time for any condition being checked.

Creating a decision table record:

You create a new decision table record by copying an existing decision table record and modifying its values.

To create a decision table record, perform the following steps:

1. Click **Copy** next to the selected rule logic record for the selected decision table.

The Edit page opens for the new record, with a title Edit Mode:Copy_of_TableName-Decision Table.

2. In the **Name** field enter a name for the new decision table record.
3. In the **Description field** enter a short description of the new record.
4. 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.
5. Click **Save**.

A message appears in the message field indicating that the record 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 a business rule record" on page 140
You can delete existing business rule selection records from the rule selection logic table. When a selection record is deleted, the associated rule set or decision table definition remains with the rule group and is listed in the Available Rule Logics section of the page. The rule can be added back either as the default rule 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 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

Menu option	Description	Modifies condition	Modifies action
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	
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 a decision table record:

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.

You need to be in the edit mode for the decision table you want to modify.

To modify the values of a decision table, perform the following steps:

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 templatize the actions associated with 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**.

The rule is modified locally and is ready to be published to the server. For more information, see "Publishing and reverting business rules" on page 141.

Related tasks

"Deleting a business rule record"
You can delete existing business rule selection records from the rule selection logic table. When a selection record is deleted, the associated rule set or decision table definition remains with the rule group and is listed in the Available Rule Logics section of the page. The rule can be added back either as the default rule or with a specific date and time.

Modifying a template value of a decision table:

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.

You need to be in the edit mode for the decision table you want to modify.

To modify a decision table template, perform the following steps:

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**.

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 a business rule record:

You can delete existing business rule selection records from the rule selection logic table. When a selection record is deleted, the associated rule set or decision table definition remains with the rule group and is listed in the Available Rule Logics section of the page. The rule can be added back either as the default rule or with a specific date and time.

You need to be in the edit mode for the rule you want to delete.

To delete a business rule record, perform the following steps:

1. On the Rule page, select the rule; and click **Delete**.

   The selection record is deleted. The associated rule set or decision table definition remains with the rule group and is listed in the Available Rule Logics section of the page. The rule can be added back either as the default rule or with a specific date and time.

Note: Each operation on a business rule group must have at least one active business rule associated with it, either as a selection record or as a default destination. Attempting to delete all business rule records will result in an error.

2. Click **Save**.

The record is temporarily saved and is ready for publishing to the server.

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 database that resides on the application server, you need to *publish* the rule changes. Alternatively, you can cancel the changes that have been saved locally to a business rule by *reverting* the rule to its original state.

You need to be on any business rules manager page that has a navigation pane.

The server publishes changes at the Rule page level.

To publish the changes to a business rule, perform the following steps:

1. Click **Publish and Revert**.
2. On the Publish and Revert page, select the pages to send to the database by clicking their check boxes in the left-hand column of the content area. You can publish all the Rule pages together as a single transaction, or just a subset of the pages.

   Note: To cancel all changes that have been saved locally to a business rule and replace the changed rule with the original copy of the rule that resides on the database, select the check box for the Rule page and click **Revert**. Business rules cannot be reverted after publishing since publishing changes the original copy that resides on the database.
3. Click **Publish**.

   The selected Rule pages do not display when you refresh the Publish and Revert page, signifying that they have been written to the database.

The business rule is ready to be exported to the server.

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.

Login error:

A log in error occurs upon logging in.

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:

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**.

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.

Login conflict error:

A login conflict error occurs when another user with the same user ID is already logged in to the application.

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.

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.

Access conflict errors:

An access conflict error occurs when a business rule is updated in the database by one user at the same time another user is updating the same rule.

This error is reported when you publish your local changes to the database.

To correct access conflict errors, perform the following actions:

- Publish the Rule page.
- 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 rule manager, you must reload those Rule pages in error from the database as your local changes of Rule pages in error are no longer usable. You can still use local changes in other Rule pages that are not in error.
- Reload a Rule page, by clicking **Reload** in the Publish and Revert page of the rule for which the error was reported.

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 comprises an interface and a selector table. The selector table determines, based on a criteria, 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 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 panels, you must supply a userid when you log in to the administrative console. If you are logged in without a userid, you will receive a warning to log out and log back in with a valid userid.

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.

You must be at the administrative console for the WebSphere Process Server to perform this task.

Perform this task to determine what selector components exist in your server.

Restriction: To access any of the selector component panels, you must supply a userid when you log in to the administrative console. If you are logged in without a userid, you will receive a warning to log out and log back in with a valid userid.

1. Click on **Servers** to display the different server types.
2. Click on **Application servers** to expand the Application server list.
3. Click *servername* to select the server from the list server list which has the selectors to display.
4. Click **Selectors** under **Business Integration**.

   The console displays a list of all the selector components defined with each components' descriptions.

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.

You must be at the administrative console for the WebSphere Process Server to perform this task.

To perform other selector table-related tasks, display a selector table when you are determining the entries that comprise the table. This task begins after you display selector components.

1. Click on the selector component from the Selector Components display. The browser displays the selector tables in the selected component.
2. Click on one of the selector tables in the display. The browser displays 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.

To perform this task, a selector table must exist.

Change a target component to alter the selection criteria or use a different target component for an entry in the selector table.

Important: Before changing target components for long-running applications, stop the application before proceeding. Do not change target components while a long-running application is processing.

1. Display the selector table as described in "Displaying selector tables."
2. Click the **Target ID** next to the target component to change. The system displays the Target component details panel.
3. Change the entry.

Portion of entry to change	Steps to change
Target destination	1. Click arrow next to the target component list. The system displays 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 be displayed according to the locale format. For the US English locale the format displayed is: • 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 arrow next to the target component list. The system displays 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 be displayed according to the locale format. For the US English locale the format displayed is: • 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.

4. **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.
5. Click **Apply** or **OK**.

   To continue working in this display, click **Apply**. To return to the target component display, click **OK**.
6. Click **Save** on the target component display to save the changes to the selector table.

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.

To perform this task, a selector table must exist.

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.

1. Display the selector components as described in "Displaying selector components."
2. Display the selector table as described in "Displaying selector tables".
3. Click **New** on the selector table display. The browser displays a pre-filled target component detail panel.
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 component display.

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.

To perform this task, a selector table must exist.

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.

1. Display the target components as described in "Displaying selector tables.". The panel displays the selector table display.
2. Click the check box next to the target components to delete, then click **Delete**. The system updates the panel by displaying the remaining target components.
3. Click **Save**. The system saves the updated selector table with the entries representing the remaining target components.

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:

Exporting selector components creates a file that you can then import into your development or other production environments, thereby keeping the development artifacts synchronized with the actual production system artifacts.

Before starting this task, you should already have displayed your selector components as described in "Displaying selector components.".

Export selector components when you have made changes to the selector tables and you need to synchronize your development environment with your production environment. This task begins with the selector component display.

1. Choose the selectors to export.

   Click the check boxes next to the selectors and then click **Export**. The browser displays a list of HTML links to the selector components you chose. (This is the **Selectors to Export** panel.) Each selector has a file extension of zip.

2. Download the files.

   Click on each filename and the system prompts you to save the file. When prompted, click **OK** to place the file in your file system.

   Note: If you choose to, you can rename the file as you download it.
3. Return to the selectors display.

   Click **Back** to return to the list of selectors.

The system saves file where you specified.

Importing selector components using the administrative console:

Importing selectors components allows you to update installed selector components without reinstalling an application.

You must be at the administrative console and have the location of a compressed file created by the export facility.

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.

Tip: You can also import selector components using the command line.
1. Display the selectors on the server to which you are importing the selector components as described in "Displaying selector components."
2. Click **Import**.
3. Specify the path to the file on the **Preparing for importing selectors** panel.

Display the selector tables for the updated selectors to verify the changes.

Overview of targets

Targets provide additional flexibility by providing the capability of modifying processing by changing the target configured for a reference.

A component can call a component in another module thereby reusing existing logic to minimize the time and cost in building an application. WebSphere Process Server provides additional flexibility through targets. Targets allow an installed application to benefit from advances in processing by allowing the application to change the endpoint of a cross-module invocation, using the administrative console, 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 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

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.

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.

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.

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.

Delete J2C activation specifications when you inadvertently saved a configuration after installing an application that contains services and do not require the specifications.

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 stand alone server or at the **server** scope in a Network 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.**

The system removes selected specifications from the display.

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.

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.

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.

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.**

The panel displays only the remaining destinations.

Delete the J2C activation specifications related to the module that created these destinations.

Administering extended messaging resources: Overview

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.

The administrative console interface enables you to configure the resources needed by the extended messaging service and by the applications that use the 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: The Extended Messaging Service feature is being deprecated in WebSphere Process Server 6.0.1. Although you can continue to use extended messaging with new and existing applications in this release, you will need to replace these applications with ones that use standard JMS APIs, or replace them with equivalent messaging technologies.

Enabling the extended messaging service

The Extended Messaging Service provides runtime service to support container-managed messaging (extended messaging). The service can be automatically started when the application server is started, or it can be started manually.

Security role required: You must be logged in as administrator or configurator to complete this task.

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. Stop and restart the application server in order for the changes to take effect.

You can also use the Extended Messaging Service page to configure a listener port extension to handle late responses, as described in “Configuring listener port extensions to handle late responses” on page 152.

Configuring listener port extensions to handle late responses

An application's listener port can be configured with an extension to handle late responses in an extended messaging environment.

Security role required: You must be logged in as administrator or configurator to complete this task.

1. Ensure you have a listener port defined and configured.
2. From the administrative console, click **Servers > Application servers >** *server_name* **> Extended Messaging Service > Listener Port Extensions**. The Listener Port Extensions page opens.
3. Click **New** to create a new listener port extension. The New Listener Port Extension page opens.
4. Select the **Enabled** check box to enable late response handling.
5. In the **Request Interval** field, either accept the default value or specify a new value. The request interval specifies 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. The request timeout value specifies how long the listener port waits for late responses. Responses received after that time are discarded.
7. Use the **Listener Ports** drop-down menu to specify the listener port you want 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.

Managing extended messaging providers

An extended messaging provider manages the resources defined for use with container-managed messaging (extended messaging). Use the Extended Messaging Provider page in the administrative console to view the general properties for each resource and to select a resource for editing.

Extended messaging resources can be defined at a cell, node, or server scope:

- Cell scope—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 scope—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 scope—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 for z/OS Information Center.

Security role required: You must be logged in as administrator, operator, configurator, or monitor to complete this task.

1. From the administrative console, click **Resources > Extended Messaging Provider**. The Extended Messaging Provider page opens.
2. Click the radio button next to the appropriate scope.

3. Click **Apply**. The **Scope**, **Name**, and **Description** fields on the bottom of the page are updated to reflect the values for the selected resource provider.

You can now create, modify or delete input ports, output ports, or other custom properties for the selected extended messaging provider.

Related tasks

"Adding a new input port"

"Adding a new output port" on page 155

Adding a new input port:

A receiver bean constructed from a session bean requires an input port to define the properties for the receiving Java Message Service (JMS) destination. The input port can also provide details for selecting and handling messages and for the reply destination (when required).

Note: Receiver beans that are constructed from message-driven beans do not require an input port; the details they contain are associated with the deployed message-driven bean and the Message Listener Service.

During this task, you configure the initial properties of the input port. You can later change the properties of the port as needed.

Security role required: You must be logged in as administrator or configurator to complete this task.

1. From the administrative console, click **Resources > Extended Messaging Provider**. The Extended Messaging Provider page opens.
2. Select the scope for the resource provider to which you want to add the new port.
3. Click **Apply**.
4. Click **Input Ports** from the Additional Properties table. The Input Port page opens.
5. Click **New**. The Input Port configuration page opens.
6. Specify the appropriate properties for the new input port. See "Input port settings" on page 154 for detailed information about property settings.
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.

Related tasks

"Modifying the configuration of an input port"

Modifying the configuration of an input port:

You can modify the configuration of an existing input port as needed. Input ports define the properties for the receiving Java Message Service (JMS) destination. They also provide details for selecting and handling messages and for the reply destination (when required).

Security role required: You must be logged in as administrator or configurator to complete this task.

1. From the administrative console, click **Resources > Extended Messaging Provider**. The Extended Messaging Provider page opens.
2. Select the scope for the resource provider to which you want to add the new port.
3. Click **Apply**.
4. Click **Input Ports** from the Additional Properties table. The Input Port page opens.
5. Select the input port that you want to modify. The Input Port configuration page opens. It displays the current configuration properties for that port.
6. Modify the properties for the input port. See "Input port settings" for detailed information about property settings.
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: 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.

Name The name of the input port, used for administrative purposes. The value for this field must be a string.

JNDI Name
The Java Naming and Directory Interface (JNDI) name for the resource. The value for this field must be a string.

Description
A description of the input port, used for administrative purposes. The value for this field must be a string.

This field is optional.

Category
A category string to use when classifying or grouping the resource. The value for this field must a a string between 1 and 30 ASCII characters in length.

This field is optional.

JMS Connection Factory JNDI Name
The JNDI name for the Java Message Service (JMS) connection factory used by the input port (for example, jms/connFactory1). The value for this field must be a string.

JMS Destination JNDI Name
The JNDI name for the JMS destination used by the input port (for example, jms/destn1). The value for this field must be a string.

JMS Acknowledgement Mode
The JMS mode that is used to acknowledge messages. This property is used only for message-driven beans that use bean-managed transaction demarcation (in other words, the transaction type is set to `Bean`).

The valid values for this field are as follows:

- Auto Acknowledge—The session automatically acknowledges a message in either of the following cases:

- When the session has successfully returned from a call to receive a message
- When the session has called a message listener to process the message and received 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. The valid values for this field are as follows:

- 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. The valid values for this field are as follows:

- 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 Durable.

This field is required only if the JMS destination type is a topic.

Reply JMS Connection Factory JNDI Name

The JNDI name of the JMS connection factory that is used for replies. The value for this field must be a string.

Reply JMS Destination JNDI Name

The JNDI name of the JMS destination that is used for replies. The value for this field must be a string.

Adding a new output port:

An output port specifies the properties needed by sender beans to define the destination for the message being sent. It also specifies optional properties if a response is expected. The output port is associated with the sender bean at deployment time.

During this task, you are configuring the initial properties of the output port. You can modify these properties later, as needed.

Security role required: You must be logged in as administrator or configurator to complete this task.

1. From the administrative console, click **Resources > Extended Messaging Provider**. The Extended Messaging Provider page opens.

2. Select the scope for the resource provider to which you want to add the new port.
3. Click **Apply**.
4. Click **Output Ports** from the Additional Properties table. The Output Port page opens.
5. Click **New**. The Output Port configuration page opens.
6. Specify the appropriate properties for the new output port. See "Output port settings" for detailed information about property settings.
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.

   Related tasks

   "Modifying the configuration of an output port"

Modifying the configuration of an output port:

You can modify the configuration of an existing output port as needed. Output ports define the destination for the message being sent and provide the destination if a response is expected.

Security role required: You must be logged in as administrator or configurator to complete this task.

1. From the administrative console, click **Resources > Extended Messaging Provider**. The Extended Messaging Provider page opens.
2. Select the scope for the resource provider to which you want to add the new port.
3. Click **Apply**.
4. Click **Output Ports** from the Additional Properties table. The Output Port page opens.
5. Select the output port that you want to modify. T he Output Port configuration page opens. It displays the current configuration properties for that port.
6. Modify the properties for the output port. See "Output port settings" for detailed information about property settings.
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: An input port has the following configuration properties:

Scope This field specifies the extended messaging provider scope. The value represents the location of the configuration file.

This field cannot be edited.

Name The name of the output port, used for administrative purposes. The value for this field must be a string.

JNDI Name
The Java Naming and Directory Interface (JNDI) name for the output port. The value for this field must be a string.

Description

A description of the output port, used for administrative purposes. The value for this field must be a string.

This field is optional.

Category

A category string to use when classifying or grouping the resource. The value for this field must a a string with a maximum of 30 ASCII characters.

This field is optional.

JMS Connection Factory JNDI Name

The JNDI name for the Java Message Service (JMS) connection factory used by the output port (for example, jms/connFactory1). The value for this field must be a string.

JMS Destination JNDI Name

The JNDI name for the JMS destination used by the output port. The value for this field must be a string (for example, jms/destn1).

JMS Delivery Mode

The JMS mode that is used to deliver messages. The value must be one of the following:

- 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. The value must be an integer 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.

The value must be an integer 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. The value must be one of the following:

- 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 Disable Message Time Stamp

Specifies whether the system generates a JMS message timestamp. The value must be one of the following:

- Selected—Message timestamps are not added to sent messages
- Cleared—Message timestamps are automatically added to sent messages

By default, message timestamps are added.

Response JMS Connection Factory JNDI Name
The JNDI name of the JMS connection factory that is used for responses handled by the output port (for example, jms/connFactory1). The value for this field must be a string.

Response JMS Destination JNDI Name
The JNDI name of the JMS destination that is used for responses handled by the output port (for example, jms/destn1). The value for this field must be a string.

Administering Enterprise Service Bus

Administering Enterprise Service Bus involves managing a service bus environment, deploying and managing mediation modules, managing resources used for service integration and mediation modules, and managing the clients and adapters that interact with the mediation modules. Administrators can also administer the full range of features of the underlying WebSphere Application Server.

With Enterprise Service Bus, administrators create an environment of ESB servers and service integration buses that support the deployment of mediation modules as service applications. When you install WebSphere Process Server, you get two service integration buses to use for service applications deployed into the ESB. You can start with one server in a bus and optionally add capacity and enhanced availability by build up to multiple servers or server clusters. You can add other buses if you need, to provide separate ESBs, to deploy applications that connect to an ESB, or to enable integration with WebSphere MQ.

Administrators can deploy mediation modules (as SCA modules) into the server and bus environment. They can view the modules that you have deployed, and monitor that requests are being processed correctly. Administrators can start or stop mediation modules, and can administer modules in other ways; for example, to change the configuration of a module, to stop or update the module, and otherwise manage its activity. Administrators can also make changes to the SCA imports of a mediation module; for example, to redirect one module to another module. This allows a module to invoke different service providers or process service requests and responses in different ways without having to rebuild and redeploy the module.

Administrators can use a variety of tools to administer Enterprise Service Bus, including the WebSphere administrative console, the WebSphere administrative (wsadmin) scripting program, command-line tools, and administrative programs.

The main descriptions of administrative tasks are based on use of the administrative console. Each task in the administrative console is supported by one or more panels. You can use a task filter to display the set of panels that are most appropriate to the tasks that you want to complete, and thereby focus your activities on only those panels.

All This displays all administrative console panels. This is most appropriate for an administrator interested in managing all parts of WebSphere Process Server and the underlying WebSphere Application Server.

For more information about administration of enterprise service bus features, see the related tasks listed below.

Application Integration
This displays panels suitable for the following task areas:

- Adjusting the configuration of service integration buses, servers, server clusters, messaging engines and network topologies needed to support the deployment of mediation modules and service applications
- Creating resources (for example, JMS connection factories and Common Event Infrastructure profiles) needed by deployed service applications and mediation modules
- Operational control of mediation modules and service applications

This is most appropriate for an administrator interested in deploying and managing mediation modules as service applications, as described in "Managing service applications" on page 201.

Server and Bus

This displays panels suitable for the following task areas:

- Defining service integration buses, servers, server clusters, messaging engines and network topologies needed to support the deployment of mediation modules and service applications
- Enabling and disabling infrastructure services
- Installing applications and mediation modules
- Creating resources (for example, JMS connection factories and Common Event Infrastructure profiles) needed by deployed service applications and mediation modules
- Operational control of the bus and server environment

This is most appropriate for an administrator interested in managing the server and bus environment needed to support the deployment of service applications and mediation modules. This includes defining the network and bus topology, defining appropriate resources and monitoring the runtime system and troubleshooting any runtime errors. For more information about managing the bus and server environment, see "Managing the bus environment."

Managing the bus environment

A service integrator can deploy an SCA module without knowing anything about the server and bus environment that powers the enterprise service bus. However, as an administrator, you might want to manage the server and bus environment. For example, you might want to start or stop a server, change the quality of service provided to SCA modules, add extra server capacity, or create a more distributed bus environment.

In a stand-alone server, the default bus environment comprises two service integration buses to be used for deploying SCA modules. You can also use the server to deploy J2EE components and resources, like servlets, enterprise Java beans, and JMS destinations.

The default bus environment might be adequate for your requirements. You can, however, create a bus environment with more server capacity, or a more distributed bus environment for different departments, or distinct bus environments to provide separate test and production facilities.

Managing the bus environment includes some installation and customization activities, bus topology planning, and creating product configurations. The focus is on the administration of service integration buses, servers, their resources, the set up and management of logical administrative domains of cells and nodes, and how to balance workload through clustering and high availability configurations.

The bus environment in WebSphere Process Server uses the service integration technologies provided by WebSphere Application Server. This set of topics provides information about tasks involving the service integration technologies needed to manage the bus environment.

- Getting started with the server and bus environment
- Managing the SCA.SYSTEM bus topology
- Configuring a server or cluster for the SCA runtime
- Managing servers
- Balancing workloads with server clusters
- Doing more with bus topologies

Getting started with the bus environment

Service Integrators can deploy SCA modules without considering the server and bus environment that powers the enterprise service bus. However, an Administrator might want to manage the server and bus environment, and therefore need to understand what that management involves.

This set of topics provides some background to managing the bus environment, to help you get started for the first time. If you are familiar with managing bus environments, you might choose instead to go directly to the other sets of task descriptions that are sub-topics of “Managing the bus environment” on page 159.

Overview of the bus environment:

The bus environment comprises one or more service integration buses, ESB servers, and their resources, organized into logical administrative domains of cells and nodes.

Although a service integrator can deploy an SCA module without knowing anything about the bus environment, as an administrator you might want to manage servers and buses, and therefore need to understand what those tasks involve:

- The SCA runtime (exploited by mediation modules) uses queues on an SCA.SYSTEM service integration bus as a robust infrastructure to support asynchronous interactions between components and modules. The queues are hosted by the server as a member of the SCA.SYSTEM bus.
- The ESB server provides the integration technologies, infrastructure services, configuration, and runtime administration needed to run mediation modules and service applications in WebSphere Process Server. As a bus member, the server has a messaging engine that provides the core messaging functionality of the SCA.SYSTEM bus.

Both the server and SCA.SYSTEM bus are configured with default properties that might be suitable for you to deploy and run your SCA modules.

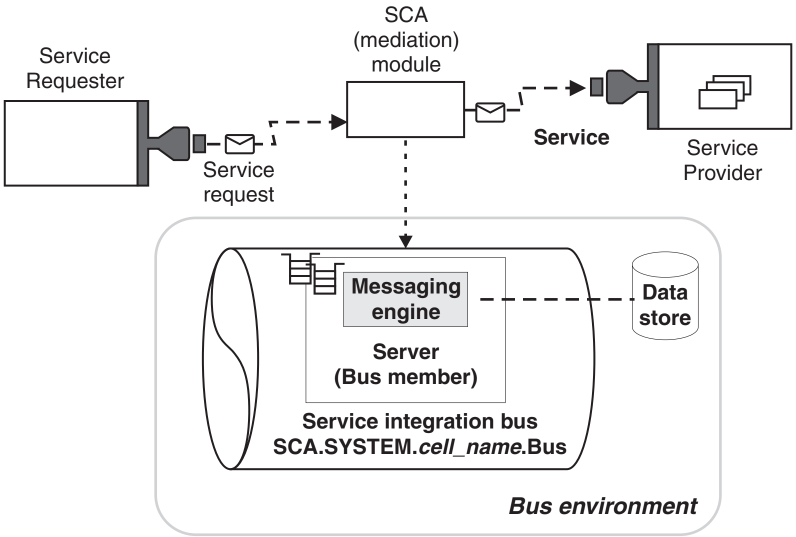


Figure 10. A bus environment with one server assigned to the SCA.SYSTEM service integration bus. As a bus member, the server is assigned one messaging engine, which has a data store for storing state data and messages. This figure also shows a mediation module deployed into the bus environment and assigned to the bus member.

For more advanced usage, you might want to change the configuration of the bus environment for a stand-alone server, or create a bus environment for a network deployment cell. For example:

- You can configure a variety of quality of service from secure, assured delivery (where messages are guaranteed not to get lost and are transported securely) to best-effort (where messages might get lost in case of a system failure).
- You might want to set up a network deployment cell to provide several servers to host mediation modules. This provides advantages of scalability, the ability to handle more client connections, and greater message throughput. You can also create server clusters, which enables you to manage a group of servers together and enables those servers to participate in workload management.
- Your complete bus environment might be made up of several stand-alone and deployment manager profiles, to provide separate administrative domains for different departments or to separate test and production facilities. Each profile has its own SCA.SYSTEM service integration bus.

Besides the SCA.SYSTEM bus used for SCA modules, you can also create other service integration buses that you can use to support the service integration logic provided by the modules. For example, the SCA.APPLICATION bus is provided and used to define JMS queue destinations and other JMS resources for modules deployed with JMS bindings.

You can create other buses for use as in WebSphere Application Server; for example, for applications acting as service requesters and providers within WebSphere Process Server, or to link to WebSphere MQ. You can also use a WebSphere Process Server deployment manager to manage separate application servers for use with applications and modules deployed onto WebSphere Application Server.

The application server:

WebSphere Process Server incorporates WebSphere Application Server, which provides a high performance, secure and manageable middleware container that supports standard J2EE application components, such as enterprise beans, Web services, servlets, JCA connectors, and asynchronous beans.

WebSphere Process Server provides functions for the deployment and administration of service applications, including a browser-based administrative console, command scripting, topology management (of service buses, servers, and clusters), a set of standard management programming interfaces called JMX, and an administrative model encompassing all hosted application components and infrastructure.

The server infrastructure provides network I/O, queuing, scheduling, threading, and dispatching functions common to most middleware. It also supports various transports and protocols to relieve developers of coding protocol semantics and message encodings.

WebSphere Process Server can augment and enhance an application by providing qualities of service as runtime capabilities that are normally difficult to implement. For example, requirements on the hosting container such as atomic transactions or a specific security role for the caller are expressed declaratively in a service's metadata. Operational capabilities such as performance monitoring and measurement, workload routing, and workload management can be enabled administratively.

WebSphere Process Server provides application runtime services that implement standard programming interfaces of the J2EE programming model; for example, application tracing and logging interfaces are integrated with those of the server runtime to provide a consistent view of problem diagnosis. Security interfaces extend WebSphere Process Server security to address application-specific business needs. Name space services enable the discovery and binding of services, and can be persisted throughout the administrative domain.

A service implementation deployed in WebSphere Process Server gains all of these advantages without the need to explicitly code them, nor to be aware of any of the server infrastructure.

Service integration buses:

A service integration bus provides a scope within which you can configure resources for mediation modules and interaction endpoints deployed in WebSphere Process Server.

Remember: An enterprise service bus is an architecture used for the integration logic and connectivity for SCA-described interaction endpoints. A service integration bus is part of the technologies provided by the WebSphere Application Server on which WebSphere Process Server is built. Your enterprise service bus might include multiple service integration buses, WebSphere MQ, and other technologies.

When abbreviating references about an enterprise service bus, we use the term *ESB*. When abbreviating references about a service integration bus, we use the term *bus*.

For WebSphere Process Server, a bus enables message routing between endpoints with specific quality of interaction service and can temporarily persist messages if required. You can configure a variety of quality of service from secure, assured delivery (where messages are guaranteed not to get lost and are transported securely) to best-effort (where messages might get lost in case of a system failure). Endpoint implementers choose their desired quality of service by declaring the annotations on SCA module exports and imports. If a quality of service is unspecified, WebSphere Process Server applies its defaults.

When you install WebSphere Process Server, an SCA.SYSTEM bus is created for you to deploy SCA modules. If the configuration of this bus is not adequate for your SCA modules, or you want service integration buses for other uses, you can choose between a variety of bus environments including a multiple-server bus, several single-server buses that use different servers, and buses linked to WebSphere MQ.

Many scenarios require a simple bus topology; perhaps, for example, a single server. By adding multiple servers to a single bus, you can increase the number of connection points. By adding server clusters as members of a bus, you can increase scalability and achieve high availability. Servers, however, do not have to be bus members to connect to a bus. In more complex bus topologies, multiple buses are configured, and might be interconnected to form complex networks. An enterprise might deploy multiple buses for organizational reasons; for example, an enterprise with several autonomous departments might want to have separately-administered buses in each location.

Server clusters:

A *server cluster* is a set of servers that are managed together and participate in workload management. The servers that are members of a cluster can be on different host machines.

Servers that belong to a cluster are *members* of that cluster set and must all have identical application components deployed on them. Other than the applications configured to run on them, cluster members do not have to share any other configuration data. One cluster member might be running on a huge multi-processor enterprise server system, while another member of that same cluster might be running on a smaller system. The server configuration settings for each of these two cluster members are very different, except in the area of application components assigned to them. In that area of configuration, they are identical. This allows client work to be distributed across all the members of a cluster instead of all workload being handled by a single application server.

In WebSphere Process Server, the support for server clusters is provided by the underlying WebSphere Application Server.

For more information about server clusters, see the related topics.

Choosing a bus environment

In a stand-alone server, the default bus environment comprises two service integration buses to be used for deploying SCA modules. If this is not adequate for your service applications, you have a choice of bus environments, including a multiple-server bus in a deployment manager cell, several single-server buses that use different servers, and other buses for serving applications or linking to WebSphere MQ.

In addition to the SCA.SYSTEM bus provided for SCA modules, you can create other application servers and service integration buses to support other applications and modules, or to connect to WebSphere MQ networks. Information about using other service integration buses, as in WebSphere Application Server

Network Deployment, version 6, is provided by links into the WebSphere Application Server topics. This set of topics is mainly focused on using the SCA.SYSTEM bus to support SCA modules.

The bus environment is created during installation and configuration, so you need to consider your requirements for a bus environment before deploying SCA modules. To choose a bus environment, consider the following points and then read the relevant sub-topics listed at the end of this topic.

- Consider the number of client connections and the throughput that you want for modules deployed to a bus.

  The aim is to identify the point at which the performance of the modules, as perceived by the clients, starts to deteriorate:

  – The number of concurrent client connections to the bus beyond which the performance starts to deteriorate when new client connections are made.

  – The number of requests and replies flowing through a messaging engine beyond which the performance starts to deteriorate when new attempts are made to send requests through the messaging engine.

  It is not possible to give a specific formula that applies to all environments, because it depends on the characteristics of the host on which the server runs, the nature of the modules deployed, and other factors.

  If you use a single-server bus and observe that the number of client connections is causing a deterioration in performance, or that the throughput starts to deteriorate, you can increase the capacity of the bus environment in several ways:

  – In a stand-alone profile, you might create several single-server buses using the same server. This enables the client connections to be distributed across several buses, but the throughput of requests still depends on the one server.

  – For greater capacity of client connections and request throughput, you might use multiple servers distributed across several buses. (To use multiple servers distributed over one or more buses, you need to have a server profile for a managed node in a network deployment cell.)

- Consider the size of requests flowing through a messaging engine.

  Every messaging engine manages two memory buffers that contain requests and request-related data. If there is not enough space when the messaging engine attempts to add data to a buffer, the messaging engine might discard data already in the buffer to make space.

  You might observe that a running messaging engine discards data from its buffer more often than is acceptable. In this case, you might add another server to the bus, to provide another messaging engine. Alternatively, you might choose to create several single-server buses that use different servers as their bus members. The messaging engine in each server uses a separate set of memory buffers, and a separate data store. (To use multiple servers distributed over one or more buses, you need to have a server profile for a node in a network deployment cell.)

- Consider if you want to use different qualities of service for your service applications.

  Each bus has a unique configuration of qualities of service and other properties. You might choose to create several buses and configure them with different qualities of service, then deploy each of your modules to the bus that has a suitable configuration.

- Consider other reasons for using multiple servers within a bus.

A service integration bus consisting of just one server is adequate for some applications. However, there are advantages in using more than one server in a bus (with each server providing a messaging engine):

- – Spreading messaging workload across multiple servers.
- – Placing request processing close to the requester applications, so as to reduce network traffic. For example, if the sending and receiving applications are running in the same server process it would be inefficient to route all the requests that flow between them through a messaging engine running in a remote server.
- – Improving availability in the face of system or link failure. This includes removing a single point of failure, and allowing store and forward between two servers should this be required.
- – Providing improved scalability
- – Accommodating firewalls or other network restrictions that limit the ability of network hosts to all connect to a single messaging engine.

- Consider other reasons for using a multiple SCA.SYSTEM bus environment.

  Because each service integration bus has a separate configuration, you might choose to have several buses each with a different configuration suitable for separate modules; for example, you might have some buses for a production environment with security, and other buses for testing without security.

  You might also choose to create several buses to separate the administration of modules; for example, separate administrative cells and their SCA.SYSTEM buses might be used for different departments within organizations, or perhaps to separate test and production facilities.

  Besides the SCA.SYSTEM buses, other buses might be created for other application use, and can be connected to allow messaging across the buses. Buses in different organizations can also be connected. When buses are interconnected, applications can send messages to applications on other buses, and use resources provided on other buses. Published messages can also span multiple buses, if the links between the buses are configured to allow it.
- Consider reasons for using non-SCA service integration buses.

  Besides the SCA.SYSTEM bus used for SCA modules, you can also create other service integration buses that you can use to support the service integration logic provided by the modules. For example, the SCA.APPLICATION bus is provided and used to define JMS queue destinations and other JMS resources for modules deployed with JMS bindings.

  You can create other buses for use as in WebSphere Application Server, for applications acting as service requesters and providers within WebSphere Process Server, or to link a bus to WebSphere MQ. You can also use a WebSphere Process Server deployment manager to manage separate application servers, for use with applications and modules deployed onto WebSphere Application Server.
- Consider if you want application servers that do not support SCA modules.

  A WebSphere Process Server deployment manager cell can include application server nodes that run WebSphere Application Server servers. You can use these application servers for applications and modules supported by WebSphere Application Server. You do not need to add the application servers into a service integration bus, unless you want to exploit the service integration technologies of WebSphere Application Server.

The SCA* buses provided:

To support the deployment of SCA modules, one or two service integration buses are created for you. These buses, with names starting SCA, are sometimes referred to as the SCA* buses.

If you create a WebSphere Process Server stand-alone profile, the single application server is automatically configured for SCA support, and is made a member of the SCA.SYSTEM bus and the SCA.APPLICATION bus.

In a new network deployment cell, the SCA.SYSTEM bus is created for you when you configure the first cluster or server for asynchronous SCA support with local messaging. The server or cluster being configured becomes a member of the bus. Similarly, when you configure the first cluster or server for JMS SCA support, the SCA.APPLICATION bus is created for you.

SCA.SYSTEM bus

This is the bus used to host queue destinations for SCA modules, such as mediation modules that provide service integration logic in the enterprise service bus. The SCA runtime (exploited by mediation modules) uses queue destinations on the SCA.SYSTEM bus as a robust infrastructure to support asynchronous interactions between components and modules.

For a network deployment cell, this bus is created when you configure the first server or cluster for asynchronous SCA support.

SCA.APPLICATION bus

This bus is used to create resources for modules deployed with JMS bindings. This bus is an example of how you might create a service integration bus for use other than to deploy SCA modules.

For a network deployment cell, this bus is created when you configure the first server or cluster for JMS SCA support.

The SCA.SYSTEM bus provides a scope within which resources, such as queue destinations, are configured for mediation modules and interaction endpoints. The bus enables message routing between endpoints with specific quality of interaction service and can temporarily persist messages if required. You can configure a variety of quality of service from secure, assured delivery (where messages are guaranteed not to get lost and are transported securely) to best-effort (where messages might get lost in case of a system failure). Endpoint implementers choose their desired quality of service by declaring the annotations on SCA module exports and imports. If a quality of service is not specified, WebSphere Process Server applies its defaults.

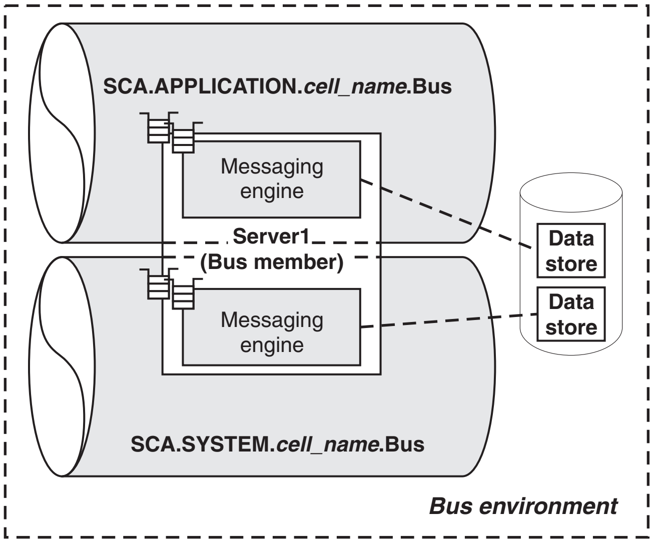


*Figure 11. The SCA.*buses for a stand-alone server profile.* The service integration buses created when you install WebSphere Process Server.

In a stand-alone profile, there is one bus member that provides the one messaging engine in a bus, a topology that is adequate for some applications.

A single messaging engine might not be adequate if the number of client connections becomes excessive, if the rate of message throughput cannot be sustained by the one messaging engine, or if the size of messages has a detrimental affect on the message buffers used by the messaging engine.

To add more than one messaging engine to a service integration bus, you need to have a managed node in a network deployment cell.

Creating a single-server enterprise service bus:

The simplest topology is an enterprise service bus consisting of a single stand-alone server.

In a single-server enterprise service bus, the SCA.SYSTEM bus has the one server as its only bus member. When you install a mediation module into WebSphere Process Server, the queue destinations used by the module are defined on that bus member. These 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.

The server hosts a messaging engine that provides the service integration technologies used by the queue destinations and the processing of requests.

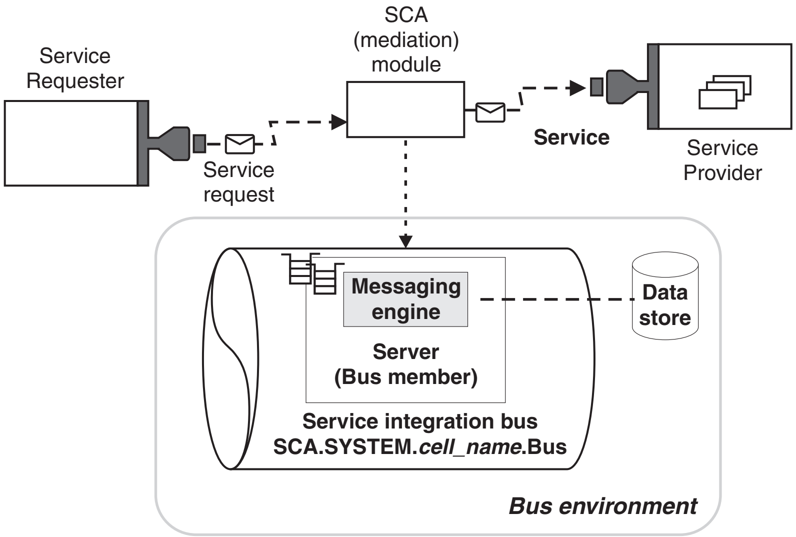


Figure 12. A single-server bus. A bus environment showing one service integration bus, with one ESB server as a bus member. This figure also shows an SCA module deployed into the bus environment, where the destinations needed are assigned to the bus member.

In a stand-alone server, there can be only one messaging engine in a bus, a topology that is adequate for some scenarios.

A single messaging engine might not be adequate if the number of client connections becomes excessive, if the rate of message throughput cannot be sustained by the one messaging engine, of if the size of messages needs more space than the one messaging engine can provide in its data store.

To add more than one messaging engine to a service integration bus, you need to create a managed node in a network deployment cell.

1. Perform a Complete installation of WebSphere Process Server. This installs the core product files and creates the first stand-alone server profile.
2. Start server1 using the First steps console or the startServer server1 command.

You can now run the WebSphere Process Server samples and deploy service applications into your enterprise service bus.

Creating a multiple-server enterprise service bus without clustering:

A enterprise service bus that consists of multiple servers provides advantages of scalability, the ability to handle more client connections and greater message throughput. You can also deploy SCA modules to different servers; for example, to provide different resources and qualities of service, or to provide some separation for different departments within organizations, or perhaps to separate test and production facilities.

To create more than one server in your bus environment, you need to have a managed node in a network deployment cell.

You configure each server for the SCA runtime required by mediation modules. This advanced configuration defines whether queue destinations exploited by the SCA runtime are hosted locally on the server or on a remote server. If you specify that the server is to host queue destinations, the server is made a member of the

SCA.SYSTEM bus and gets a messaging engine to which the queue destinations are assigned. If you specify that the server does not host queue destinations, it does not need to be a member of the SCA.SYSTEM bus, so does not need a messaging engine.

Consider the scenario shown in the figure Figure 13.

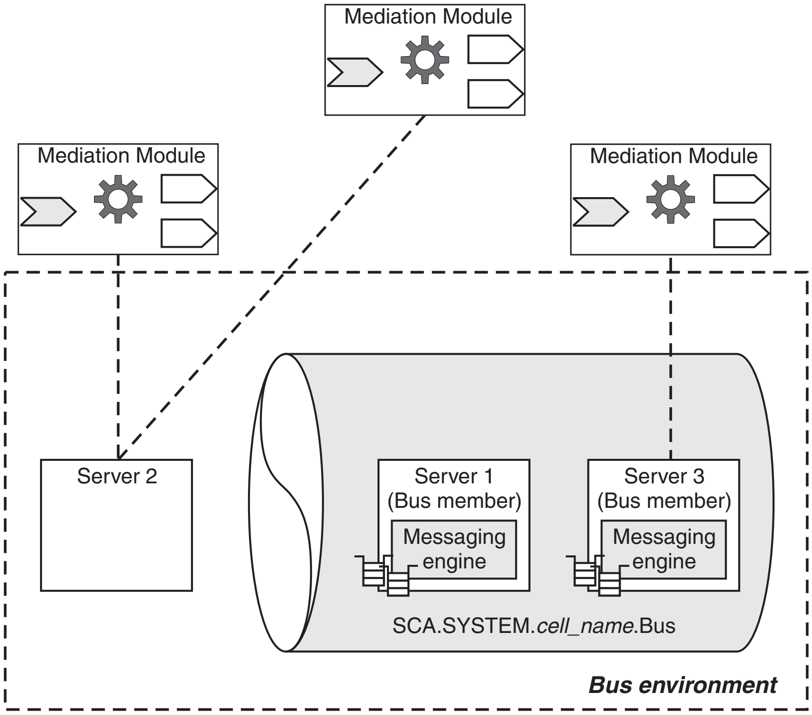


Figure 13. A multiple-server bus without clustering. Server 1 is configured to host queue destinations for mediation modules deployed to any of the servers, but not to host mediation modules or other service applications. Server 2 is configured to host mediation modules or other service applications, but to use queue destinations hosted on another server (server 2 or server 3 in this figure.) Server 3 is configured to both host mediation modules or other service applications and to host queue destinations for mediation modules deployed to any of the servers.

All of the messaging engines in the SCA.SYSTEM bus are implicitly connected, and requests can be processed by any messaging engine in the bus. Knowledge of the resources assigned to each messaging engine in a bus is shared between all the messaging engines in the bus.

There is no requirement for all the messaging engines in the bus to be running at the same time; if one of the messaging engines is stopped, the rest of the messaging engines continue to operate. However resources owned by a messaging engine, specifically queue points for mediation modules, are unavailable if the engine is stopped. Also, a messaging engine can only run in the server it was created for. The server is therefore a single point of failure; if the server cannot run, the messaging engine is unavailable. By configuring a server cluster as a bus member, a messaging engine has the ability to run in one server in the cluster, and if that server fails, the messaging engine can run in an alternative server within the cluster.

You can create a multiple-server enterprise service bus in any network deployment cell. In general, the following steps are involved, as described in Configuring for network deployment.

1. Decide which network deployment scenario you are creating.

2. Configure a network deployment cell with one or more managed nodes.
3. From the deployment manager administrative console, create servers on the managed nodes.
4. Configure the servers for the SCA runtime.

You can now run the WebSphere Process Server samples and deploy service applications into your enterprise service bus.

Multiple-server enterprise service bus with clustering:

A network deployment cell can be used for an enterprise service bus that consists of multiple servers, some or all of which are members of server clusters.

A server that hosts queue destinations for SCA modules has one messaging engine in the SCA.SYSTEM bus. For many purposes this is sufficient, but such a messaging engine can only run in the server it was created for. The server is therefore a single point of failure; if the server cannot run, the messaging engine is unavailable. By configuring a server cluster as a bus member instead, the messaging engine has the ability to run in one server in the cluster, and if that server fails, the messaging engine can run in an alternative server. This is illustrated in Figure 14.

Another advantage of configuring a cluster bus member is the ability to share the workload associated with an SCA module across multiple servers. For an SCA module deployed to a cluster bus member, the queue destinations used are partitioned across the set of messaging engines run by the cluster servers. The messaging engines in the cluster each handle a share of the messages passing through the SCA module.

To summarize, with a cluster bus member you can achieve either failover, workload sharing, or both, depending on policies that you can configure.

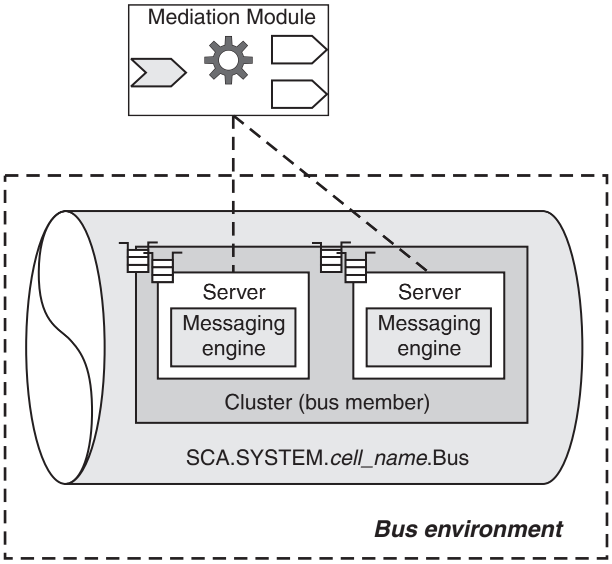


Figure 14. A multiple-server bus with clustered servers for failover

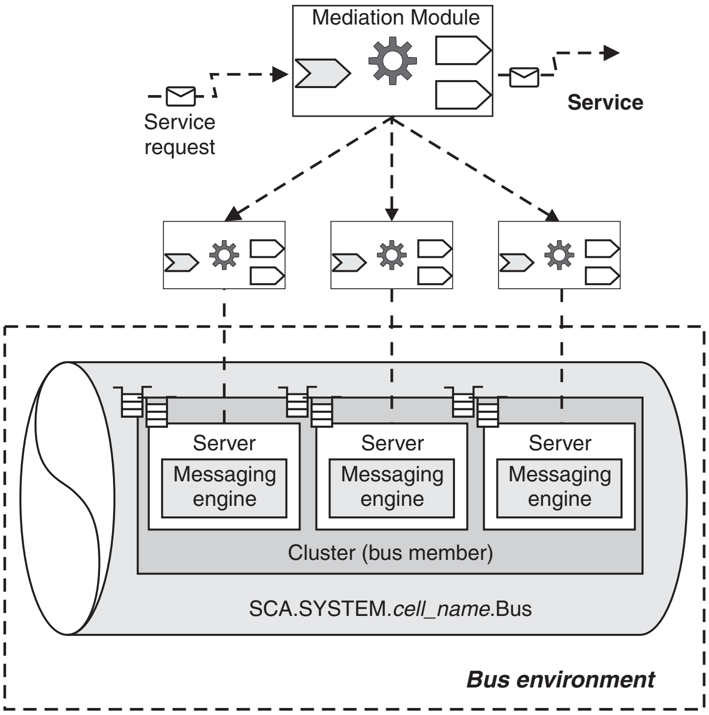


Figure 15. A multiple-server bus with clustered servers for workload sharing

You can create a multiple-server enterprise service bus with clustering in any network deployment cell. In general, the following steps are involved, as described in “Configuring for network deployment.”

1. Decide which network deployment scenario you are creating.
2. Configure a network deployment cell with one or more managed nodes.
3. From the deployment manager administrative console, create clusters on the managed nodes.
4. Configure the clusters for the SCA runtime.

You can now run the WebSphere Process Server samples and deploy service applications into your enterprise service bus.

Configuring for network deployment:

Network deployment allows you to administer a group of servers centrally from a deployment manager. A deployment manager administers the configuration and state for all of the managed nodes in its cell and deploys applications to any managed node in the cell.

Before configuring for network deployment, ensure that you have completed the following steps:

- Decide which network deployment scenario you are creating. .
- Decide whether you want to use nodes on both distributed and z/OS operating systems in the same cell.
- Decide whether you want to use multiple cells.
- Decide whether or not to use server clusters.

If you have decided that your solution requires network deployment, then you require a deployment manager and managed nodes.

To configure for network deployment, complete the following steps:

1. Create and start the deployment manager, ready to add managed nodes to the cell.
   a. On one system, configure a deployment manager.
   b. Start the deployment manager.
   c. On the deployment manager, start the administrative console.
2. Add managed nodes to the cell.
   a. **Optional:** Federate a stand-alone server node to the deployment manager, as the first managed node. After using a stand-alone server to validate the use of service applications, you can federate that node into a new network deployment cell. This enables you to evolve the use of those service applications into a distributed group of servers and server clusters, on one or more nodes.

      This step adapts the SCA* buses of the stand-alone server to be the SCA* buses of the deployment manager cell.
   b. Configure a managed node, as a custom profile, on a different system to the deployment manager. You can federate a custom node to the deployment manager, to become an empty managed node.
   c. If you did not federate the custom node when creating its profile, you must now federate that custom node to the deployment manager.
3. **Optional:** Configure your own choice of data store configuration for messaging engines on the SCA* buses.

   When you configure a server cluster to host destinations for the SCA runtime, a new messaging engine is created on the SCA* buses. You must explicitly configure both the data store and the JDBC data source that the messaging engine uses to interact with its data store.

   Similarly, if you configure a server for the SCA runtime, a new messaging engine is created on the SCA* buses and configured to use a Cloudscape data source by default. This default configuration enables the messaging engine to run without needing any additional configuration by the Administrator. If you do not want to use the default data source configuration, you can change the configuration parameters. For example, you can change the data source or you can configure the data store to use a different JDBC provider.
4. Configure a server cluster for each managed node.
   a. Create a server cluster. For more information about how to do this, see "Creating a server cluster" in the Administering the Bus book.
   b. Configure the cluster for the SCA runtime required by mediation modules, as described in "Configuring servers or clusters for the SCA runtime" in the Administering the Bus book. Before deploying any mediation modules onto the new cluster, configure the cluster to indicate where queue destinations used by the SCA runtime for the module are to be hosted.
   c. Start the cluster. For more information about how to do this, see "Managing available server clusters" in the Administering the Bus book.

   Repeat this step for each managed node that is to support a server cluster.
5. Configure any servers to run outside clusters. It is not common to add single servers to a network deployment environment. Generally, it is advisable to have clusters in your network deployment environment to provide high availability.

a. On a managed node, create a server. For more information about how to do this, see "Creating a server" in the Administering the Bus book.

b. Configure the server for the SCA runtime required by mediation modules, as described in "Configuring servers or clusters for the SCA runtime" in the Administering the Bus book. Before deploying any mediation modules onto the new server, configure the server to indicate where queue destinations used by the SCA runtime for the module are to be hosted.

Repeat this step for each managed node that is to support a server.

Creating a multiple enterprise service bus topology:

You might want to deploy and manage SCA modules in a distributed bus environment; for example, with separate enterprise service buses for different departments or to separate test and production facilities.

Each stand-alone server or deployment manager that you create has its own administrative cell that can be seen as the administrative domain for a separate enterprise service bus.

Your complete bus environment might be made up of several stand-alone server and network deployment cells, each representing a separate enterprise service bus, with its own SCA.SYSTEM bus used for SCA modules.

Besides the SCA.SYSTEM bus used for SCA modules, you can also create other service integration buses that you can use to support the service integration logic provided by the modules. For example, the SCA.APPLICATION bus is provided for JMS SCA support, and used to define JMS queue destinations and other JMS resources for modules deployed with JMS bindings.

You can create other buses for use as in WebSphere Application Server; for example, for applications acting as service requesters and providers within WebSphere Process Server, or to link a bus to WebSphere MQ.

You can also use a WebSphere Process Server deployment manager to manage separate application servers, for use with applications and modules deployed onto WebSphere Application Server.

You can use these other buses separately or you can connect them to allow messages to pass between the buses. You can also connect together buses in different organizations. When buses are interconnected, applications can send messages to applications on other buses, and use resources provided on other buses. Published messages can span multiple buses where the links between the buses are configured to allow it.

A service integration bus must be contained within a single cell; that is, a bus cannot span multiple cells. However, a cell can contain more than one bus. In this case, each bus in the cell is "foreign" to each other bus in the cell. You can connect buses together within a cell, or between different cells.

The process for linking one bus to another bus is the same, whether the buses are in the same cell or in different cells.

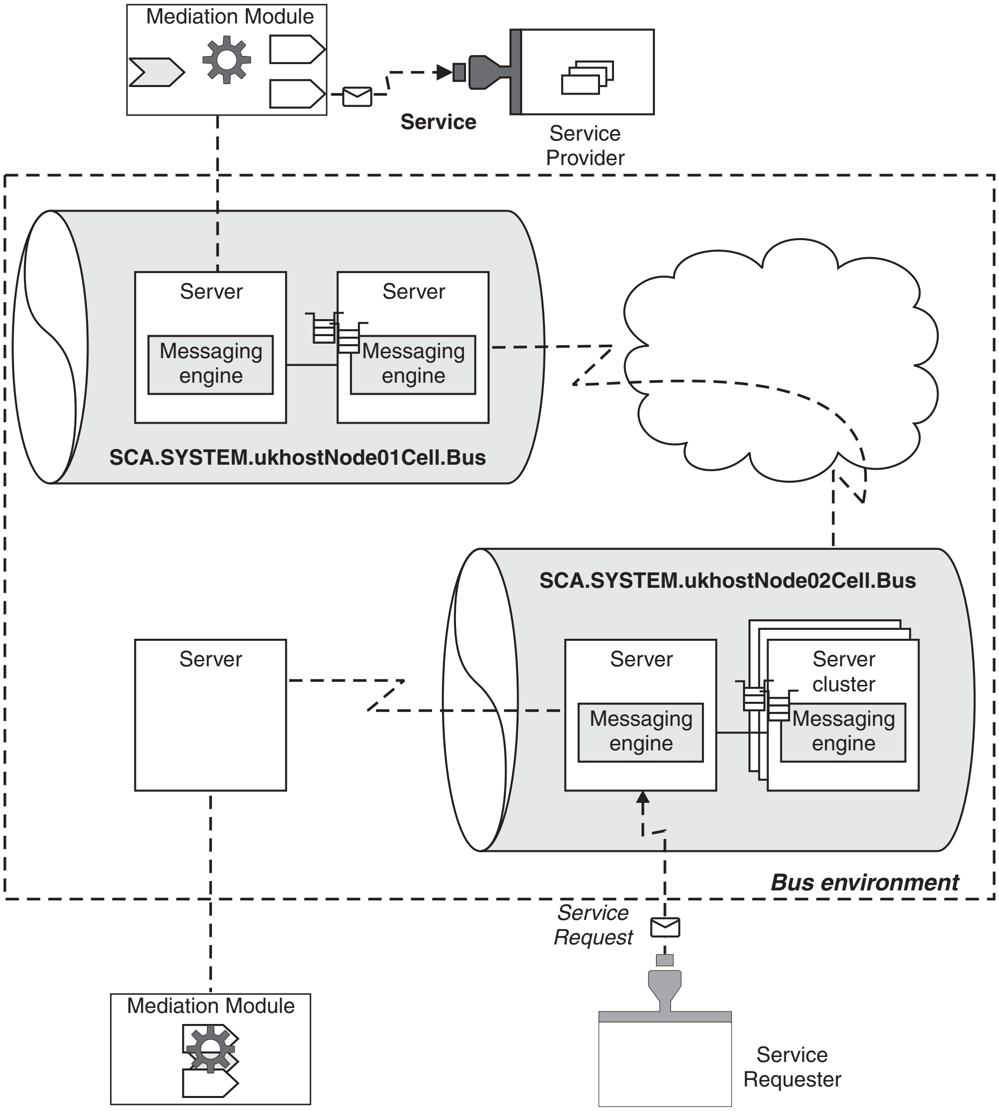


Figure 16. A multiple-enterprise service bus topology

To create a multiple enterprise service bus topology you can configure several stand-alone servers and network deployment scenarios.

- Create several separate stand-alone configurations.
- Create several network deployment configurations.

You can now run the WebSphere Process Server samples and deploy service applications into your enterprise service bus. If you have connected buses in different cells, modules in one cell can send messages to modules in the other cell, and can use resources provided by the buses in the other cell.

An enterprise service bus (ESB) with links to WebSphere MQ networks:

An ESB can contain links to WebSphere MQ networks. This allows applications connected to a WebSphere MQ queue manager to send messages to an application that is attached to a service integration bus, and for such an application to send messages to WebSphere MQ.

When a service integration bus is connected to a WebSphere MQ network, the network is extended by adding support for:

- Service applications deployed into the service integration bus
- JMS applications attached to the service integration bus
- Web services requesters or providers

The WebSphere MQ network is represented by a *foreign bus* configured on a messaging engine. A construct called an *WebSphere MQ link* connects the messaging engine to an MQ queue manager using sender and receiver channels, thereby providing a bridge between the bus and the so-called *gateway queue manager* of a WebSphere MQ network.

The WebSphere MQ link provides connectivity not just with the messaging engine that hosts the link, but also with the other messaging engines in the bus. All the messaging engines in the bus appear to the WebSphere MQ network as if they were a single queue manager (they inherit the queue manager name from the WebSphere MQ link)

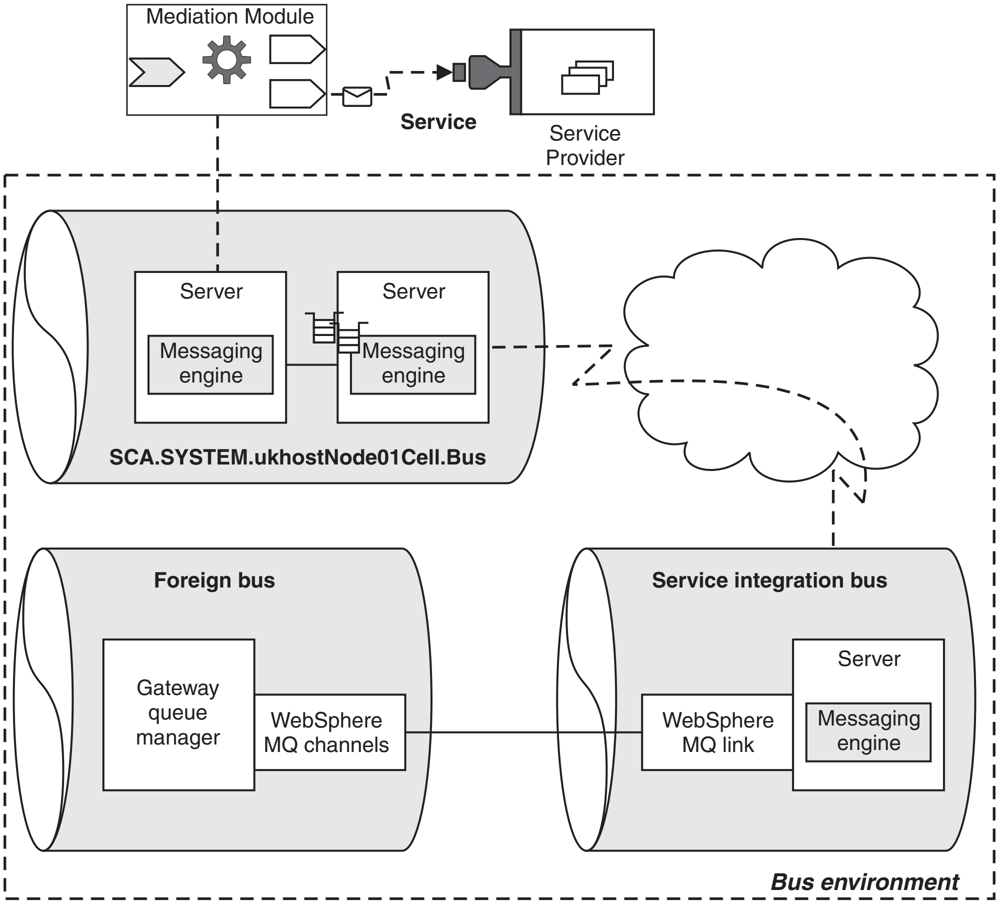


Figure 17. Service integration buses with links to a WebSphere MQ network

WebSphere MQ links can be used in a number of different configurations. A messaging engine can contain multiple WebSphere MQ links to different gateway queue managers.

Links to WebSphere MQ networks are implemented by the service integration technologies of WebSphere Application Server. For more information about using links to WebSphere MQ networks, see Interoperating with WebSphere MQ in the WebSphere Application Server documentation.

Server and bus features of the administrative interfaces

You can view and change aspects of the server and bus environment from the administrative console or from the command line.

Servers → Application servers

This set of pages provides access to ESB servers, which are application servers created from a special template provided by WebSphere Process Server.

The ESB server provides the integration technologies, infrastructure services, configuration, and runtime administration needed to run mediation modules.

The detail settings page for a server provides access to the service integration properties under the **Server messaging** category label, including messaging engines, messaging engine inbound transports, and WebSphere MQ link inbound transports.

Service integration → Buses

This set of pages provides access to the service integration technologies of WebSphere Application Server, which are used in the deployment of mediation modules (and other SCA modules).

Each service integration bus provides a scope within which you can configure resources for mediation modules and interaction endpoints deployed in WebSphere Process Server. The bus enables message routing between endpoints with specific quality of interaction service and can temporarily persist messages if required.

If you create a WebSphere Process Server stand-alone profile, the single application server is automatically configured for SCA support, and is made a member of the SCA.SYSTEM bus and the SCA.APPLICATION bus.

In a new network deployment cell, the SCA.SYSTEM bus is created for you when you configure the first cluster or server for asynchronous SCA support with local messaging. The server or cluster being configured becomes a member of the bus. Similarly, when you configure the first cluster or server for JMS SCA support, the SCA.APPLICATION bus is created for you.

When you install a mediation module into WebSphere Process Server, the necessary bus destinations are created on the SCA* buses.

The detail settings page for a bus provides access to service integration resources under a number of category labels, including bus destinations.

Managing the SCA.SYSTEM bus topology

The SCA.SYSTEM bus is a service integration bus that is used to deploy SCA modules. You can display the local topology of the bus, change the properties of the bus, add servers and clusters as bus members, and perform other administrative tasks to manage the overall topology.

A stand-alone server with its own administrative cell is assigned to the SCA.SYSTEM service integration bus for the cell. This bus topology is adequate to deploy SCA modules for some application integration scenarios, but you may want to create a different bus topology to take advantage of using more than one member in a bus, or of using more advanced environments involving multiple service integration buses and perhaps links to WebSphere MQ.

If you create a deployment manager, you can add managed nodes to the network deployment cell, and create one or more servers or server clusters on those nodes, as members of the SCA.SYSTEM bus, to deploy SCA modules. When created, the SCA.SYSTEM bus and servers are assigned values for only some of the important properties, with other properties taking default values.

Use the information in the following sub-topics to display the topology and properties of the SCA.SYSTEM bus and servers, to configure them to your needs, and to manage their runtime state. These sub-topics are aimed at routine tasks to support the deployment of mediation modules into the SCA.SYSTEM bus. These tasks, and other tasks, for the general management of the server and bus environment are provided in the list of related tasks and under “Doing more with bus topologies” on page 201.

When managing the SCA.SYSTEM bus and servers, consider the following:

- How are SCA modules to be distributed across servers?
- If you want servers to be managed together or participate in workload management, what server clusters do you want?
- What are the naming conventions for buses, servers, and other resources?
- What message persistence and other qualities of services do you want to provide?
- What databases do you want to use for messaging engines to persist messages?

Creating the initial SCA.SYSTEM bus topology:

If you create a WebSphere Process Server stand-alone profile, the single application server is automatically configured for SCA support, and is made a member of the SCA.SYSTEM bus and the SCA.APPLICATION bus. In a network deployment cell, when you configure the first cluster or server for asynchronous SCA support, the SCA.SYSTEM bus is created for you.

Before continuing with this task, consider whether you are likely to need more than one server or server cluster in the bus environment. In a stand-alone server profile, there can be only the one server. If you are likely to want several servers, or server clusters, in the same bus environment, you should consider creating a network deployment cell with managed nodes. You can also create several stand-alone servers on one or machines.

Each stand-alone node or network deployment cell forms a separate administrative cell that has its own service integration bus called SCA.SYSTEM bus that is used to deploy SCA modules within the cell.

Use this task to create the SCA.SYSTEM bus for a cell, with one ESB server as a bus member. The SCA.SYSTEM bus and server are created with values for only some of the important properties, with other properties taking default values.

You can change the configuration of the buses; for example, to change bus properties or, for a network deployment cell to add server clusters or servers as bus members.

To create the SCA.SYSTEM bus with one server as a bus member, complete one of the following steps:

- Create a stand-alone server.

  If you create a WebSphere Process Server stand-alone profile, the single application server is automatically configured for SCA support, and is made a member of the SCA.SYSTEM bus and the SCA.APPLICATION bus.

  This topology is limited to the one server.
- Create a network deployment cell then add a cluster or server on a managed node. In a new network deployment cell, the SCA.SYSTEM bus is created for you when you configure the first cluster or server for asynchronous SCA support with local messaging. The server or cluster being configured becomes a member of the bus. Similarly, when you configure the first cluster or server for JMS SCA support, the SCA.APPLICATION bus is created for you.

  Network deployment has advantages of centralized administration and potential workload balancing across server clusters. You can start with a cell containing one managed node for a custom profile, then add more clusters or servers to that node, or add more managed nodes, if needed later.

When you log in to the administrative console for the cell, you can see the SCA.SYSTEM bus in the list on the **Service integration** → **Buses** page. You can also see the new server in the list on the **Servers** → **Application Servers** page.

After starting the new server, you can use the SCA.SYSTEM bus to deploy SCA modules.

You can change the configuration of the buses; for example, to add servers or server clusters to a network deployment cell.

Displaying the topology of a service integration bus:

The bus topology view of the administrative console displays the bus members and messaging engines of a selected bus. For each messaging engine, the runtime status is displayed.

Use this task to display a tree view of the bus members and messaging engines in a service integration bus. You can also use this *local topology* view to add servers and server clusters as members of the bus.

To display the local topology view of a service integration bus, use the administrative console to complete the following steps:

1. In the navigation pane, click **Service integration** → **Buses**. A list of buses is displayed in the content pane.
2. In the content pane, click the name of the bus. For example, `SCA.SYSTEM.localhostCell01.Bus`
3. Click the tab **Local Topology**

The topology of the bus is displayed as an expandable tree.

You can expand nodes of the tree to display the bus members and their messaging engines.

You can also add servers and server clusters as members of the bus.

Changing bus properties:

You can create a service integration bus with its properties left to take their default values. You can later change the values of properties to adapt the use of a bus to meet changes in your needs.

If you think that you might need to change the use of a service integration bus, you can use this task to help you choose which properties to change. This task provides an overview of the considerations for changing bus properties, and links to a set of task descriptions provided for the service integration technologies of the underlying WebSphere Application Server.

- Configuring messaging security for the SCA.SYSTEM bus.

  When messaging security is switched on, all access to the bus itself and to all the destinations on the bus must be authorized. This means that all users who want to connect to the bus must have permission to use the bus resources, either directly or as part of a user group.

  When the SCA.SYSTEM bus is created, messaging security for a bus is switched on by default. Only authorized messaging engines are allowed to create a connection to a secure bus.

  The SCA.SYSTEM bus is configured with the authentication alias called SCA_Auth_Alias, with an initial set of default authorization permissions that allows SCA to connect to the secured bus, and grants full access to all local destinations on the bus. You can change the default authorization permissions to restrict access to a bus to a specific set of users, and can choose to create and use a different authentication alias.

  By default, the same SCA_Auth_Alias is used as the inter-engine authentication alias, to authorize communication between messaging engines on the bus. You can choose to create and use a different inter-engine authentication alias.

  To further configure security, you can use secure transport connections (SSL or HTTPS) to ensure confidentiality and integrity of messages in transit between application clients and messaging engines and between messaging engines. You can specify the transport chain InboundSecureMessaging (for JFAP over SSL over TCP/IP) that is provided with WebSphere Process Server. Alternatively, you can also create and specify another transport chain.
- Configure whether messages on a deleted message point should be retained or discarded.

  If you uninstall the application for an SCA module, the queue points that the SCA runtime uses for that module are deleted. By default, the Discard messages property is cleared, so any messages on any of those queue points are retained at a system exception destination so that you can process them later. You can select the Discard messages property so that any messages on deleted queue points are discarded.
- Configure whether any updates to the configuration information are dynamically passed to messaging engines.

  By default, the Configuration reload enabled property of the SCA.SYSTEM bus is selected. This indicates that any updates to the configuration information are dynamically passed to the server, and therefore made available to messaging engines whether or not they are started. (When a messaging engine is started, it uses the information in the server that it is running in.)
- Configure the maximum total number of messages that a messaging engine can place on its message points.

  By default, the High message threshold property of the SCA.SYSTEM bus is set to 50000 messages. This sets the initial value of that property on each messaging engine in the bus. If the current message depth on a queue point is equal to the high message threshold, the messaging engine does not accept new messages until the queued messages have been consumed. You can change the High message threshold property (on the bus or on a messaging engine) as one way to overcome problems with the throughput of messages.

Setting up the data store for a messaging engine

The database administrator and WebSphere Process Server administrator need to set up a data store for each messaging engine of the SCA runtime.

Before you can deploy a mediation module into WebSphere Process Server, you must have configured a server or server cluster to host destinations for the SCA runtime required. This advanced configuration adds the server or cluster as a member of the SCA.SYSTEM bus, and configures the messaging engine for the server or cluster on that bus. If you perform the JMS element of SCA configuration for a server or cluster, a second messaging engine is created for the SCA.APPLICATION bus. Each messaging engine requires a separate database or schema to be created.

Each messaging engine has its own *data store* to store persistent data in a database. All the tables in the data store are held in the same database schema. You can create each data store in a separate database. Alternatively, you can create multiple data stores in the same database, with each data store using a different schema.

A messaging engine uses an instance of a JDBC data source to interact with the database that contains the data store for that messaging engine.

Before configuring a server cluster to host destinations for the SCA runtime, you must configure both the data store and the JDBC data source that the messaging engine uses to interact with its data store.

If you configure a server to host destinations for the SCA runtime, the messaging engine is configured to use a Cloudscape data source by default. This default configuration enables the messaging engine to run without needing any additional configuration by the Administrator. If you do not want to use the default data source configuration, you can change the configuration parameters. For example, you can change the data source or you can configure the data store to use a different JDBC provider.

Note: For detailed set up instructions for the z/OS platform, see the Redbook Architecting High Availability Using WebSphere V6 on z/OS.

To set up the data store for a messaging engine, complete the following steps. These steps are the same as for the underlying WebSphere Application Server, and link to WebSphere Application Server topics for more information.

1. Ask your database administrator to perform the following steps:
   a. Create the database. Choose which relational database management system (RDBMS) you want to use for the data store. Unless you want a server to use the default Cloudscape provider, create the database before you configure a server to host destinations for the SCA runtime. Refer to the documentation for your chosen RDBMS for information about how to create a database.

      The default database name for SCA messaging engines is SCADB.

   b. Create users and schemas. The default schema name is IBMWSSIB.

      If your database type does not support multiple schemas in the same database instance, you must configure a separate database instance for each messaging engine.

For more information about creating users and schemas, see Creating users and schemas in the database in the WebSphere Application Server documentation.

c. Create the data store tables. Decide whether you want your database administrator to create the tables, or you want to allow WebSphere to create the tables automatically. To create the tables automatically, ensure that WebSphere Process Server has sufficient authority to create tables and indexes. Later, when you configure your messaging engine, you indicate that its tables should be created automatically.

   For more information about creating data store tables for a messaging engine, see Creating the tables in the WebSphere Application Server documentation

d. Create a database user ID with sufficient privilege to access the data store. A messaging engine uses a database user ID to authorize access to its data store tables. The user ID must have the authority to be able to create tablespaces, tables, indexes, and stored procedures for the data store.

   For more information about the privileges that you require for your chosen database, see Database privileges in the WebSphere Application Server documentation.

2. Perform the following steps:
   a. Obtain the database user ID from your database administrator.
   b. Configure a JDBC data source. The following naming convention is used for the JDBC data source that the messaging engine uses to interact with the database:
      - SCA.SYSTEM bus: *nodeserver|cluster-SCA.SYSTEM_bus_name*
      - SCA.APPLICATION bus: *nodeserver|cluster-SCA.APPLICATION_bus_name*

      For a messaging engine in a server cluster, the data store configuration depends on the purpose of the cluster:
      - For workload sharing, every messaging engine must have its own data store.
      - For failover, you must have one data store for all of the messaging engines in the failover cluster. When a messaging engine runs in a failover cluster, its data store must be accessible from every server that can potentially run the messaging engine that owns the data store.

      When a messaging engine is created as a member of a cluster, you must explicitly configure both the data store and the JDBC data source that the messaging engine uses to interact with the data store:
      - If each messaging engine has its own JDBC data source, you can create the data sources at node scope.
      - If several messaging engines share the same JDBC data source, you must create the data source at cluster scope, not node scope.

      You must also ensure that you install the JDBC driver jars at every node on which a cluster member runs.

      For more information about configuring a data source, see Configuring a JDBC data source in the WebSphere Application Server documentation.

Configuring servers or clusters for the SCA runtime

Before you can deploy a mediation module onto a server or server cluster, you must have configured that server or cluster for the SCA runtime required. This advanced configuration defines whether queue destinations exploited by the SCA runtime are hosted locally or on a remote server or cluster.

If you have created a stand-alone server profile, that server has already been configured for the SCA runtime, so do not perform this task for that server.

If you have created a server or server cluster in a network deployment to support SCA modules, you must complete this task to configure that server or cluster for the SCA runtime required.

Before starting this task, you must have already created one or more servers or server clusters required for the SCA runtime:

- The server or cluster on which you want to deploy mediation modules.
- The server or cluster on which you want to create queue destinations for the SCA runtime to exploit.

You can use the same server or cluster in both cases.

For more information about creating servers or server clusters, see the following task descriptions:

- Creating a server
- Creating a server cluster

When you install a mediation module into WebSphere Process Server, a number of queue destinations are created for components of the mediation module that use asynchronous interactions. The queue destinations are used by the SCA runtime to hold messages being processed for the mediation module.

To specify where the queue destinations are created, you complete one of the following configuration alternatives for the server or cluster.

- Set the advanced configuration of the server or cluster to host destinations. The configuration defines that this server or cluster can host queue destinations for mediation modules deployed locally or for mediation modules deployed on other servers or clusters. This configuration also adds the server or cluster as a member of the SCA.SYSTEM bus, and configures the messaging engine for the server or cluster.

  To perform this configuration task, see “Configuring a server or cluster to host queue destinations for mediation modules.”
- Set the advanced configuration of the server or cluster to use remote destinations. The configuration defines that this server or cluster uses a *remote destination location* for destinations.

  This server or cluster does not need to be a member of the SCA.SYSTEM bus.

  If you deploy a mediation module to this server or cluster, the queue destinations are configured on a different server or cluster that is a member of the SCA.SYSTEM bus.

  To perform this configuration task, see “Configuring a server or cluster to use remote destinations for mediation modules” on page 185.

The server or cluster is configured to accept the deployment of mediation modules.

You can now deploy mediations onto the server or cluster or take other actions to manage the server or cluster.

Configuring a server or cluster to host queue destinations for mediation modules:

For mediation modules, a cluster or server can be configured to host the queue destinations used by the SCA runtime. The server or cluster can host queue destinations for mediation modules that are deployed to any server or cluster in the administrative cell.

Before starting this task, you must have already completed the following actions:

- Set up the data store for the messaging engines in the cluster, as described in "Setting up the data store for a messaging engine" on page 180.
- Created the server or cluster, as described in one of the following:
  - Creating a server
  - Creating a server cluster

When you install a mediation module into WebSphere Process Server, a number of queue destinations are created for components of the mediation module that use asynchronous interactions. The queue destinations are used by the SCA runtime to hold messages being processed for the mediation module.

You can configure your servers and clusters so that the queue destinations are created on the server or server cluster that the mediation module is deployed to, or on a different server or cluster.

Note: You must complete this task before the server or cluster can be used to deploy mediation modules and host their queue destinations. Alternatively, you can configure the server or cluster to use a *remote destination location* for destinations, as described in "Configuring a server or cluster to use remote destinations for mediation modules" on page 185.

A server or cluster that hosts the queue destinations of mediation modules can be optimized for queue serving, and can provide those queues to a number of other servers and clusters on which mediation modules are deployed.

This configuration task defines that the server or cluster can host queue destinations for mediation modules deployed locally or for mediation modules deployed on other servers or clusters. This configuration also adds the server or cluster as a member of the SCA.SYSTEM bus, and configures the messaging engine for the server or cluster.

- If you configure a server, WebSphere can create a messaging engine with default properties. By default, the messaging engine is configured to use the default JDBC data source and Cloudscape JDBC Provider for its data store. This enables the messaging engine to run without needing any additional configuration.

  If you do not want to use the default data source configuration, you can choose to use a different data source or you can configure the data store to use a different JDBC provider.
- If you configure a cluster, you must explicitly create both the data store and the JDBC data source that the messaging engine uses to interact with the data store. You must also set the connection pool Purge policy to EntirePool.

To configure a server or cluster to host the queue destinations for mediation modules, use the administrative console to complete the following steps:

1. Display the Advanced Configuration page of the Configure your Network Deployment Environment guided activity.
   a. In the navigation pane, click **Guided Activities** → **Configure your Network Deployment Environment**

   b. In the content pane, click **Configure your Application Servers** → **Click to perform**
2. Select the cluster or server that you want to configure
   - Select a cluster name in the list under Select a cluster to configure
   - Select a server name in the list under Select a server to configure

   You can repeat this step several times to be able to configure several clusters and servers at the same time.
3. Click **Add** This adds the cluster or server to the list displayed. You can use the Remove button to remove entries from the list.
4. Select **Setup SCA Destination**.
5. Click **Next**. This displays the Choose your Common Database Template Settings page.
6. Specify details of the database used for messaging engine data stores.

   Specify your database provider type, and the user name and password to be used to access the database, to match what you configured for the data store in the prerequisite tasks.

   You can later change these settings, but you might want to choose your most common database type here to avoid typing in the same information again when configuring other clusters or servers. Skip this step if you plan to use many different database types for your setup.
7. Ensure that the Configure a destination location option is selected and that the Do not host SCA applications option is cleared.
8. Under Database Setup, specify details of the database used for messaging engine data stores. For a server cluster, you must complete this step. For a server, you only need to complete this step if the server does not use the default Cloudscape database.

   Specify your database provider type, and the user name and password to be used to access the database, to match what you configured for the data store in the prerequisite tasks.

   You can later change these settings, but you might want to choose your most common database type here to avoid typing in the same information again when configuring other clusters or servers. Skip this step if you plan to use many different database types for your setup.
9. Under System Bus, specify details of the data store for the messaging engine on the SCA.SYSTEM bus.
   a. **Optional:** If you want to use an existing data source, select the Use existing data source check box, then select the name from the drop down list.
   b. In the Schema name field, type the name of the database schema used to contain the tables for the SCA.SYSTEM bus data source. The default schema name is IBMWSSIB.
10. Under Application Bus, specify details of the data store for the messaging engine on the SCA.APPLICATION bus.
    a. **Optional:** If you want to use an existing data source, select the Use existing data source check box, then select the name from the drop down list.
    b. In the Schema name field, type the name of the database schema used to contain the tables for the SCA.SYSTEM bus data source. The default schema name is IBMWSSIB.

11. If you want messaging engines to create automatically the database tables for the data source, select the **Create tables** check box. The tables are created when the server hosting the messaging engine is started.) Otherwise, the database administrator must create the database tables.
12. Click **Next**.
13. Verify the value in the Emitter Factory Profile JNDI Name field. This field specifies the JNDI name of the event emitter profile factory that will be used to submit events to the events infrastructure.

    If you want to change the default configuration for emitting events to the CEI server, select the appropriate emitter factory profile JNDI name from the drop down list or type an alternative value in the text box provided. In a clustered environment, the value you select is used for all servers in the cluster.
14. Click **Next**.
15. Check the details on the Summary page. To configure the cluster or server, click **Finish**.

    If any of the details that you selected are incorrect, you can click **Previous**, to return to earlier pages and change the details.
16. Click **Finish**.

    The cluster or server is configured for the SCA runtime and for the CEI events emitter.

    If the cluster or server is the first one that you have configured for the deployment manager cell, the activity creates the SCA.SYSTEM and SCA.APPLICATION buses for the cell, and configures the messaging engines on those buses.

    When completed successfully, you should see messages like the following:

    ```
    The SCA installation for the following target finished successfully:
        cells:localhostCell01:clusters:ESBcluster1

    The CEI events emitter configuration for the following target finished
        successfully: cells:localhostCell01:clusters:ESBcluster1
    ```
17. Save your changes to the master configuration.

To display the SCA configuration for a cluster, you can use the Service Component Architecture panel:**Servers** → **Clusters** → *name* → **[Additional Properties] Service Component Architecture**. From this panel you can access the messaging engines that have been added to the SCA.SYSTEM and SCA.APPLICATION buses.

You can deploy mediation modules to another server or cluster that has been configured to use a *remote destination location*. If that other server or cluster has its Remote Destination Location property set to this server or cluster, the queue destinations for the mediation module are configured on this server or cluster.

You can tune the JDBC data source of a messaging engine for performance.

The user name and password that you specified for the data store are used to create an authentication alias whose name is specified on the data store configuration: **Buses** → *SCA.SYSTEM_bus_name* → **Messaging engines** → *me_name* → **Data store**. You can use that panel to change the data source JNDI name, schema name, and authentication alias.

Configuring a server or cluster to use remote destinations for mediation modules:

For mediation modules, a cluster or server can be configured so that the queue destinations are created on a different server or cluster.

Before starting this task, you must have already created one or more servers or server clusters required:

- The server or cluster on which you want to deploy mediation modules. This is the server or cluster to be configured in this task.
- The server or cluster on which you want to create queue destinations for the SCA runtime to exploit.

  Note: This server or cluster must have been configured to host queue destinations for mediation modules, as described in "Configuring a server or cluster to host queue destinations for mediation modules" on page 182.

You can use the same server or cluster in both cases.

When you install a mediation module into WebSphere Process Server, a number of queue destinations are created for components of the mediation module that use asynchronous interactions. The queue destinations are used by the SCA runtime to hold messages being processed for the mediation module.

You can configure your servers and clusters so that the queue destinations are created on the server or server cluster that the mediation module is deployed to, or on a different server or cluster.

Note: You must complete this task before the server or cluster can be used to deploy mediation modules that want their queue destinations hosted elsewhere. Alternatively, you can configure the server or cluster to host the queue destinations, as described in "Configuring a server or cluster to host queue destinations for mediation modules" on page 182.

A server or cluster that does not host the queue destinations for mediation modules can be optimized to run other components, and does not have to be a member of the SCA.SYSTEM bus.

This configuration task defines that the server or cluster uses queue destinations created on a different server or cluster. This configuration does not add the server or cluster as a member of the SCA.SYSTEM bus

To configure a server or cluster to use queue destinations created on a different server or cluster, use the administrative console to complete the following steps:

1. Display the Advanced Configuration page of the Configure your Network Deployment Environment guided activity.
   a. In the navigation pane, click **Guided Activities** → **Configure your Network Deployment Environment**
   b. In the content pane, click **Configure your Application Servers** → **Click to perform**
2. Select the cluster or server that you want to configure
   - Select a cluster name in the list under Select a cluster to configure
   - Select a server name in the list under Select a server to configure

   You can repeat this step several times to be able to configure several clusters and servers at the same time.

3. Click **Add** This adds the cluster or server to the list displayed. You can use the Remove button to remove entries from the list.
4. Select **Use Remote SCA Destination**. This option is selected by default.
5. Click **Next**. This displays the Choose your Common Database Template Settings page.
6. Specify details of the database used for messaging engine data stores.

   Specify your database provider type, and the user name and password to be used to access the database, to match what you configured for the data store in the prerequisite tasks.

   You can later change these settings, but you might want to choose your most common database type here to avoid typing in the same information again when configuring other clusters or servers. Skip this step if you plan to use many different database types for your setup.
7. Click **Next**. This displays the Use Remote Destination Location for Service Component Architecture page.
8. Select the cluster or server that is to host SCA destinations.
   a. Ensure that the Use a remote destination location is selected and that other options are cleared.
   b. Use the associated menu to select the name of the remote cluster or server that hosts SCA destinations.
9. Click **Next**.
10. Verify the value in the Emitter Factory Profile JNDI Name field. This field specifies the JNDI name of the event emitter profile factory that will be used to submit events to the events infrastructure.

    If you want to change the default configuration for emitting events to the CEI server, select the appropriate emitter factory profile JNDI name from the drop down list or type an alternative value in the text box provided. In a clustered environment, the value you select is used for all servers in the cluster.
11. Click **Next**.
12. Check the details on the Summary page. To configure the cluster or server, click **Finish**.

    If any of the details that you selected are incorrect, you can click **Previous**, to return to earlier pages and change the details.
13. Click **Finish**.

    The cluster or server is configured for the SCA runtime and for the CEI events emitter.

    When completed successfully, you should see messages like the following:

    ```
    The SCA installation for the following target finished successfully: cells:localhostCell01:cl

    The CEI events emitter configuration for the following target finished successfully: cells:lo
    ```
14. Save your changes to the master configuration.

To display the SCA configuration for a cluster, you can use the Service Component Architecture panel:**Servers** → **Clusters** → *name* → **[Additional Properties] Service Component Architecture**

If you deploy a mediation module to this server or cluster, the queue destinations for the mediation module are configured on the server or cluster specified by the Use a remote destination location property.

You can change the configuration of the server or cluster, add it as a member of a service integration bus, or take other actions to manage the server or cluster.

Managing ESB servers

Server configuration defines settings that control how a server provides services for running applications and their components. Administrators can create and configure servers in an existing server and bus environment.

- To create new servers, you need to have a managed node in a network deployment cell.

If you choose a Complete (default) installation for WebSphere Process Server, a default bus environment is created for you. This bus environment comprises a single server assigned to two service integration buses that are used for deploying SCA modules. You can also use the server to deploy J2EE components and resources, like servlets, enterprise Java beans, and JMS destinations.

The configuration of the single server might be adequate for your SCA modules. However, you might want to change the configuration; for example, to add extra server capacity, change the transport chains that provide networking services, or provide your own custom services.

For more information about managing servers, see the following sub-topics:

- Create servers.

  You can create servers using either the Create New Application Server wizard in the administrative console or the createApplicationServer wsadmin command.
- Manage available servers.

  After creating a server, you can monitor its runtime state, change its configuration, start and stop the server, and perform a range of other administrative tasks to manage the server.
- Configure transport chains.

  You need to configure transport chains to provide networking services to such functions as the service integration technologies, WebSphere Secure Caching Proxy, and the high availability manager core group bridge service.
- Develop custom services.

  To define a hook point to be run when a server or node agent starts and shuts down, you develop a custom service class and then configure a custom service instance. When the application server or node agent starts, the custom service starts and initializes.
- Define processes for a server.

  To enhance the operation of a server, you can define command-line information for starting or initializing an application server process. Such settings define run-time properties such as the program to run, arguments to run the program, and the working directory.
- Manage server use of the Java virtual machine.

  As part of configuring a server, you might define settings that enhance your system's use of the Java virtual machine (JVM).

Server templates:

A server template is used to define the initial configuration settings for a new application server. When you create a new application server, you select a server template for the type of server. The default template is used if you do not specify a different template when you create the server.

All servers in a server cluster are created with the same server template.

In a heterogenous cell, you can have managed nodes on a mixture of distributed and z/OS operating system platforms. After you select a managed node on which to create a server, the administrative console offers you the choice of templates that can be used for the operating system platform of that node.

The following default server templates are provided for managed nodes on the operating system platforms indicated in parentheses:

defaultWPSServer (distributed platforms) or defaultWPSServerZOS (z/OS platform)

Use this template if you want the server to support service applications for advanced integration capabilities such as business processes and state machines for endpoint orchestration, and business rules for dynamic decision making.

The server can also support:

- Service applications for mediation modules on an enterprise service bus, as in Enterprise Service Bus.
- Enterprise applications as in WebSphere Application Server.

defaultESBServer (distributed platforms) or defaultESBServerZOS (z/OS platform)

Use this template if you want the server to support service applications for mediation modules on an enterprise service bus.

The server can also support enterprise applications as in WebSphere Application Server.

default (distributed platforms) or defaultZOS (z/OS platform)

Use this template if you want the server to support enterprise applications as in WebSphere Application Server.

Instead of these default server templates, you can create your own templates based on another, already existing application server. This enables you to create a template with the same configuration properties as the existing server.

Creating a server:

You can create servers using either the Create New Application Server wizard in the administrative console or the createApplicationServer wsadmin command.

Before you can create a server, you must have created a custom profile then federated the node into a network deployment cell.

Before you can deploy an SCA module, you must create a server that is to run the SCA module.

You can deploy several SCA modules to the same server, so you only need to complete this task if you want to create a new server; for example, to handle more client connections or provide greater message throughput than the one server, or to separate the administration of SCA modules onto different servers.

To create a server, use the administrative console to complete the following steps:

1. Display the list of servers.

   In the navigation pane, expand **Servers** → **Application servers**.

2. Click **New**.
3. On the Create New Application Server page, follow the instructions to define your server.
   a. Select the node on which you want the server to run.

      The node must be a WebSphere Process Server managed node.
   b. Type a unique name for the server.

      The name must be unique within the node.
   c. Click **Next**
   d. Select the defaultESBServer template (or an existing server) to be used to create the server. The selected template or server defines the initial properties of the new server.
      - Instead of selecting a template, you can select an existing ESB server as a template. The new server inherits all properties of the selected server.
      - If you have selected a z/OS node, the template names offered indicate that platform.
      - For more information about the available server templates, see "Server templates" on page 188
   e. Click **Next**
   f. For the **Generate Unique Http Ports** option, choose whether the new server will have unique ports for each HTTP transport.

      By default, the **Generate Unique Http Ports** option is selected. If you select this option, you might need to update the alias list for the virtual host that you plan to use with this server to contain these new port values. If you deselect this option, ensure that the default port values do not conflict with other servers on the same physical machine.

      For more information about using unique HTTP ports, see Configuring HTTP transports.
   g. Click **Next**
   h. If you create the new server using an existing server as a template, do not select to map applications from the existing server to the new server. By default, this option is disabled.
   i. Click **Next**
   j. On the Confirm new server page, check the summary. To complete the server creation, click the **Finish**.

      If there are settings you wish to change, click **Previous** to review or change the server settings.
4. **Optional:** To use multiple language encoding support in the administrative console, configure an application server with UTF-8 encoding enabled.
5. Save your changes to the master configuration.

The server is created and appears in the list of servers on the Application Servers page.

You can now manage the server; for example, to start the server or deploy SCA modules.

Note: The server was created with default values for some of its properties. A server has many properties that can be set, and creating a server on the Create New Application Server page specifies values for only some of the important properties. To view all of the properties of your server, and to customize your server further, click on the name of your server on the

Application Servers page. This displays the server detail settings page, on which you can view and change properties as needed.

Creating a new WebSphere Process Server custom profile:

A custom profile is an empty node that you must federate into a network deployment cell to make operational. In contrast to a stand-alone server profile, a custom profile does not have a default server on its node. Nor are there any default applications on the custom node. Federating the custom profile changes it into a managed node. After federation, a custom profile has a nodeagent process but does not have a server process. You must use the administrative console of the deployment manager to customize the empty node for production or other uses. After you start the nodeagent, it responds to commands from the deployment manager.

Before using this procedure:

- Ensure you have an existing installation of WebSphere Process Server.
- Ensure you want to create a custom server profile and that you want to do so interactively. If your plan does not meet these criteria, see Creating and augmenting profiles using the Profile wizard for descriptions of other documented profile creation or augmentation procedures.
- Ensure you have enough disk and temporary space to create the new profile.

Profile	Required disk space	Required temp space
Deployment manager profile	30 MB	40 MB
Custom profile	10 MB	40 MB
Application server profile	200 MB	40 MB

- If you have enabled global security on your installation of WebSphere Process Server, ensure you disable it before creating the profile. For information on enabling and disabling global security, see Setting up WebSphere Process Server security.

The Full installation procedure for WebSphere Process Server creates one stand-alone server profile named `default` with a server named `server1`. However, to make use of the networking advantage of the product, you can use the Profile wizard to create additional profiles.

Use the following procedure to create a new custom profile.

1. Log on as root on a Linux or UNIX system, or as a member of the Administrator group on a Windows system.
2. Start the WebSphere Process Server Profile wizard to create a new runtime environment.

   See Starting the Profile wizard for descriptions of the many ways to start the Profile wizard on various platforms.

   The system starts InstallShield Multiplatform to run the Profile wizard and the Welcome panel is displayed.
3. Click **Next**. One of the following panels is displayed:
   - If a WebSphere Application Server or WebSphere Application Server Network Deployment profile exists, the Existing Profile Detection panel is displayed. This procedure assumes that you want to create a new profile, rather than augment an existing one into a WebSphere Process Server

profile. To create a new profile, select **Create a new WebSphere Process Server profile** and click **Next**. The Profile Type Selection panel is displayed.
- If no profile exists the Profile Type Selection panel is displayed.

4. Select **Custom WebSphere Process Server**, then click **Next**. The Federation panel is displayed.
5. Choose whether to federate the custom node into a deployment manager.
   - If you choose to federate the node, specify the host name or IP address and SOAP port of the deployment manager and click **Next** to continue. The Profile Name panel is displayed.

     Attention:

     Federate the custom node at this time only if all of the following are true:
     - The deployment manager is running.
     - The deployment manager is a WebSphere Process Server deployment manager profile.
     - Security is not enabled on the deployment manager node.
     - The deployment manager uses the default SOAP JMX connector type and the connector is enabled.

     Do *not* federate the custom node at this time if any one of the following is true:
     - The deployment manager is not running or you are not sure if it is running.
     - The deployment manager is a WebSphere Application Server Network Deployment profile that has not yet been augmented into a WebSphere Process Server profile.
     - Security is enabled on the deployment manager node.
     - The SOAP connector is disabled.
     - The deployment manager is reconfigured to use the non-default remote method invocation (RMI) as the preferred Java Management Extensions (JMX) connector. (Select **System administration > Deployment manager > Administration services** in the administrative console of the deployment manager to verify the preferred connector type.)

     If you federate a custom node when the deployment manager is not running or is not available for other reasons, the Profile Creation Failed panel is displayed. Click **Finish** to exit this panel.
   - If you choose not to federate the node, select the check box next to **Federate this node later using the addNode command** and click **Next** to continue. The Profile Name panel is displayed. See Federating custom nodes to a deployment manager for more information on how to federate a node from the administrative console or by using this command.
6. Specify a unique name for the profile or accept the default name. If any other profiles exist, you have the option of making this profile the default profile by selecting the check box next to **Make this profile the default**. Then click **Next**.

   Each profile that you create must have a unique name. When you have more than one profile, you can tell them apart at their highest level by this name. See Naming considerations for profiles, nodes, hosts, and cells for information on issues you must consider when naming the profile. The Profile Directory panel is displayed.
7. Accept the default directory location for the profile or use the **Browse** button to specify another location, then click **Next**.

This directory will contain the files that define the runtime environment, such as commands, configuration files, and log files. By default, this directory location is:

- Linux UNIX **On Linux and UNIX platforms:** `*install_root*/profiles/*profile_name*`
- Windows **On Windows platforms:** `*install_root*\profiles\*profile_name*`

where *profile_name* is the name that you specified. This directory location can be changed to any valid directory location on the system. If the directory already exists, it must be empty or the Profile wizard will display an error.

The Node And Host Names panel is displayed.

8. Specify the node and host names for the custom profile, or accept the defaults, then select **Next**.

   Use a unique name for each node that you create. See Naming considerations for profiles, nodes, hosts, and cells for information on reserved terms and other issues you must consider when naming the node and host.

   One of the following panels is displayed, depending on whether you elected to federate the node on the Federation panel in Step 5:
   - If you chose to federate the node later, the Profile Summary panel is displayed. In this case, proceed to Step 10.
   - If you chose to federate the node now, the Port Value Assignment panel is displayed. In this case, proceed to Step 9.
9. The Profile wizard will detect ports currently used by other WebSphere products, but if you have other applications which use specified ports, verify that the ports do not conflict. Click **Next**.

   When federating a custom profile, the **addNode** command uses non-conflicting ports. This means that you can take the default port assignments as you create the profile, and let the **addNode** command specify non-conflicting ports as you federate the node. Port assignments must be unique on a machine. Server processes on different machines can use the same port assignments without conflict. The Profile Summary panel is displayed.
10. Review the profile characteristics, and click **Next** to create the profile or **Back** to change the characteristics of the profile.

    The Profile wizard shows a progress panel to indicate that the profile is being created. If no errors are detected, the Profile Creation is Complete panel is displayed at the end of the process.

    Attention: If errors are detected during profile creation, other panels might be displayed in place of the Profile Creation is Complete panel, for example:
    - Profile Creation is Complete With Warnings panel, which indicates that a profile was created but warnings were generated.
    - Profile Creation is Incomplete panel, which indicates that a profile was created but exists in an unusable state.
    - Profile Creation Failed panel, which indicates that a profile was not created.

    Each of these panels identifies the log file which contains more details to help you to resolve the errors.
11. Ensure the check box to launch the First steps console is selected and click **Finish** to close the Profile wizard and start the First steps console.

A new custom profile exists. The node within the profile is empty until you federate the node (if you have not done so during profile creation) and use the deployment manager to customize the node.

Federate the node into the network deployment cell if you have not already done so as you created the custom node.

Configuring servers or clusters for the SCA runtime:

Before you can deploy a mediation module onto a server or server cluster, you must have configured that server or cluster for the SCA runtime required. This advanced configuration defines whether queue destinations exploited by the SCA runtime are hosted locally or on a remote server or cluster.

If you have created a stand-alone server profile, that server has already been configured for the SCA runtime, so do not perform this task for that server.

If you have created a server or server cluster in a network deployment to support SCA modules, you must complete this task to configure that server or cluster for the SCA runtime required.

Before starting this task, you must have already created one or more servers or server clusters required for the SCA runtime:

- The server or cluster on which you want to deploy mediation modules.
- The server or cluster on which you want to create queue destinations for the SCA runtime to exploit.

You can use the same server or cluster in both cases.

For more information about creating servers or server clusters, see the following task descriptions:

- Creating a server
- Creating a server cluster

When you install a mediation module into WebSphere Process Server, a number of queue destinations are created for components of the mediation module that use asynchronous interactions. The queue destinations are used by the SCA runtime to hold messages being processed for the mediation module.

To specify where the queue destinations are created, you complete one of the following configuration alternatives for the server or cluster.

- Set the advanced configuration of the server or cluster to host destinations. The configuration defines that this server or cluster can host queue destinations for mediation modules deployed locally or for mediation modules deployed on other servers or clusters. This configuration also adds the server or cluster as a member of the SCA.SYSTEM bus, and configures the messaging engine for the server or cluster.

  To perform this configuration task, see “Configuring a server or cluster to host queue destinations for mediation modules” on page 182.
- Set the advanced configuration of the server or cluster to use remote destinations. The configuration defines that this server or cluster uses a *remote destination location* for destinations.

  This server or cluster does not need to be a member of the SCA.SYSTEM bus.

  If you deploy a mediation module to this server or cluster, the queue destinations are configured on a different server or cluster that is a member of the SCA.SYSTEM bus.

  To perform this configuration task, see “Configuring a server or cluster to use remote destinations for mediation modules” on page 185.

The server or cluster is configured to accept the deployment of mediation modules.

You can now deploy mediations onto the server or cluster or take other actions to manage the server or cluster.

Managing available servers:

After creating a server, you can monitor its runtime state, change its configuration, start and stop the server, and perform a range of other administrative tasks to manage the server.

These tasks are for servers that you have already created.

The running of applications and services depends on the configuration and running of ESB servers. After creating a server, you can perform a range of other administrative tasks to manage the server.

You can manage servers using either the administrative console or a range of wsadmin commands.

This topic provides a summary of alternative ways to manage available servers with the administrative console, and provides links to where the associated tasks are described.

- Display the list of servers.

  In the navigation pane, click **Servers** → **Application Servers**.

  This displays the Application Servers collection page, which lists the servers, their cell, and the nodes holding the servers. For each server, this page also shows the status that indicates whether a server is started, stopped, or unavailable. You can use this page to perform a number of actions on servers; for example, to create new servers, start or stop servers, or open a page of detail settings for a server.
- Display details about a server.

  Display the list of servers on the Application Servers page, then click the name of the server.

  This displays the settings page for that server. You can use this page to view detailed information about a server and its resources, or to change the configuration of the server and its resources.
- ../../../com.ibm.websphere.nd.doc/info/ae/ae/trun_svr_start.html.

  To be able to run applications and services, a server must be started. If the server is in a network deployment cell, the node agent must already be running.

  – To start a node agent for a managed node, run the command `*wesb_install*\profiles\*profile_name*\bin\startNode` where *profile_name* is the name of the custom profile for the managed node.

  – To start a server, display the list of servers on the Application Servers collection page, select the checkbox for the server, then click **Start**.
- ../../../com.ibm.websphere.nd.doc/info/ae/ae/trun_watch_app.html.

  You must monitor the status of run-time components to ensure that after starting they remain operational as needed.

  You can browse messages displayed in the server logs. You can also use the **Troubleshooting** → **Logs and Trace** page of the administrative console to monitor the status of run-time components.
- ../../../com.ibm.websphere.nd.doc/info/ae/ae/trun_svr_stop.html.

Stopping a server prevents it being used to run applications and services.

Display the list of servers on the Application Servers collection page, select the checkbox for the server, then click **Stop**.

- "Deleting a server"

  Deleting a server deletes the server identity and configuration, including the deployment of applications and service on that server. The server is then no longer available to deploy applications and services.

  Display the list of servers on the Application Servers page, select the check box for the server, then click **Delete**.

Deleting a server:

Deleting a server deletes the server identity and configuration, including the deployment of applications and service on that server. The server is then no longer available to deploy applications and services.

If you no longer need a specific server to run applications and services, you can delete the server.

To delete a server, use the administrative console to complete the following steps:

1. In the navigation pane, click **Servers** → **Application Servers** This displays the Application Servers page, which lists the servers and shows the status that indicates whether a server is started, stopped, or unavailable.
2. Ensure that the server is stopped.

   To stop the server, select the check box next to the server name, then click **Stop**.
3. Delete the server

   To delete the server, select the check box next to the server name, then click **Delete**.
4. Save your changes to the master configuration.

The server is removed from the list on the Application Servers page, and is no longer available to deploy applications and services.

Balancing workloads with server clusters

A server cluster is a set of servers that are managed together and participate in workload management. Administrators can create and configure server clusters in an existing server and bus environment.

Before creating a clustered environment, you should take the following actions:

1. Plan and prepare your environment to use server clusters. For example, consider the following items:
   - Ensure that you have adequate resources to implement clustering successfully.
   - Determine if clustered servers would be beneficial for your applications. If an application has a large number of requests, installing on a cluster could help improve processing throughput. Other reasons to consider installing on a cluster includes availability of the application during scheduled maintenance and during equipment failures.
   - Determine if any of the applications contain services.
   - Familiarize yourself with Network Deployment and clustering support provided by WebSphere Application Server Network Deployment, version 6. For more information about the clustering support, see the related topics.

- Review this task before performing any of the steps.

2. Create a network deployment cell, by creating a Deployment Manager profile, one or more custom profiles, and federating the custom profiles into the deployment manager cell.

If you want to balance workload, such as service requests, over a set of servers, you can create a server cluster, then add servers as members of that cluster. You can also create a *backup cluster*, to provide failover support for the server cluster that it is assigned to.

To assist you in understanding how to configure and use clusters for workload management, consider this scenario. Client requests are distributed among the cluster members on a single machine. (A client refers to any service requester, servlet, Java application, or other program or component that connects the end user with a server.) In more complex workload management scenarios, you can distribute cluster members within the same sysplex.

For more information about managing server clusters, see the following sub-topics, which provide information related to using server clusters with enterprise service bus and provide links to appropriate WebSphere Application Server topics:

- Create a server cluster.

  For more information about creating a server cluster, see Creating a server cluster.
- Create several server members of the cluster.

  For more information about creating a server member, see “Creating a cluster member.”
- Managing available clusters.

  After creating a server cluster, you can monitor its runtime state, change its configuration, start and stop the cluster, and perform a range of other administrative tasks to manage the cluster.

Creating a cluster member:

Use this task to create a server as a member of a server cluster.

Before you can create a cluster member, you must have created the cluster, as described in Creating a server cluster.

If you want a server to take part in the distribution of workload in a server cluster, create a cluster member for the server.

To create a cluster member, use the administrative console to complete the following steps:

1. In the navigation pane, click **Servers → Clusters**. This displays the Server Cluster page, which lists the clusters in the network deployment cell.
2. In the content pane, click the name of the cluster.
3. Click **Cluster members**. This displays the Cluster Members page, which lists members of the cluster, states the nodes on which members reside, and states whether members are started, stopped, or encountering problems.
4. Click **New**, then follow the steps on the Create new cluster members page.
   a. For Member name, type the name of a new server to add as a member of the cluster.
   b. For Select node, select the node on which the server runs.

c. Make sure that Generate Unique HTTP Port is selected.

d. Under Select template:, select the defaultESBServer template.

   1) Select Default application server template

   2) From the drop down list, select defaultESBServer.

   - Instead of selecting a template, you can select an existing ESB server. The new server inherits all properties of the selected server.
   - If you have selected a z/OS node, the template names offered indicate that platform.
   - For more information about the available server templates, see "Server templates" on page 188

e. Click **Apply** to finish the cluster member. You can add more cluster members. All cluster members you add are based on the same server template, so there are less steps to complete.

f. Click **Next**.

g. Review the summary of information on new cluster members and click **Finish**.

5. Save your changes to the master configuration. As part of saving the change to the configuration, you can select Synchronize changes with Nodes before clicking **Save** on the Save page.
6. To display the properties of a cluster member, click the member's name under Member name on the Cluster members page. This displays the settings page for the cluster member instance.

You created application servers that became members of an existing server cluster.

You can display the properties and local topology of the cluster. Click **Servers** → **Clusters** → **[Content pane]** ***cluster_name***.

You can change the configuration of the cluster. Note that if you have not clicked **Save** and saved your administrative configuration, you only see the Configuration and Local Topology tabs. To see the Runtime tab, you must save your administrative configuration. Also, ensure that changes are synchronized to the nodes as described above.

You can create more cluster members or start the cluster. For more information about cluster configuration options, see Balancing workloads with clusters.

You can use scripting to automate the task of creating cluster members. For information about using scripting for clusters, see Creating clusters using scripting.

To manage cluster members, you can use the Cluster members page.. You can use this page to list the cluster members, and to select members to display more details or to act on.

You can use scripting to automate the task of creating cluster members. For information about using scripting for cluster members, see Creating clusters using scripting.

Configuring servers or clusters for the SCA runtime:

Before you can deploy a mediation module onto a server or server cluster, you must have configured that server or cluster for the SCA runtime required. This advanced configuration defines whether queue destinations exploited by the SCA runtime are hosted locally or on a remote server or cluster.

If you have created a stand-alone server profile, that server has already been configured for the SCA runtime, so do not perform this task for that server.

If you have created a server or server cluster in a network deployment to support SCA modules, you must complete this task to configure that server or cluster for the SCA runtime required.

Before starting this task, you must have already created one or more servers or server clusters required for the SCA runtime:

- The server or cluster on which you want to deploy mediation modules.
- The server or cluster on which you want to create queue destinations for the SCA runtime to exploit.

You can use the same server or cluster in both cases.

For more information about creating servers or server clusters, see the following task descriptions:

- Creating a server
- Creating a server cluster

When you install a mediation module into WebSphere Process Server, a number of queue destinations are created for components of the mediation module that use asynchronous interactions. The queue destinations are used by the SCA runtime to hold messages being processed for the mediation module.

To specify where the queue destinations are created, you complete one of the following configuration alternatives for the server or cluster.

- Set the advanced configuration of the server or cluster to host destinations. The configuration defines that this server or cluster can host queue destinations for mediation modules deployed locally or for mediation modules deployed on other servers or clusters. This configuration also adds the server or cluster as a member of the SCA.SYSTEM bus, and configures the messaging engine for the server or cluster.

  To perform this configuration task, see "Configuring a server or cluster to host queue destinations for mediation modules" on page 182.
- Set the advanced configuration of the server or cluster to use remote destinations. The configuration defines that this server or cluster uses a *remote destination location* for destinations.

  This server or cluster does not need to be a member of the SCA.SYSTEM bus.

  If you deploy a mediation module to this server or cluster, the queue destinations are configured on a different server or cluster that is a member of the SCA.SYSTEM bus.

  To perform this configuration task, see "Configuring a server or cluster to use remote destinations for mediation modules" on page 185.

The server or cluster is configured to accept the deployment of mediation modules.

You can now deploy mediations onto the server or cluster or take other actions to manage the server or cluster.

Managing available server clusters:

After creating a server cluster, you can monitor its runtime state, change its configuration, start and stop the cluster or cluster members, and perform a range of other administrative tasks to manage the cluster.

The running of applications and services depends on the configuration and running of the cluster and the servers that are members of the cluster. After creating a server cluster, you can perform a range of administrative tasks to manage the cluster.

You can manage clusters using either the administrative console or a range of wsadmin commands.

This topic provides a summary of alternative ways to manage available clusters with the administrative console, and provides links to where the associated tasks are described.

- Display the list of clusters.

  In the navigation pane, click **Servers** → **Clusters**.

  This displays the Server Cluster collection page, which lists the clusters. For each cluster, this page also shows the status that indicates whether a cluster is started, stopped, or unavailable. You can use this page to perform a number of actions on clusters; for example, to create new clusters, start or stop clusters, or open a page of detail settings for a cluster.
- Display details about a cluster.

  Display the list of clusters then, on the Server Cluster collection page, click the name of the cluster.

  This displays the settings page for that cluster. You can use this page to view or change the configuration and local topology of a server cluster. If you saved your administrative configuration after creating the server cluster, you can also view runtime information about the server cluster.
- Display the list of servers that are members of a cluster.

  Display details about a cluster then, on the Server Cluster settings page, click **Cluster members**.

  This displays the Cluster Members collection page, which lists the members of the cluster, states the nodes on which members reside, and states whether members are started, stopped or encountering problems. You can use this page to perform a number of actions on cluster members; for example, to create new cluster members, or open a page of detail settings for a cluster member.
- ../../com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm_member.html

  Display the Cluster Members collection page, then click **Cluster members**. The Cluster members page lists members of a cluster, states the nodes on which members reside, and states whether members are started, stopped or encountering problems.
- ../../com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm_cluster_start.html.

  To be able to run applications and services, a cluster must be started.

  Display the list of clusters then, on the Server Cluster collection page, select the checkbox for the cluster, then click **Start** or **Ripplestart**.
- ../../com.ibm.websphere.nd.doc/info/ae/ae/trun_watch_app.html.

  You must monitor the status of run-time components to ensure that after starting they remain operational as needed.

  You can browse messages displayed in the cluster logs. You can also use the **Troubleshooting** → **Logs and Trace** page of the administrative console to monitor the status of run-time components.

- Tune the behavior of the workload management run time

  If your application is experiencing problems with timeouts or your network experiences extreme latency, change the timeout interval for the `com.ibm.CORBA.RequestTimeout` property. Or, if the workload management state of the client is refreshing too soon or too late, change the interval for the `com.ibm.websphere.wlm.unusable.interval` property.
- Stop a cluster.

  Stopping a server cluster prevents it being used to run applications and services.

  Display the list of clusters then, on the Server Cluster collection page, select the checkbox for the cluster, then click **Stop** or **Immediate Stop**.
- ../../com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm_clustermember_delete.html

  Deleting a cluster member removes the server from the cluster, and deletes the server. You cannot delete a cluster member without deleting the server.

  Display the list of cluster members then, on the Cluster Members collection page, select the checkbox for the cluster member, then click **Delete**.
- ../../com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm_cluster_delete.html

  If you delete a server cluster, the action deletes the cluster and all its cluster members. When you delete a cluster, there is no option to keep certain cluster members or applications that you have installed on any part of the cluster

  Display the list of clusters then, on the Server Cluster collection page, select the checkbox for the cluster, then click **Delete**.

Doing more with bus topologies

The SCA.SYSTEM bus used for SCA modules is a standard service integration bus, as used in WebSphere Application Server. Besides the SCA.SYSTEM bus, you can also create other service integration buses to support service integration logic or other applications.

For example, the SCA.APPLICATION bus is provided and used to define JMS queue destinations and other JMS resources for modules deployed with JMS bindings.

You can create other buses for use as in WebSphere Application Server; for example, for applications acting as service requesters and providers within WebSphere Process Server, or to link a bus to WebSphere MQ. You can also use a WebSphere Process Server deployment manager to manage separate application servers, for use with applications and modules deployed onto WebSphere Application Server.

Detailed information about doing more with bus topologies is provided in the Service integration set of topics for WebSphere Application Server.

Managing 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.

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.

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.

Overview of the application integration environment:

A service application has an associated Service Component Architecture (SCA) module. You deploy service applications, to WebSphere Process Server, within EAR (enterprise archive) 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 other 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 following.

- The service application.
- The SCA module associated with the service application.
  - 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 include some of the following. The precise details you can display depend upon the SCA module.

- SCA module name.
- SCA module description.
- Associated application name.
- SCA module imports.
  - Interfaces.
    - Import interfaces are abstract definitions that describe how an SCA module accesses a service.
  - Bindings.
    - Import bindings are concrete definitions. They specify the physical mechanism by which an SCA module accesses a service. For example, using SOAP/HTTP.
- SCA module exports.
  - Interfaces.
    - Export interfaces are abstract definitions that describe how service requesters access an SCA module.
  - Bindings.
    - Export bindings are concrete definitions. They 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.

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. They allow you to route messages to different service providers. They also let you transform messages: you can amend message content or form. In addition, 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

Among the items that mediation modules contain are the following:

- Imports
  - Imports define interactions between SCA modules and service providers.
  - Imports allow SCA modules to call external services as if they were local.
  - Mediation module imports can be viewed from WebSphere Process Server, and if the import binding is a Web service or SCA binding then it can be modified.
- Exports
  - Exports define interactions between SCA modules and service requesters.
  - Exports allow an SCA module to offer a service. Exports define the external interfaces (access points) of an SCA module.
  - Mediation module exports can be viewed from WebSphere Process Server.
- SCA components
  - SCA components, or service components, are SCA building blocks. You build SCA modules such as mediation modules, using SCA components. You can create and customize SCA modules and components graphically, using WebSphere Integration Developer. In addition, you can customize some of the properties of SCA modules dynamically, from WebSphere Process Server administrative console, without having to redeploy the module.
  - Typically, 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, at most, one mediation flow component.
  - A mediation flow component can contain one mediation primitive, a number of mediation primitives or no mediation primitives. WebSphere Process Server supports a supplied set of mediation primitives that provide functionality for

message routing and transformation. One of the mediation primitives that WebSphere Process Server supports allows you to invoke custom logic.

  - A mediation module does not have to contain a mediation flow component. 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 need transforming to SOAP/HTTP before sending on.

Note: You can view mediation modules from WebSphere Process Server. You can also make certain changes to mediation modules from WebSphere Process Server. However, you cannot view or change the SCA components from inside a WebSphere Process Server module. Use WebSphere Integration Developer to customize SCA components.

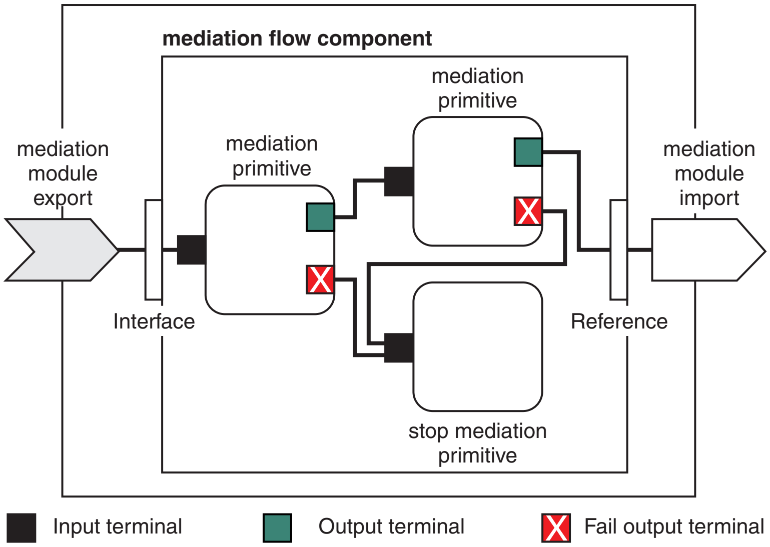


Figure 18. Simplified example of a mediation module. The mediation module contains one mediation flow component. The mediation flow component contains mediation primitives.

- Properties
  - Mediation primitives have properties, some of which can be displayed in the administrative console as additional properties of an SCA module.
  - In order 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, typically because modifying them affects the mediation flow in such a way that you need to redeploy the mediation module. 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. New invocations of mediation flows use property changes immediately, unless the changes occur in a deployment manager cell. If changes occur in a deployment manager cell then they take

effect after all nodes in the cell have been synchronized. Inflight invocations of mediation flows 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 EAR (enterprise archive) 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 JAR (Java archive) 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.

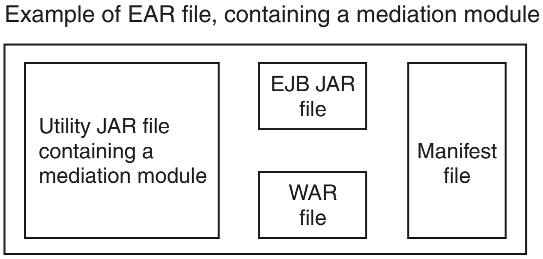

Example of EAR file, containing a mediation module
Utility JAR file containing a mediation module
EJB JAR file
WAR file
Manifest file

Figure 19. Simplified example of an EAR file containing a mediation module. The EAR file contains JARs. The utility JAR file contains a mediation module.

Imports and import bindings:

Imports define interactions between Service Component Architecture (SCA) modules and service providers. Imports let SCA modules access external services, (services that are outside the SCA module) as if they were local. Import bindings define the specific way that an external service is accessed.

SCA modules are not required to have imports, if they do not need to access external services. Mediation modules are types of SCA modules.

Interfaces and bindings

An SCA module import needs at least one interface, and an SCA module import has only one binding.

- Interfaces.
  - Import interfaces are abstract definitions. They define access points.
  - Import interfaces are defined using Web Services Description Language (WSDL), an XML language for describing Web services.

  - An SCA module can have many import interfaces.
- Bindings.
  - Import bindings are concrete definitions. They 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
  - Web Service bindings allow you to access web services.
  - The supported protocols are SOAP/HTTP and SOAP/JMS.
- SCA Bindings
  - 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
  - JMS 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
  - WebSphere MQ JMS bindings allow interoperability with WebSphere MQ based JMS providers.
  - You might have WebSphere MQ JMS bindings if you want to use WebSphere MQ as a JMS provider.
  - The JMS Message class and its five subtypes (Text, Bytes, Object, Stream, and Map) are automatically supported.
- WebSphere MQ Bindings
  - WebSphere MQ bindings allow interoperability with WebSphere MQ.
  - You might have WebSphere MQ bindings if you want to communicate with native WebSphere MQ applications.
  - 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.
- WebSphere Adapter Bindings
  - WebSphere adapters bindings enable interaction with Enterprise Information Systems (EIS).

Exports and export bindings:

Exports define interactions between Service Component Architecture (SCA) modules and service requesters. Exports let SCA modules offer services to others. Export bindings define the specific way that an SCA module is accessed by service requesters.

Mediation modules are types of SCA modules.

Interfaces and bindings

An SCA module export needs at least one interface.

- Interfaces.
  - Export interfaces are abstract definitions. They define access points.
  - Export interfaces are defined using Web Services Description Language (WSDL), an XML language for describing Web services.
  - An SCA module can have many export interfaces.
- Bindings.
  - Export bindings are concrete definitions. They specify the physical mechanism that service requesters use to access a service.
  - Typically, an SCA module export has one binding specified. An export with no binding specified is interpreted by the runtime as an export with an SCA binding.

Supported export bindings

enterprise service bus supports the following export bindings.

- Web Service Bindings
  - Web Service bindings allow you to access web services.
  - The supported protocols are SOAP/HTTP and SOAP/JMS.
- SCA Bindings
  - 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
  - JMS 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
  - WebSphere MQ JMS bindings allow interoperability with WebSphere MQ based JMS providers.
  - You might have WebSphere MQ JMS bindings if you want to use WebSphere MQ as a JMS provider.
  - The JMS Message class and its five subtypes (Text, Bytes, Object, Stream, and Map) are automatically supported.
- WebSphere MQ Bindings
  - WebSphere MQ bindings allow interoperability with WebSphere MQ.
  - You might have WebSphere MQ bindings if you want to communicate with native WebSphere MQ applications.
  - 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.
- WebSphere Adapter Bindings
  - WebSphere adapters bindings enable interaction with Enterprise Information Systems (EIS).

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 Service Component Architecture (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.
- 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.

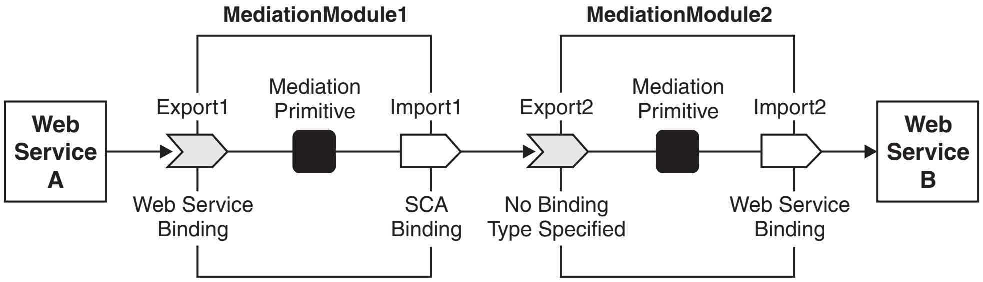


Figure 20. Example showing one mediation module interacting with another mediation module. MediationModule1 connects to MediationModule2

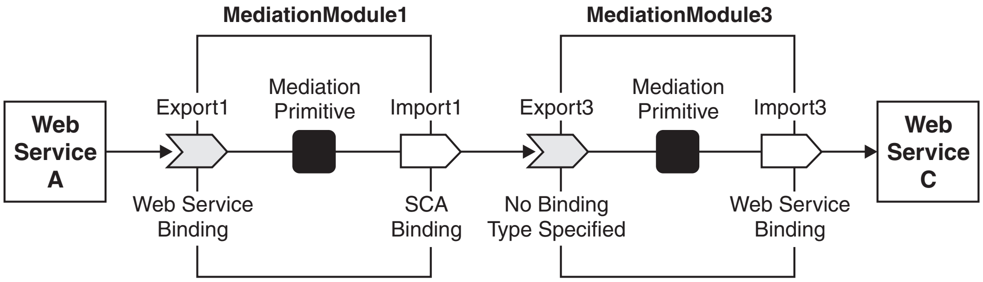


Figure 21. Example showing one mediation module interacting with another mediation module. MediationModule1 connects to MediationModule3

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 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 for z/OS for more information.

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. 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.

The application is now active and work should flow through the module.

Monitor the application to make sure the server is processing requests correctly.

Installing a mediation module EAR file with the console:

Installing a mediation module consists of moving the installable EAR file, for the mediation module, onto a server or cluster that will host the mediation module. Installed mediation modules that start and run properly are considered *deployed*.

If you have exported your mediation module to a JAR file then use the serviceDeploy command to create an installable EAR file from the mediation module JAR file. For more information about creating an installable EAR file for a mediation module, see “Installing a module on a production server” on page 209.

To enable WebSphere Process Server to use the functions provided by a mediation module to integrate applications and services, you need to install the EAR file for the module into a server or cluster, then start the deployed module.

This topic describes how to use the administrative console to install a mediation module EAR file. Alternatively, you can also use other methods, like the install or installinteractive command with the wsadmin tool, in the same way as you install enterprise application files into WebSphere Application Server.

Important: After you start performing the steps below, click **Cancel** to exit if you decide not to install the application. Do not simply move to another administrative console page without first clicking **Cancel** on an application installation page.

To use the administrative console to install a mediation module EAR file, complete the following steps:

1. Click **Applications** → **Install New Application** in the console navigation tree. The first of two Preparing for application installation pages is displayed.
2. On the first Preparing for application installation page, complete the following substeps:
   a. Specify the full path name of the mediation module EAR file (`.ear` file). The EAR file that you are installing can be either on the client machine (the machine that runs the administrative console Web browser) or on the server machine (the machine to which the client is connected). If you specify an EAR file on the client machine, then the administrative console uploads the EAR file to the machine on which the console is running and proceeds with application installation.
   b. Click **Next**.
3. On the second Preparing for application installation page, complete the following substeps:
   a. Select whether to generate default bindings.

Using the default bindings causes any incomplete bindings in the application to be filled in with default values. Existing bindings are not altered.

You can customize default values used in generating default bindings.

b. Click **Next**. If security warnings are displayed, click **Continue**. The Install New Application pages are displayed. If you chose to generate default bindings, and your application does not use a resource adapter, you can proceed to the Summary step (last step below).

4. On the Step: Select installation options panel, provide values for the following settings. For more information about the settings on this page, see Select installation options settings. Default values are used if you do not specify a value.

   a. For Directory to install application, specify the directory into which the application EAR file will be installed. The default value is the value of *APP_INSTALL_ROOT/cell_name*, where the *APP_INSTALL_ROOT* variable is `install_root/installedApps`. For example, for WebSphere Process Server installed on Windows into C:\Program Files\WESB\, the default location is C:\Program Files\WESB\profile_name\installedApps*cell_name*.

      Note: If an installation directory is not specified when an application is installed in a stand-alone profile, the application is installed in *APP_INSTALL_ROOT/base_cell_name*. If you add the stand-alone server to a network deployment cell, the cell name of the new server configuration becomes the cell name of the deployment manager node. If the `-includeapps` option is used for the addNode utility, then the applications that are installed prior to the addNode operation still use the installation directory *`APP_INSTALL_ROOT/base_cell_name`*. However, an application that is installed after the stand-alone server has been added to the network configuration uses the default installation directory *`APP_INSTALL_ROOT/network_cell_name`*. To move the application to the *`APP_INSTALL_ROOT/network_cell_name`* location upon running the addNode operation, you should explicitly specify the installation directory as `${APP_INSTALL_ROOT}/${CELL}` during installation. In such a case, the application files can always be found under *`APP_INSTALL_ROOT/current_cell_name`*.

   b. For Distribute application, specify whether WebSphere Process Server expands or deletes application binaries in the installation destination. The default is to enable application distribution. As a result, when you save changes in the console, application binaries for newly installed applications are expanded to the directory specified. The binaries are also deleted when you uninstall and save changes to the configuration. If you disable this option, then you must ensure that the application binaries are expanded appropriately in the destination directories of all nodes where the application is expected to run.

      Important: If you disable this option and you do not copy and expand the application binaries to the nodes, a later saving of the configuration or manual synchronization does not move the application binaries to the nodes for you.

   c. For Use Binary Configuration, specify whether the server or cluster uses the binding, extensions, and deployment descriptors located with the application deployment document, the `deployment.xml` file (default), or those located in the EAR file.

d. For **Application name**, type a name for the application. Application names must be unique within a cell and cannot contain characters that are not allowed in object names. For a list of characters that are not allowed in object names, see Object names

e. For **Create MBeans for resources**, specify whether to create MBeans for various resources (such as servlets or JSP files) within an application when the application is started. The default is to create MBean instances.

f. For **Enable class reloading**, specify whether to enable class reloading when application files are updated. The default is not to enable class reloading. Enabling class reloading sets reloadEnabled to `true` in the `deployment.xml` file for the mediation module. If a mediation module's class definition changes, the server run time stops and starts the application to reload application classes.

g. For **Reload interval in seconds**, specify the number of seconds to scan the application's file system for updated files. The default is the value of the reload interval attribute in the IBM extension (`META-INF/ibm-application-ext.xmi`) file of the EAR file. To enable reloading, specify a value greater than zero (for example, 1 to 2147483647). To disable reloading, specify zero (0).

   The reload interval specified here takes effect only if class reloading is enabled.

h. For **Deploy Web services**, specify whether the Web services deploy tool `wsdeploy` runs during application installation. The tool generates code needed to run applications using Web services. The default is not to run the `wsdeploy` tool. You must enable this setting if the EAR file contains modules using Web services and has not previously had the `wsdeploy` tool run on it, either from the **Deploy** menu choice of an assembly tool or from a command line.

i. For **Validate Input off/warn/fail,** specify whether WebSphere Process Server examines the application references specified during application installation or updating and, if validation is enabled, warns you of incorrect references or fails the operation. An application typically refers to resources using data sources for container managed persistence (CMP) beans or using resource references or resource environment references defined in deployment descriptors. The validation checks whether the resource referred to by the application is defined in the scope of the deployment target of that application. Select **off** for no resource validation, **warn** for warning messages about incorrect resource references, or **fail** to stop operations that fail as a result of incorrect resource references.

j. For **Process embedded configuration**, specify whether the embedded configuration should be processed. An embedded configuration consists of files such as `resource.xml` and `variables.xml`. When selected or `true`, the embedded configuration is loaded to the application scope from the `.ear` file. If the `.ear` file does not contain an embedded configuration, the default is `false`. If the `.ear` file contains an embedded configuration, the default is `true`.

5. On the Step: Map modules to servers panel, for every module select a target server or cluster from the Clusters and Servers list. Select the check box beside Module to select the mediation module.

   If the application uses a WebSphere Adapter, specify target servers or clusters for each RAR file. Also map all other modules that use the resource adapters defined in the RAR modules to the same targets.

Note: When installing a RAR file onto a server, WebSphere Process Server looks for the manifest (MANIFEST.MF) for the connector module. It looks first in the connectorModule.jar file for the RAR file and loads the manifest from the _connectorModule.jar file. If the class path entry is in the manifest from the connectorModule.jar file, then the RAR uses that class path. To ensure that the installed connector module finds the classes and resources that it needs, check the Class path setting for the RAR using the console. For more information about the Class path setting, see the Resource Adapter settings and WebSphere relational resource adapter settings for the administrative console.

You can specify Web servers as targets that route requests to the application. The plug-in configuration file `plugin-cfg.xml` for each Web server is generated based on the applications which are routed through it. If you want a Web server to serve the application, use the **Ctrl** key to select an application server or cluster and the Web server together in order to have the plug-in configuration file `plugin-cfg.xml` for that Web server generated based on the applications which are routed through it.

6. If your application defines resource references, for Step: Map resource references to resources, specify JNDI names for the resources that represent the logical names defined in resource references. Each resource reference defined in the application must be bound to a resource defined in your WebSphere Process Server configuration before clicking on **Finish** on the Summary panel.
   a. **Optional:** Specify login configuration name and authentication properties for the resource.
   b. Click **OK** to save the values and return to the mapping step.
7. If your application uses Web modules, for **Step: Map virtual hosts for web modules**, select a virtual host from the list that should map to a Web module defined in the application.

   The port number specified in the virtual host definition is used in the URL that is used to access artifacts such as servlets and JSP files in the Web module. Each Web module must have a virtual host to which it maps. Not specifying all needed virtual hosts will result in a validation error displaying after you click **Finish** on the Summary panel.
8. If the application has security roles defined in its deployment descriptor then, for **Step: Map security roles to users/groups**, specify users and groups that are mapped to each of the security roles.

   Select **Role** to select all of the roles or select individual roles. For each role, you select one of the following choices for how security should be applied:

Option	Description
Everyone	This is equivalent to no security.
All authenticated	Anyone who authenticates with a valid user name and password is a member of the role.
Mapped users	Individual users are listed as members of the role.
Mapped groups	Groups are the most convenient way to add the users, every member of the identified groups becomes a member of the role.

For **Mapped users** or **Mapped groups**, to select specific users or groups from the user registry, complete the following sub-steps:

a. Select a role and click **Lookup users** or **Lookup groups**.

   b. On the Lookup users/groups panel displayed, enter search criteria to extract a list of users or groups from the user registry.
   c. Select individual users or groups from the results displayed.
   d. Click **OK** to map the selected users or groups to the role selected on the **Step: Map security roles to users/groups** panel.

9. If the application has Run As roles defined in its deployment descriptor, for **Step: Map RunAs roles to user**, specify the Run As user name and password for every Run As role. Run As roles are used by enterprise beans that must run as a particular role while interacting with another enterprise bean. Select **Role** to select all of the roles or select individual roles. After selecting a role, enter values for the user name, password, and verify password and click **Apply**.
10. If your application contains resource environment references, for **Step: Map resource environment references to resources**, specify JNDI names of resources that map to the logical names defined in resource environment references. If each resource environment reference does not have a resource associated with it, after you click **Finish** a validation error is displayed.
11. If your application defines **Run-As Identity** as *System Identity*, for **Step: Replace RunAs System to RunAs Roles**, you can optionally change it to *Run-As role* and specify a user name and password for the Run As role specified. Selecting *System Identity* implies that the invocation is done using the WebSphere Application Server security server ID and should be used with caution as this ID has more privileges.
12. If your application has resource references that map to resources that have an Oracle database doing backend processing, for **Step: Specify the isolation level for Oracle type provider**, specify or correct the isolation level to be used for such resources when used by the application. Oracle databases support ReadCommitted and Serializable isolation levels only.
13. If your application contains a module that has promoted properties, these are displayed along with their current values; you can modify the current values at this step, if required. Any modified values are validated against the type of the property and any other associated constraints.
14. On the Summary panel, verify the cell, node, and server onto which the application modules will install:
    a. Beside **Cell/Node/Server**, click **Click here**.
    b. Verify the settings.
    c. Click **Finish**.

Several messages are displayed, indicating whether your application file is installing successfully.

If you receive an OutOfMemory exception and the source application file does not install, your system might not have enough memory or your application might have too many modules in it to install successfully onto the server. If lack of system memory is not the cause of the exception, package your application again so the `.ear` file has fewer modules. If lack of system memory and the number of modules are not the cause of the exception, check the options you specified on the Java Virtual Machine page of the application server running the administrative console. Then, try installing the application file again.

▶Windows During installation certain application files are extracted in the directory represented by the configuration session and, when the configuration is saved, these files are saved in the WebSphere Application Server configuration repository.

On Windows machines, there is a limit of 256 characters for file paths. Therefore, the application installation might fail if the path for application files in the configuration session or in the configuration repository exceeds the limit of 256 characters. You might see FileNotFound exceptions with *path name too long* in the message. To overcome such problems, make application names and module URI names shorter in length, which helps reduce the file path length. Then, try installing the application file again.

After the application file installs successfully, complete the following actions:

1. Associate any shared libraries that the application needs to the application.
2. Save the changes to your configuration. The application is registered with the administrative configuration and application files are copied to the target directory, which is `install_root/installedApps/cell_name` by default or the directory that you designate. When installed into a Network Deployment profile, files are copied to remote nodes when the configuration on the deployment manager synchronizes with the configuration on individual nodes.
3. If the module is deployed on a server cluster, click **Rollout Update** on the Enterprise Applications page to propagate the changed configuration on all members of the cluster. Rollout Update sequentially updates the configuration on the nodes that contain cluster members.

To enable WebSphere Process Server to use the functions provided by a mediation module to integrate applications and services, the deployed module must be started. You can start the module manually or configure it to start automatically. You can also administer the module in other ways; for example, to change the configuration of the module, to stop or update the module, and otherwise manage its activity.

Deploying applications using ANT tasks:

This topic describes how to use ANT tasks to automate the deployment of applications to WebSphere Process Server. By using ANT tasks, you can define the deployment of multiple applications and have them run unattended on a server.

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 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 for z/OS contains a section on application programming interfaces. 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.

If you need to install multiple applications concurrently, develop an ANT task before deployment. The ANT task can then deploy and install the applications on the servers without your involvement in the process.

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 ANT task to run the ServiceDeploy command to create the EAR file for each server.
5. Create an ANT task to run the InstallApplication command for each EAR file from step 4 on the applicable servers.
6. Run the ServiceDeploy ANT task to create the EAR file for the applications.
7. Run the InstallApplication ANT task to install the EAR files from step 6.

The applications are correctly deployed on the target servers.

Example of deploying an application unattended

This example shows an 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 ANT task.

```
${WAS}/bin/ws_ant -f myBuildScript.xml
```

Tip: Multiple applications can be deployed unattended by adding additional project statements into the file.

Use the administrative console to verify that the newly installed applications are started and processing the workflow correctly.

Managing mediation modules

You can list the mediation modules that have been deployed to WebSphere Process Server. You can also view information associated with individual mediation modules and make changes to some import bindings.

After deploying service applications, you can view and manage the associated Service Component Architecture (SCA) modules. Mediation modules are types of Service Component Architecture (SCA) modules.

Working with mediation modules:

You can list the mediation modules that have been deployed to WebSphere Process Server. You can also start or stop mediation modules, and view details of a mediation module or its application.

To display the mediation modules that you have deployed, use the administrative console to complete the following steps.

Note: Mediation modules are types of Service Component Architecture (SCA) modules.

List the SCA modules. In the navigation pane, expand **Applications** → **SCA Modules**.

This displays, in the content pane, 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.

Displaying details of a mediation module:

You can display information on mediation modules that have been deployed to WebSphere Process Server.

To display details about the mediation modules that you have deployed, use the administrative console to complete the following steps.

1. List the SCA modules. In the navigation pane, expand **Applications** → **SCA Modules**.
2. Choose the particular SCA module that you are interested in. In the content pane, click an SCA module.

This displays, in the content pane, the SCA module name and description; and the name of the associated enterprise application. Also displayed are 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.

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.

1. List the SCA modules. In the navigation pane, expand **Applications** → **SCA Modules**.
2. For the SCA module that you are interested in, click the application name.

   The application name is listed in the column labelled Application.

This displays, in the content pane, the application details page that provides the application's configuration properties and, if appropriate, local topology. From this page, you can modify property values and link to additional console pages to further review and change the application's configuration.

You can review and, if needed, change the configuration properties for the application, as described in Configuring an application.

Starting and stopping mediation modules:

You can start a mediation module that is not running (has a status of Stopped) or stop an module that is running (has a status of Started). To change the status of a mediation module, you start or stop the application used to deploy the module.

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" on page 209. This installs the application onto an appropriate server (or server cluster).

To use the services of a mediation module and associated components, you start the associated application. By default, the application starts automatically when the server starts.

You can start and stop applications manually 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, you can use the administrative console to complete the following steps.
1. List the SCA modules. In the navigation pane, expand **Applications** → **SCA Modules**.
2. Select the check box for the SCA module that you want started or stopped.
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*.

To restart a running application, select the application you want to restart, click **Stop** and then click **Start**.

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.

You might want to check that property values are what you expect before running a service application.

To display the properties of mediation modules that you have deployed, use the administrative console to complete the following steps.
1. List the SCA modules. In the navigation pane, expand **Applications** → **SCA Modules**.
2. Choose an SCA module. In the content pane, click an SCA module.

3. List the SCA module properties. In the content pane, under Additional Properties, click **Module Properties**.

This displays, in the content pane, the updatable properties for the SCA module. The properties are displayed 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 can use WebSphere Integration Developer. If there are no properties that you can update, then a message is displayed.

Changing mediation module properties:

You can change the value of some mediation module properties.

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.

1. List the SCA modules. In the navigation pane, expand **Applications** → **SCA Modules**.
2. Choose an SCA module. In the content pane, click an SCA module.
3. Display the SCA module properties. In the content pane, under Additional Properties, select **Module Properties**. This displays the module properties that you can update. Property names, types and values are displayed. However, you can only update property values from the administrative console, to change property names and types you can use WebSphere Integration Developer.
4. Choose a property value. Click a property value from the Properties table.
5. Change a property value. Enter a value that conforms to the property type.
6. Save your changes. Click **OK**. Then save your changes to the master configuration.

Changes a property value. New invocations of mediation flows use property changes immediately, unless the changes occur in a deployment manager cell. If changes occur in a deployment manager cell then they take affect after all nodes in the cell have been synchronized. In-flight invocations of mediation flows continue to use previous values.

Note: If you want to change property names and types, rather than property values, you should use WebSphere Integration Developer.

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 SCA import bindings and Web service (WS) import bindings.

To list the imports of mediation modules that you have deployed, use the administrative console to complete the following steps.

1. List the SCA modules. In the navigation pane, expand **Applications** → **SCA Modules**.
2. Select the particular SCA module that you are interested in.
3. List the SCA module imports. In the content pane, under Module components, expand **Imports**.

A list of imports is displayed, in the content pane. If there are no imports then a information message is displayed.

Displaying an import interface:

You can display the import interfaces of mediation modules that have been deployed to WebSphere Process Server.

To display the import interfaces of mediation modules that you have deployed, use the administrative console to complete the following steps.

1. List the SCA modules. In the navigation pane, expand **Applications** → **SCA Modules**.
2. Select an SCA module.
3. List the SCA module imports. In the content pane, under Module components, expand **Imports**.
4. Display import details. In the content pane, under Module components expand the import you require.
5. Display import interfaces. Expand **Interfaces**.
6. Select an interface.

The WSDL (Web Services Description Language) interface is displayed, in the content pane.

Displaying an import binding:

You can display some types of import bindings, after you deploy mediation modules to WebSphere Process Server.

To display the import bindings of mediation modules, use the administrative console to complete the following steps.

1. List the SCA modules. In the navigation pane, expand **Applications** → **SCA Modules**.
2. Select an SCA module.
3. List the SCA module imports. In the content pane, under Module components, expand **Imports**.
4. Display import details. In the content pane, under Module components expand the import you require.
5. Display import bindings. Expand **Binding**. If the binding is of type Web service, SCA, JMS or Adapter then the binding type is displayed. Other binding types are not displayed.
6. Select a binding. If the binding is a Web service import binding or an SCA import binding then you can select the binding to get import binding details.

The import binding details are displayed, in the content pane. The details displayed depend on the type of binding.

If the binding is a Web service import binding then the service name, port name and endpoint URL are displayed.

If the binding is an SCA import binding then the details displayed include the current target module. Also displayed are the current target export and its interfaces. Other target modules that have been deployed to WebSphere Process Server are displayed in a drop down menu. If you select a different target module then the list of target exports, and export interfaces, changes. If you select a different target export then the list of export interfaces changes.

Changing an SCA import binding:

In some cases, you can change the import bindings of mediation modules from WebSphere Process Server. You can change import bindings if they are SCA import bindings or web service import bindings, but not if they are other types of bindings.

To change SCA import bindings of mediation modules, use the administrative console to complete the following steps.

1. List the SCA modules. In the navigation pane, expand **Applications** → **SCA Modules**.
2. Choose an SCA module.
3. List the SCA module imports. In the content pane, under Module components, expand **Imports**.
4. Display import details. In the content pane, under Module components expand the import you require.
5. Display import bindings. Expand **Binding**.
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. Choose an export. Select an export from the **Export** drop-down menu.
9. Save your changes to the master configuration.

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.

Changing a web service import binding:

In some cases, you can change the import bindings of mediation modules from WebSphere Process Server. You can change import bindings if they are SCA import bindings or web service import bindings, but not if they are other types of bindings.

To change web service import bindings of mediation modules, use the administrative console to complete the following steps.

1. List the SCA modules. In the navigation pane, expand **Applications** → **SCA Modules**.
2. Choose an SCA module.
3. List the SCA module imports. In the content pane, under Module components, expand **Imports**.

4. Display import details. In the content pane, under Module components, expand the import you require.
5. Display import bindings. Expand **Binding**.
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.

The web service import binding is changed for the selected SCA module import.

Changes take effect after you update the master configuration and restart the SCA module.

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.

To list the exports of mediation modules that you have deployed, use the administrative console to complete the following steps.

1. List the SCA modules. In the navigation pane, expand **Applications** → **SCA Modules**.
2. Select the particular SCA module that you are interested in.
3. List the SCA module exports. In the content pane, under Module components, expand **Exports**.

A list of exports is displayed, in the content pane. If there are no exports then an information message is displayed.

Displaying an export interface:

You can display the export interfaces of mediation modules that have been deployed to WebSphere Process Server.

To display the export interfaces of mediation modules that you have deployed, use the administrative console to complete the following steps.

1. List the SCA modules. In the navigation pane, expand **Applications** → **SCA Modules**.
2. Select an SCA module.
3. List the SCA module exports. In the content pane, under Module components, expand **Exports**.
4. Display export details. In the content pane, under Module components expand the export you require.
5. Display export interfaces. Expand **Interfaces**.
6. Select an interface.

The WSDL (Web Services Description Language) interface is displayed, in the content pane.

Displaying an export binding:

You can display some types of export bindings, after you deploy mediation modules to WebSphere Process Server.

To display the export bindings of mediation modules, use the administrative console to complete the following steps.

1. List the SCA modules. In the navigation pane, expand **Applications** → **SCA Modules**.
2. Select an SCA module.
3. List the SCA module exports. In the content pane, under Module components, expand **Exports**.
4. Display export details. In the content pane, under Module components expand the export you require.
5. Display export bindings. Expand **Binding**. If the binding is of type Web service, SCA, JMS or Adapter then the binding type is displayed in square brackets. If the binding is of type MQ or MQ JMS then empty square brackets are displayed. If an export has no binding specified then the administrative console displays a message indicating that there is no binding. In this case the runtime assumes that the binding is of the default SCA type.
6. Select a binding. If the binding is a Web service export binding then you can select the binding to get export binding details.

The export binding details are displayed, in the content pane. A Web service export binding displays service and port names.

Managing access to WSRR

You can create, configure, and view all your WebSphere Service Registry and Repository (WSRR) access definitions using the administrative console.

Install and configure WSRR as described in the WSRR information center at http://publib.boulder.ibm.com/infocenter/sr/v6r0/topic/com.ibm.sr.doc/cwsr_overview_overview.html

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 invoke. A WSRR definition and its connection properties is the mechanism used to connect to a registry instance and look up a Web service to invoke. To display a list of all your WSRR definitions, use the administrative console to complete the following steps.

In the navigation pane, expand **Service integration** → **WSRR definitions**.

This displays, in the content pane, 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.

Click on a WSRR definition to display the general properties for that definition. From the content pane, you can also access the detailed connection properties.

Creating a new WSRR definition:

You can create a new WSRR definition using the administrative console.

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.

1. In the navigation pane, expand **Service integration** → **WSRR definitions**. 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 can optionally 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
   Currently the only choice is **Web service**.
4. Click **Apply** to save these properties.
5. Click **Connection properties**, under Additional properties in the content pane. The Connection properties configuration page is displayed.
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.
7. Click **Apply** to save these properties.
8. Click on **Save** to apply your changes to the master configuration. Click **Save**.

A new WSRR definition is created.

Deleting a WSRR definition:

You can delete a WSRR definition from the administrative console.

To delete a WSRR definition, use the administrative console to perform the following steps.

1. In the navigation pane, expand **Service integration** → **WSRR definitions**. The WSRR definitions page is displayed in the content pane, which shows 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. Click **Save**.

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, first set a different definition to be the default.

Setting a default WSRR definition:

You can set a WSRR definition to be the default definition from the administrative console.

To set a WSRR definition to be the default, use the administrative console to perform the following steps.

1. In the navigation pane, expand **Service integration** → **WSRR definitions**. The WSRR definitions page is displayed in the content pane, which shows 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** now displays in the Default column for that WSRR definition.
3. Click on **Save** to apply your changes to the master configuration. Click **Save**.

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.

Managing 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 if appropriate take further action to manage the throughput of SCA requests.

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 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 can use the CloudView tool to view data in a Cloudscape data store for a messaging engine, you must have stopped the messaging engine.

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.

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...**.

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.

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:
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 mediation modules

Besides using the WebSphere administrative console to manage mediation modules themselves, you can manage resources used by mediation modules and manage applications that contain mediation modules; you can also use commands to do these tasks.

The more routine tasks for managing mediation modules are described in "Managing mediation modules" on page 216.

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 make use of 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 into 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*.

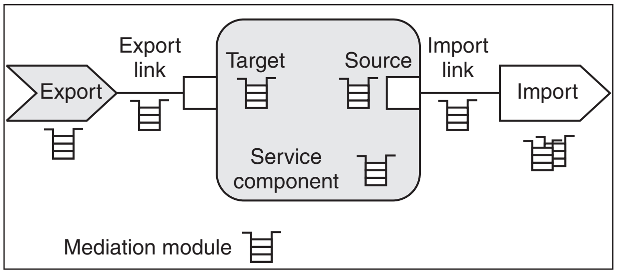
Queue sca/*module_name***/import/***import_name*
: This is the destination used to buffer asynchronous requests sent to import *import_name*.

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 defined on the messaging engine of the 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. For mediation modules, JMS support is provided by the default messaging provider of WebSphere Application Server or by WebSphere MQ via an MQ JMS binding. 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.

Using asynchronous messaging:

These topics describe how to use asynchronous messaging with WebSphere Process Server, to enable enterprise applications to use JMS resources and message-driven beans.

WebSphere Process Server supports asynchronous messaging as a method of communication based on the Java Message Service (JMS) programming interface. This JMS support is provided by one or more *JMS providers*, and associated services and resources, that you configure for use by enterprise applications. You can deploy EJB 2.1 applications that use the JMS 1.1 interfaces and EJB 2.0 applications that use the JMS 1.0.2 interfaces. This support is provided by the WebSphere Application Server on which WebSphere Process Server is built.

You can use the WebSphere administrative console to administer the WebSphere Process Server support for asynchronous messaging. For example, you can configure messaging providers and their resources, and can control the activity of messaging services.

For more information about implementing WebSphere enterprise applications that use asynchronous messaging, see the following sub-topics in the infocenter. Alternatively, you can see Administering applications and their environment and other PDF books available through theWebSphere Application Server Network Deployment, version 6 library Web page at http://www-306.ibm.com/software/webservers/appserv/was/library/.

- Learning about messaging with WebSphere Application Server

  Use this topic to learn about the use of asynchronous messaging for enterprise applications with WebSphere Process Server.
- Installing and configuring a JMS provider

  This topic describes the different ways that you can use JMS providers with WebSphere Process Server. A JMS provider enables use of the Java Message Service (JMS) and other message resources in WebSphere Process Server.
- Using the default messaging provider

  This topic is the entry-point into a set of topics about enabling WebSphere applications to use messaging resources provided by the default messaging provider. The default messaging provider is installed and runs as part of WebSphere Application Server, and is based on service integration technologies.
- Maintaining Version 5 default messaging resources

  This topic is the entry-point into a set of topics about maintaining messaging resources provided for WebSphere Application Server, version 5 applications by the default messaging provider.

- Using JMS resources of WebSphere MQ

  This topic is the entry-point into a set of topics about enabling WebSphere applications to use JMS resources provided by WebSphere MQ.
- Using JMS resources of a generic provider

  This topic is the entry-point into a set of topics about enabling WebSphere applications to use JMS resources provided by a generic messaging provider (other than a WebSphere default messaging provider or WebSphere MQ).
- Administering support for message-driven beans

  Use these tasks to manage resources used to support message-driven beans. These tasks are in addition to the tasks for administering resource adapters, JMS providers and the resources they provide.
- Troubleshooting WebSphere messaging

  Use this overview task to help resolve a problem that you think is related to the WebSphere Messaging. To identify and resolve problems that you think are related to WebSphere Messaging, you can use the standard WebSphere Process Server troubleshooting facilities.

Using commands to manage service applications:

You can manage service applications using commands. The commands can be used within scripts.

You must use the wsadmin tool to run service application commands.

You can use the wsadmin tool in different ways. You can use the tool interactively, as an individual command or in a script. Running multiple commands in a script can be useful if you are administering multiple machines.

WebSphere Process Server has commands that let you display SCA modules, and their imports and exports. You can also make changes to Web service and SCA import bindings.

1. List the SCA administration commands. `$AdminTask help SCAAdminCommands`
2. Display detailed help about a given command. `$AdminTask help command_name`

```
$AdminTask help listSCAModules
```

Managing mediation modules:

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.

You must use the wsadmin tool to run WebSphere Process Server commands.

You can run commands individually or in a script. Running multiple commands in a script is useful if you are administering multiple machines, or producing regular reports.

Listing mediation modules:

You can use a command to list the mediation modules that have been deployed to WebSphere Process Server.

You must use the wsadmin tool to run the following command.

To list the mediation modules that you have deployed, use the wsadmin tool to complete the following step.

Mediation modules are types of Service Component Architecture (SCA) modules. Therefore, to list the mediation modules you have deployed, you list the SCA modules.

List the deployed SCA modules. `$AdminTask listSCAModules`

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:

You can use a command to display attributes of mediation modules.

You must use the wsadmin tool to run the following commands.

To display the description of a mediation module, use the wsadmin tool to complete the following steps.

In order to show the description of a particular mediation module, you need to know the mediation module name.

Note: Mediation modules are types of Service Component Architecture (SCA) modules.

The listSCAModules command lists SCA modules that have been deployed to WebSphere Process Server.

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.

Displays the name and description of a particular SCA module.

```
$AdminTask showSCAModule {-moduleName myModule -applicationName
myApplication}
```

Displaying the properties of a mediation module:

You can use a command to show the properties for a specified mediation module.

You must use the wsadmin tool to run the following commands.

To display the properties of a mediation module, use the wsadmin tool to complete the following steps.

In order to show the properties of a particular mediation module, you need to know the mediation module name.

Note: Mediation modules are types of Service Component Architecture (SCA) modules.

The listSCAModules command lists SCA modules that have been deployed to WebSphere Process Server.

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.

Displays the properties of a particular SCA module.

```
$AdminTask showSCAModuleProperties {-moduleName myModule -applicationName
myApplication}
```

Changing a mediation module property:

You can use a command to change a property value for a specified mediation module.

You must use the wsadmin tool to run the following commands.

The listSCAModules command lists all SCA modules that have been deployed to WebSphere Process Server.

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.

Changes the property value for a particular SCA module property.

```
$AdminTask modifySCAModuleProperty {-moduleName myModule -applicationName
myApplication -propertyName myPropertyName -newPropertyValue
myNewPropertyValue}
```

Listing imports:

You can use a command to list the imports of any mediation module that has been deployed to WebSphere Process Server.

You must use the wsadmin tool to run the following commands.

To list the imports of a mediation module, use the wsadmin tool to complete the following steps.

In order to list the imports of a mediation module you need to know the name of the mediation module.

Note: Mediation modules are types of Service Component Architecture (SCA) modules.

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.

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.

Lists the imports for a particular SCA module.

```
$AdminTask listSCAImports {-moduleName myModule -applicationName
myApplication}
```

Displaying details of an import:

You can use a command to display import details of a mediation module that has been deployed to WebSphere Process Server.

You must use the wsadmin tool to run the following commands.

To display the import details of mediation module, use the wsadmin tool to complete the following steps.

In order to show the details of a particular mediation module import, you need to know the mediation module name and the import name.

Note: Mediation modules are types of Service Component Architecture (SCA) modules.

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.

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}`

Note: In addition to specifying the *moduleName* and *importName*, you have the option of specifying the *applicationName*. Providing an *applicationName* improves performance.

Displays the import details for a particular SCA module import.

```
$AdminTask showSCAImport {-moduleName myModule -applicationName
myApplication -import myImport}
```

Displaying an import binding:

You can use a command to display the import bindings of a mediation module that has been deployed to WebSphere Process Server.

You must use the wsadmin tool to run the following commands.

To display the import bindings of a particular mediation module, use the wsadmin tool to complete the following steps.

In order to show the import bindings of a particular mediation module import, you need to know the mediation module name and the import name.

Note: Mediation modules are types of Service Component Architecture (SCA) modules.

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.

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 import binding for a particular import. `$AdminTask showSCAImportBinding {-moduleName moduleName -import importName}`

   Note: In addition to specifying the *moduleName* and *importName*, you have the option of specifying the *applicationName*. Providing an *applicationName* improves performance.

Displays the import binding for a particular SCA module import.

```
$AdminTask showSCAImportBinding {-moduleName myModule -applicationName
myApplication -import myImport}
```

Changing an SCA import binding:

You can use a command to change the SCA import bindings of mediation modules that have been deployed to WebSphere Process Server.

You must use the wsadmin tool to run the following commands.

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 invoke 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.

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`*

   Note: You also have the option of specifying the *applicationName* and *targetApplicationName*. Providing an *applicationName* and a *targetApplicationName* improves performance.

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:

You can use a command to change the Web service import bindings of mediation modules that have been deployed to WebSphere Process Server.

You must use the wsadmin tool to run the following commands.

You might change an import Web service binding if you wanted a particular mediation module to invoke 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.

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 import binding, of type WS. `$AdminTask modifySCAImportWSBinding -moduleName moduleName -import importName -endpoint targetEndpointName`

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:

You can use a command to list the exports of any mediation module that has been deployed to WebSphere Process Server.

You must use the wsadmin tool to run the following commands.

To list the exports of a mediation module, use the wsadmin tool to complete the following steps.

In order to list the exports of a particular mediation module, you need to know the mediation module name.

Note: Mediation modules are types of Service Component Architecture (SCA) modules.

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.

1. List the deployed SCA modules. `$AdminTask listSCAModules`
2. List the exports of a particular SCA module. `$AdminTask listSCAExports {-moduleName moduleName}`

   Note: In addition to specifying the *moduleName*, you have the option of specifying the *applicationName*. Providing an *applicationName* improves performance.

Lists the exports for a particular SCA module.

```
$AdminTask listSCAExports {-moduleName myModule -applicationName
myApplication}
```

Displaying details of an export:

You can use a command to display export details of a mediation module that has been deployed to WebSphere Process Server.

You must use the wsadmin tool to run the following commands.

To display the export details of mediation module, use the wsadmin tool to complete the following steps.

In order to show the details of a particular mediation module export, you need to know the mediation module name and the export name.

Note: Mediation modules are types of Service Component Architecture (SCA) modules.

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.

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`

   Note: In addition to specifying the *moduleName* and *exportName*, you have the option of specifying the *applicationName*. Providing an *applicationName* improves performance.

Displays the export details for a particular SCA module export.

```
$AdminTask showSCAExport {-moduleName myModule -applicationName
myApplication -export myExport}
```

Displaying an export binding:

You can use a command to display the export bindings of a mediation module that has been deployed to WebSphere Process Server.

You must use the wsadmin tool to run the following commands.

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.

Note: Mediation modules are types of Service Component Architecture (SCA) modules.

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.

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 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.

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}
```

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 15. 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

Table 15. Buttons for administering enterprise applications (continued)

Button	Resulting action
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 for z/OS documentation.

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.

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.

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 relationship types

Perform this task to view information related to the relationship type, including the relationship name, display name, and static and identity attributes.

Security role required: To perform this task, you must be logged in as a monitor, an operator, a configurator, or an administrator.

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 relationship type, including the relationship name, display name, associated roles with their attributes, property values, and static and identity attributes.

Security role required: To perform this task, you must be logged in as a monitor, an operator, a configurator, or an administrator.

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. In the **Select** column, select 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 relationship types, 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 role, including the relationship name, role name, display name, property values, keys, role object type, and managed attribute setting.

Security role required: To perform this task, you must be logged in as a monitor, an operator, a configurator, or an administrator.

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.

To return to the Relationship detail page, click **Relationship Detail** from the path at the top of the page or click **Back**.

Querying relationship instances

Use this task to perform relationship-based instance queries.

Security role required: To perform this task, you must be logged in as a monitor, an operator, a configurator, or an administrator.

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.

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. In the **Select** column, select 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 one field is left blank, it will return only the single instance. The query will return 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 relationship types.

Viewing relationship instance results

Perform this task to view a list of instances that match the relationship instance query. The results display in table view and include the relationship instance ID and the property values associated with the instance.

Security role required: To perform this task, you must be logged in as a monitor, an operator, a configurator, or an administrator.

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. In the **Select** column, select 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 relationship instances" on page 243
6. Click **OK** to open the Relationship instance results page.

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.

- 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 cache size hint 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.

Security role required: To perform this task, you must be logged in as a monitor, an operator, a configurator, or an administrator.

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. In the **Select** column, select 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**.

   The Relationship instance results page will open.
6. You can view the relationship instance details in two ways:
   - Click the relationship instance ID.
   - In the **Select** column, select the relationship instance ID; and click **Details**.

   To return to the list of relationship instance results, 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.

Security role required: To perform this task, you must be logged in as an operator or an administrator.

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. In the **Select** column, select 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**.

   The Relationship instance results page will open.
6. Display the relationship instance details in one of two ways:

- Click the relationship instance ID.
- In the **Select** column, select 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 instance results page.

Creating new relationship instances

Perform this task to create a new relationship instance.

Security role required: To perform this task, you must be logged in as an operator or an administrator.

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 Relationship types page will open.
4. In the **Select** column, select a relationship name; and click **Create**.

   The New relationship instance page will open.
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.

Security role required: To perform this task, you must be logged in as an operator or an administrator.

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. In the **Select** column, select 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**.

   The Relationship instance results page will open.
6. In the **Select** column, select the ID of the relationship instance you want to delete.
7. Click **Delete**.

   The relationship instance will be deleted immediately from the system.

Rolling back relationship instance data

Perform this task to rollback the instance data for a relationship to a specified date and time.

Security role required: To perform this task, you must be logged in as an operator or an administrator.

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. In the **Select** column, select the relationship name; and click **Rollback**.
5. Enter the time period for the rollback in the **From date** and **To date** fields.
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.

Security role required: To perform this task, you must be logged in as a monitor, an operator, a configurator, or an administrator

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. In the **Select** column, select 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**.

   The Relationship instance results page will open.
6. Display the relationship instance details in one of two ways:
   - Click the relationship instance ID.
   - In the **Select** column, select 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.

Security role required: To perform this task, you must be logged in as an operator or an administrator.

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. In the **Select** column, select 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**.

   The Relationship instance results page will open.
6. Display the relationship instance details in one of two ways:
   - Click the relationship instance ID.
   - In the **Select** column, select 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.

Security role required: To perform this task, you must be logged in as an operator or an administrator.

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. In the **Select** column, select 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**.

   The Relationship instance results page will open.
6. Display the relationship instance details in one of two ways:
   - Click the relationship instance ID.
   - In the **Select** column, select 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.

   The New role instance page will open.
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.

Security role required: To perform this task, you must be logged in as an operator or an administrator.

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. In the **Select** column, select 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**.

   The Relationship instance results page will open.
6. Display the relationship instance details in one of two ways:
   - Click the relationship instance ID.
   - In the **Select** column, select the relationship instance ID; and click **Details**.
7. Locate the role from which you want to delete the role instance.
8. In the **Select** column, click the role instance you want to delete; and click **Delete** below the role table.

   The role instance will be deleted locally.

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.

Changing the values of a relationship instance

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.

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. In the **Select** column, select 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 displays the Relationship instance results 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 create a new role instance for this relationship instance.

   The New role instance page will open.
10. Enter `801` in the **Value** field under **Key attributes**, and click **OK**.

    The new role instance will be saved, and you should see a new role instance in the table.

You now have the correct customer ID value in the relationship instance for the GL application.

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" on page 252
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 253
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 251
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 251
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>
```

Using event sequencing with Export bindings

Event sequencing is supported with EIS Export bindings, JMS Export bindings, native 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 concepts

"Export bindings and event sequencing" on page 107
You can stipulate that your Exports process and deliver messages in the order in which they are received. For event sequencing to be possible you must configure your Export bindings appropriately.

Related tasks

"Enabling event sequencing for an EIS Export binding" on page 107
Event sequencing is supported for EIS Export bindings. The activation specification of the binding must be appropriately configured.

"Enabling event sequencing for a JMS Export binding" on page 108
JMS Export bindings support event sequencing. In order for events to be handled in the order in which they are received you need to configure properties of the binding.

"Enabling event sequencing for a WebSphere MQ JMS Export binding" on page 108
WebSphere MQ JMS Export bindings support event sequencing. In order for events to be handled in the order in which they are received you need to configure properties of the binding.

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 16. 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:

- Using 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.

- Using 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 may need to specify event sequencing on downstream components, making sure you use the same event sequencing key for all methods.
- Event sequencing for synchronous invocations to components with asynchronous invocations is not supported for this release.

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 running it on the deployment manager can produce unpredictable results.

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 17. 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.

Managing WebSphere Process Server failed events

TheWebSphere 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.

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**.

Related concepts

"Working with data in failed events" on page 266
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.

"Resubmitting failed events" on page 269
If you want to try to execute the 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.

Related tasks

"Finding failed events" on page 261
Failed events are stored in the failed event database and are retrieved through

the failed event manager's search functionality. You can search for all failed events on the all the servers within the cell, or for a specific subset of events.

"Deleting failed events" on page 271
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.

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 the *Securing applications and their environment* topic in the WebSphere Process Server Information Center.

Finding failed events

Failed events are stored in the failed event database and are retrieved through the failed event manager's search functionality. You can search for all failed events on the all the servers within the cell, or for a specific subset of events.

You must be logged in as administrator or operator to perform 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.

1. Ensure the administrative console is running.
2. Click **Integration Applications > Failed Event Manager** to enter the failed event manager.
3. From the **Failed events on this server** box, click **Get all failed events**.

The Search Results page opens, displaying a list of all the WebSphere Process Server failed events in the cell.

You can now view and modify data in a failed event, resubmit it, or delete it.

Related tasks

"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.

"Searching for failed events by source" on page 263
Use the Search page's **By Source** tab 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.

"Searching for failed events by destination"
Use the Search page's **By Destination** tab 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.

"Searching for failed events by date" on page 264
Use the Search page's **By Date** tab to find only those events that failed during a specific time period.

"Searching for failed events by business object type" on page 264
Use the Search page's **By Type** tab to find only those failed events that are associated with a specific business object.

"Searching for failed events by exception" on page 265
Use the Search page's **By Exception** tab to find only those failed events that are associated with a specific exception. You can specify part or all of the exception text.

"Performing an advanced search for failed events" on page 265
Use the Search page's **Advanced** tab to perform a more refined search for failed events by using a combination of the criteria found on the other search tabs.

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.

You must be logged in as administrator or operator to perform 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.

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.

The Search Results page opens, listing all failed events that originated during the specified session.

Searching for failed events by destination

Use the Search page's **By Destination** tab 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.

You must be logged in as administrator or operator to perform 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.

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:
   - The **Destination module** field—Use this field to specify the failed event's destination module.
   - The **Destination component** field—Use this field to specify the failed event's destination component.
   - The **Destination method** field—Use this field to specify the failed event's destination method.
4. Click **OK** to begin the search.

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 Search page's **By Source** tab 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.

You must be logged in as administrator or operator to perform 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.

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:

- The **Source module** field—Use this field to specify the module that the failed event originated from.
- The **Source component** field—Use this field to specify the component that the failed event originated from.

4. Click **OK** to begin the search.

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 Search page's **By Date** tab to find only those events that failed during a specific time period.

You must be logged in as administrator or operator to perform 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.

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 Search page's **By Type** tab to find only those failed events that are associated with a specific business object.

You must be logged in as administrator or operator to perform this task.

To search for failed events by business object type, use the following 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:
   - The **Select the business object type** menu—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.
   - The **Other business object type** field—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.

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 Search page's **By Exception** tab to find only those failed events that are associated with a specific exception. You can specify part or all of the exception text.

You must be logged in as administrator or operator to perform this task.

To search for failed events by exception type, use the following 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.

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 Search page's **Advanced** tab to perform a more refined search for failed events by using a combination of the criteria found on the other search tabs.

You must be logged in as administrator or operator to perform 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.

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:
   - The **Destination module** field**—**Use this field to specify the failed event's destination module.
   - The **Destination component** field—Use this field to specify the failed event's destination component.
   - The **Session ID** field – Use this field to specify the session in which the event executed. This field does not accept the asterisk (*) wildcard character.
   - The **Destination method** field**—**Use this field to specify the failed event's destination method.
   - The **Source module** field—Use this field to specify the module that the failed event originated from.
   - The **Source component** field—Use this field to specify the component that the failed event originated from.
   - The **From Date** field—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.
   - The **To Date** field—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.
   - The **Business object type** field—Use this field to specify the type of business object associated with the failed events.
   - The **Exception text** field—Use this field to specify the text associated with the exception you want to search against.
4. Click **OK** to begin the search.

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.

You must be logged as administrator or operator to perform 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.

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 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 paramter is a complex data type, clicking its name will expand the hierachy 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.

You must be logged in as administrator or operator to perform 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.

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 261
   Failed events are stored in the failed event database and are retrieved through the failed event manager's search functionality. 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.

You must be logged in as administrator or operator to perform 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.

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 execute the 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.

You must be logged in as administrator or operator to perform this task.

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**.

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 executes successfully. The failed event manager provides optional tracing for all failed events.

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 WebSphere Process Server Information Center.

You must be logged in as administrator or operator to perform this task.

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.

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.

You must be logged in as administrator or operator to perform 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.

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 261 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**.

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.

You must be logged in as administrator or operator to perform 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.

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 261 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**.

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.

You must be logged in as administrator or operator to perform this task.

To delete one or more failed events, use the following 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"
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"
I am having trouble deleting expired events	"Executing the Delete Expired Events function appears to suspend the failed event manager" on page 273
I am having trouble with failed events not being created	"Failed events are not being created" on page 273

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-dependendent, 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.

Executing 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 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.

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 272

Problem	Refer to the following
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 272
I am having trouble deleting expired events	"Executing the Delete Expired Events function appears to suspend the failed event manager" on page 273
I am having trouble with failed events not being created	"Failed events are not being created" on page 273

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.

Executing 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.

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.

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.

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 **Resources > JMS Providers > WebSphere MQ > WebSphere MQ queue destinations**.
4. Locate any messages that may indicate failed transactions. You will have to investigate system, SCA-specific message areas, 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 MQ error, there will be an MQException linked somewhere in the stack trace with an 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) applications.

d. Examine the 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 "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.

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: © (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © 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 and related trademarks: http://www.ibm.com/legal/copytrade.shtml

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/).

WebSphere Process Server for z/OS, Version 6.0.2

Printed in USA