WebSphere® Process Server for Multiplatforms

IBM

Version 6.2.0

Administering WebSphere Process Server

WebSphere® Process Server for Multiplatforms

Version 6.2.0

Administering WebSphere Process Server

Note

Before using this information, be sure to read the general information in the Notices section at the end of this document.

24 April 2009

This edition applies to version 6, release 2, modification 0 of WebSphere Process Server for Multiplatforms (product number 5724-L01) and to all subsequent releases and modifications until otherwise indicated in new editions.

To send us your comments about this document, send an e-mail message to 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.

PDF books and the information center

PDF books are provided as a convenience for printing and offline reading. For the latest information, see the online information center.

As a set, the PDF books contain the same content as the information center.

The PDF documentation is available within a quarter after a major release of the information center, such as Version 6.0 or Version 6.1.

The PDF documentation is updated less frequently than the information center, but more frequently than the Redbooks®. In general, PDF books are updated when enough changes are accumulated for the book.

Links to topics outside a PDF book go to the information center on the Web. Links to targets outside a PDF book are marked by icons that indicate whether the target is a PDF book or a Web page.

Table 1. Icons that prefix links to topics outside this book

Icon	Description

⇨	A link to a Web page, including a page in the information center.
	Links to the information center go through an indirection routing service, so that they continue to work even if target topic is moved to a new location.
	If you want to find a linked page in a local information center, you can search for the link title. Alternatively, you can search for the topic id. If the search results in several topics for different product variants, you can use the search result **Group by** controls to identify the topic instance that you want to view. For example:
	1. Copy the link URL; for example, right-click the link then select Copy link location. For example: `http://www14.software.ibm.com/webapp/wsbroker/redirect?version=wbpm620&product=wesb-dist&topic=tins_apply_service`
	2. Copy the topic id after `&topic=`. For example: `tins_apply_service`
	3. In the search field of your local information center, paste the topic id. If you have the documentation feature installed locally, the search result will list the topic. For example:
	`1 result(s) found for`
	`Group by: None  | Platform  | Version  | Product`
	`  Show Summary`
	`Installing fix packs and refresh packs with the Update Installer`
	4. Click the link in the search result to display the topic.

📄	A link to a PDF book.

Contents

Figures

Tables

Chapter 1. Overview of administering WebSphere Process Server

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

The administrative console

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

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

If you have installed a stand-alone profile, you have a single node in its own administrative domain, known as a cell. Use the administrative console to manage applications, buses, servers, and resources within that administrative domain.

Similarly, if you have installed and configured a network deployment cell, you have a deployment manager node and one or more managed nodes in the same cell. Use the administrative console to manage applications, set up managed nodes in the cell, and monitor and control those nodes and their resources.

In the administrative console, task filters provide a simplified user experience and, through the progressive disclosure of functions, access to the full underlying WebSphere Application Server administrative capabilities.

Administrative console areas

Use the administrative console to create and manage objects such as resources, applications, and servers. Additionally, use the administrative console to view product messages. This topic describes the main areas that display on the administrative console.

To view the administrative console, ensure that the server for the administrative console is running. If you have configured a stand-alone server, the console runs on that server. If you have configured a network deployment cell, the console runs on the deployment manager server.

Point a Web browser at the Web address for the administrative console, enter your user ID and, if security is enabled, a password on the Login page.

You can resize the width of the navigation tree and workspace simultaneously by dragging the border between them to the left or the right. The change in width does not persist between administrative console user sessions.

The console has the following main areas.

Taskbar

The taskbar offers options for logging out of the console, accessing product information, and accessing support.

Navigation tree

The navigation tree on the left side of the console offers links to console pages that you use to create and manage components in a cell.

Click a plus sign (**+**) beside a tree folder or item to expand the tree for the folder or item.

Click a minus sign (**-**) to collapse the tree for the folder or item.

Click an item in the tree view to display its console page. This also toggles the item between an expanded and collapsed tree.

Workspace

The workspace on the right side of the console contains pages that you use to create and manage configuration objects such as servers and resources.

Click links in the navigation tree to view the different types of configured objects.

Within the workspace, click configured objects to view their configurations, run-time status, and options. Click buttons to perform actions on selected objects.

Click **Welcome** in the navigation tree to display the workspace Home page, which contains links to information on using the product.

Business integration areas of the administrative console

The business integration resources used by WebSphere Process Server and WebSphere Enterprise Service Bus are grouped into several areas of the administrative console.

Use the navigation tree to locate business integration resources, as follows.

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

  This option is available only if you have installed WebSphere Application Server Network Deployment.
- **Servers > Application servers >** *server_name*: Provides access to the following:
  - Container settings for business processes and human tasks
  - Business Integration configuration (tabbed page of deployment target functions)
  - Business Space configuration
  - System REST Service Endpoints configuration
  - Service Component Architecture configuration
  - Common Event Infrastructure server and destination configuration

  - Business Process Choreographer configuration
  - Business rules configuration
  - Selectors
  - WebSphere Business Integration Adapter Service
  - Application Scheduler
- **Servers > clusters >** *cluster_name*: Provides access to the following:
  - Container settings for business processes and human tasks
  - Business Integration configuration (tabbed page of deployment target functions)
  - Business Space configuration
  - System REST Service Endpoints configuration
  - Service Component Architecture configuration
  - Common Event Infrastructure server and destination configuration
  - Business Process Choreographer configuration
  - Business rules configuration
- **Applications → SCA Modules**: Provides access to the following:
  - SCA modules and their associated service applications
  - SCA module imports, including interfaces ad bindings
  - SCA module exports, including interfaces and bindings
  - SCA module properties
- **Resources** : Provides access to the following:
  - WebSphere Business Integration Adapters
  - People directory provider
  - Remote Artifacts
- **Integration Applications**: Provides access to the following:
  - Failed event manager
  - Relationship manager
  - Common Base Event Browser
- **Service integration**: Provides access to the following:
  - WebSphere Service Registry and Repository (WSRR) definitions
  - Service Integration Bus Browser

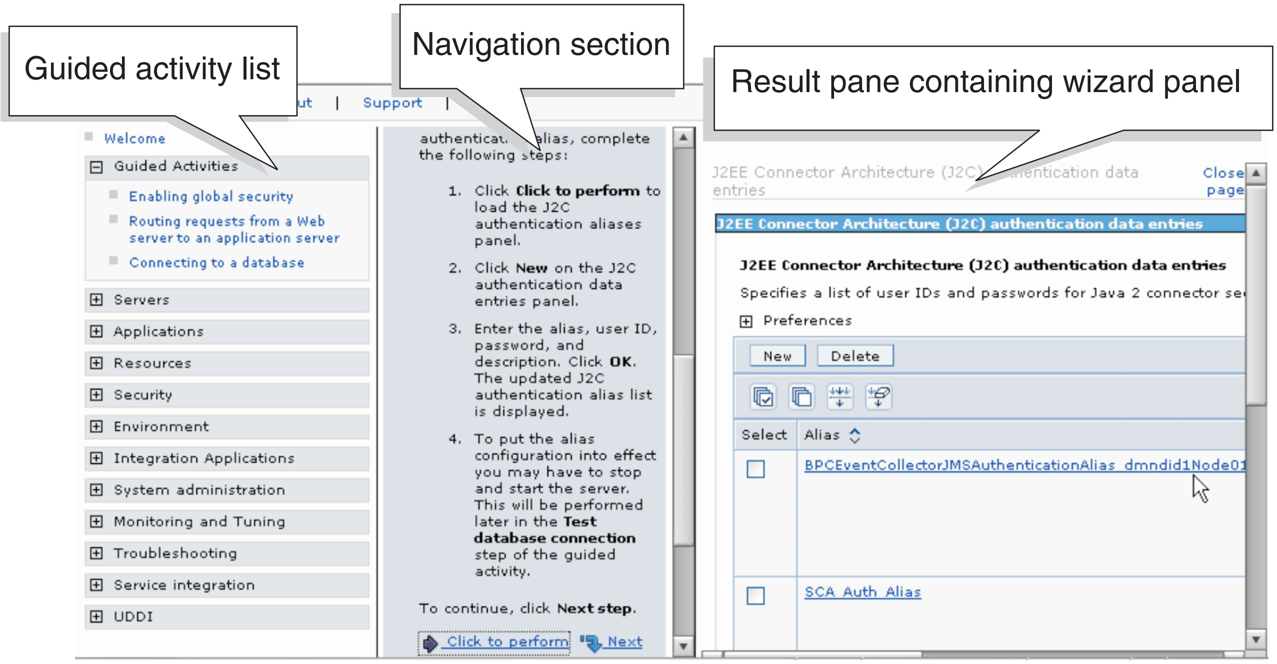
Administrative console guided activities

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

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

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

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



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

⊞ Preferences

Select	Alias
□	BPCEventCollectorJMSAuthenticationAlias_dmndid1Node01
□	SCA_Auth_Alias

Figure 1. A guided activity

Administrative console pages

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

- "Collection pages"
- "Detail pages" on page 5
- "Wizard pages" on page 5

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 5. In most cases, you need to select one or more objects in the collection table, then click a button. The action is applied to all selected objects.

Sorting toggle buttons

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

Detail pages

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

Configuration tabbed page

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

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

Runtime tabbed page

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

Changes to this tabbed page take effect immediately.

Local topology tabbed page

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

Buttons for performing actions

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

Wizard pages

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

Administrative console buttons

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

The following graphical buttons are located at the top of a table that displays server-related resources:

Table 2. Graphical buttons at the top of a console collection page

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 14.
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 an administrative console page. Not all buttons appear on all pages.

Table 3. Buttons at the bottom of a console page

Button	Resulting action
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 to administer all products and resources, refer to Administrative console buttons in the WebSphere Application Server Information Center.

Command-line tools, scripts, and programming interfaces

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

Command-line tools

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

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

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

Scripting (wsadmin)

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

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

Administrative programs

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

See Generated API and SPI documentation in this information center for details about the programming interfaces.

Business Process Choreographer Explorer overview

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

It also includes an optional reporting function, which was previously known as the Business Process Choreographer Observer.

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

A single Business Process Choreographer Explorer can only connect to one Business Process Choreographer configuration, though it does not have to connect to a local configuration. However, you can configure multiple instances of the Business Process Choreographer Explorer on the same server or cluster, and each instance can connect to different Business Process Choreographer configurations.

When you start Business Process Choreographer Explorer, the objects that you see in the user interface and the actions that you can take depend on the user group that you belong to and the authorization granted to that group. For example, if you are a business process administrator, you are responsible for the smooth operation of deployed business processes. You can view information about process and task templates, process instances, task instances, and their associated objects. You can also act on these objects; for example, you can start new process instances, create and start tasks, repair and restart failed activities, manage work items, and delete completed process instances and task instances. However, if you are a user, you can view and act on only those tasks that have been assigned to you.

Business rules manager

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

Business rules are designed and developed in WebSphere Integration Developer using if/then rule sets and decision tables to implement their operations. Business rules can also be created in WebSphere Business Modeler; however Modeler only supports the creation of business rule tasks, which become rule sets when exported out of Modeler. The rule sets and decision tables are set into templates. The templates control which aspects of a business rule you can modify and by exactly how much. They define the structure of if/then rules, condition cases, and actions for decision tables.

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

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

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

Configuration information

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

The directory in which a configuration file exists determines its scope, or how broadly or narrowly that data applies.

- Files in an individual server directory apply to only that server.
- Files in an application directory apply to only that application.
- Files in a cluster-level directory apply to only that cluster.
- Files in a node-level directory apply to every server on that node.
- Files in a cell directory apply to every server on every node within the entire cell.

Table 4. WebSphere Process Server configuration files

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

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

See Configuration file description for more information.

Chapter 2. Getting started with the administrative interfaces

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

Getting started with the administrative console

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

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

- **Start the server.**

  Before you can use the administrative console, you must start the stand-alone server or deployment manager. For instructions on starting a stand-alone server, see Starting and stopping stand-alone servers. For instructions on starting the deployment manager, see Starting and stopping the deployment manager.
- **Start the administrative console.**

  See "Starting and stopping the administrative console" for details.
- **Specify console preferences.**

  Preferences control how data is displayed in the administrative console, as well as how the workspace behaves. See "Setting administrative console preferences" on page 13.
- **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 Administrative console scope settings for details.
- **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 14.
- **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 Configuration document descriptions.
- **Save your work to the master repository.**

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

Starting and stopping the administrative console

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

Before you begin

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

About this task

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

Procedure

1. Start the administrative console:
   1. Enable cookies in the Web browser that you plan to use to access the administrative console.
   2. Optional: Enable JavaScript™. JavaScript enablement is recommended so that all the features of the administrative console are available to you.
   3. 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 system that contains the administrative server and *portNumber* is the administrative console port number. When the administrative console is on the local system, *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 system 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.
   4. Check the System.Out.log file of the server that runs the console application to verify that the console application has started successfully. A successful start produces the message `WSVR0221I: Application started: isclite`.

      If you are unable to start the console because the console port conflicts with an application that is already running on the system, change the port number in the following files:

      - *profile_root*/config/cells/*cell_name*/nodes/*node_name*/serverindex.xml
      - *profile_root*/config/cells/*cell_name*/virtualhosts.xml

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

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

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

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

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

   - Log out the other user with the same user ID. You can recover changes made during the 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.

About this task

You can set the following display preferences for collection pages:

- **Maximum rows**: Specifies the maximum number of rows that are displayed when the collection is large. If there are more rows than the specified maximum, they are displayed on subsequent pages. The default value is 20.
- **Retain filter criteria**: Specifies whether the last search criteria entered in the filter function is retained. If this is enabled, the console collection pages initially use the retained filter criteria to display the data in the table following the preferences. See "Setting administrative console filters" on page 14 for more information.
- **Max result set size**: Specifies the maximum number of resources that a search can return. The default value is 500.
- **Max column width**: Specifies the maximum number of characters viewable in a collection column. The default value is 18.

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

Procedure

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

   The page expands to display the preference fields.

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

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

What to do next

You can also set global administrative console preferences, such as whether the workspace is automatically refreshed and which scope to use by default. To access the Preferences page in the administrative console, click **System administration** → **Console settings** → **Preferences**. For more information about setting these preferences, see the WebSphere Application Server Information Center.

Setting administrative console filters

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

Procedure

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

   The filter dialog box opens above the top row of the table.
2. Use the **Filter** drop-down menu to select the column you want to include in the filter.
3. In the **Search terms** field, specify the filter criteria.

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

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

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

   ```
   *\(XA\)*
   ```
4. Click **Go**.

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

Using My tasks

Customize console navigation by creating and editing a task view.

About this task

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

Procedure

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

Results

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

Accessing product information and help from the administrative console

The administrative console provides access to product documentation as well as online help for each page and field. You can view the help in the console help browser or in the WebSphere Process Server Information Center.

About this task

Perform the following steps to access product information and administrative console help topics.

Procedure

1. Access the product information by doing the following tasks.
   a. Click **Welcome** on the administrative console navigation tree. In the workspace to the right of the navigation tree, the console displays information about the installed products.
   b. Click the appropriate links to access the product Information Center and the related technical information on developerWork
2. Access the product help in one of the following ways.

Option	Description
Access field-level help in the administrative console	• 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). **Note:** 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.

Option	Description
Access the stand-alone help browser	Click **Help** from the console task bar to view online help in a new Web browser. From here you have the following options: • 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.
View the online help in the WebSphere Process Server Information Center	• Use a browser to navigate directly to the WebSphere Process Server Information Center. Online help topics are in the **Reference** section. • Click the **Check for updates to this topic** link in any help file viewed with the help browser.
View command assistance	If command assistance is available, click **View administrative scripting command for last action** in the right-hand Help portal.

Related tasks

Accessing command assistance from the administrative console
Use command assistance to see wsadmin scripting commands that correspond to actions in the administrative console. Seeing the commands can help you develop the command-line tools needed to administer the server from the wsadmin utility.

Accessing command assistance from the administrative console

Use command assistance to see wsadmin scripting commands that correspond to actions in the administrative console. Seeing the commands can help you develop the command-line tools needed to administer the server from the wsadmin utility.

Before you begin

Before using command assistance, do the following:

- Start WebSphere Process Server and the administrative console.
- Determine whether you want to save command assistance data to a log file. When logging is enabled, a timestamp and the breadcrumb trail of the page that produced the command assistance data are provided with the wsadmin data in the commandAssistanceJythonCommands_username.log file in the logs directory for the process running the console.

  Click **System administration** → **Console preferences** → **Log command assistance commands** to save command assistance data to the log file.
- Determine whether you want to allow command assistance to emit Java Management Extensions (JMX) notifications. Enabling the notifications allows integration with product tools that can help you write automation scripts (for example, the WebSphere Application Server Toolkit Jython editor). The notification type is websphere.command.assistance.jython.*user_name*, where *user_name* specifies the current administrative console user.

Note: This option is recommended for non-production environments only.

Click **System administration** → **Console preferences** → **Enable command assistance notifications** to enable JMX notifications.

About this task

Using command assistance, you can view wsadmin scripting commands in the Jython language for the last action run in many pages in the administrative console.

If a command assistance link is listed in the help portlet, wsadmin commands exist for the last console action you completed, and command assistance is available for that action.

Examples of actions include a click on a button or a click on a link in the navigation bar, collection page, or detail page. Editing a form is not a user action and is not captured by command assistance.

The wsadmin scripting commands display in the Jython language in a secondary window. If you perform an administrative action after you launch the Administrative Scripting Commands window, the window automatically refreshes the command list to reflect the most recent console action.

When command assistance is unavailable in the help portlet: Some console actions do not have wsadmin commands directly associated with them. When the help portlet on the right side of the administrative console page does not have a command assistance link in it, no command assistance data is available for the last console action.

To use command assistance in the console, perform the following steps.

Procedure

1. Optional: Set console preferences to capture command assistance data in a log file, as follows:
   a. Click **System Administration** → **Console Preferences** to open the Preferences page.
   b. Select **Log command assistance commands**.
2. Optional: Set console preferences to allow command assistance to emit Java Management Extensions (JMX) notifications, as follows:
   a. Click **System Administration** → **Console Preferences** to open the Preferences page.
   b. Select **Enable command assistance notifications** to emit websphere.command.assistance.jython.*user_name* notifications.
3. Navigate to the console page you want to use with command assistance.
4. Click **View administrative scripting command for last action** from the Help portlet on the right side of the page. The Administrative Scripting Commands window opens and displays the Jython for the related wsadmin scripting command.
5. Optional: View the description of a specific wsadmin command by placing your cursor over the command to display hover help.

Results

You have viewed wsadmin scripting commands from the administrative console, optionally logged the commands to a file, and optionally allowed command assistance to emit JMX notifications.

What to do next

You can use the information provided by command assistance when creating wsadmin scripts to automate administrative tasks.

Related reference

Administrative console actions with command assistance
Command assistance is available for a subset of administrative console actions. When available, command assistance displays the wsadmin scripting command for the last console action you performed. You can then use this data to create wsadmin scripts that automate certain administrative tasks.

Related information

Administrative console actions with command assistance (WebSphere Application Server)

Using scripting (wsadmin)

Getting started with Business Process Choreographer Explorer

Depending on your user role, you can use Business Process Choreographer Explorer to manage business processes and human tasks, or to work with your assigned tasks. While business processes and tasks are running, WebSphere Process Server can emit events that contain information about state changes of process instances and their related activities. Using reporting, you can retrieve statistical information based on these events and create reports on processes and activities.

About this task

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

- If you are a business administrator, you can manage the life cycle of business processes, and you can repair business processes. For example, you can restart or force the completion of single activities, or compensate the business process as a whole. If compensations failed, you can retry, skip or stop the process instances. In addition, you can add and update custom properties for business processes and activities.
- If you are a human task administrator, you can manage the life cycle of human tasks, and manage work assignments. For example, you can assign responsibility to users, or manage absence handling and substitution for users. You can also change the priority and business category for human tasks, and add or update custom properties.
- With the reporting function of Business Process Choreographer Explorer you can monitor the history of process instances, activity instances, or inline human tasks. If your Business Process Choreographer Explorer configuration includes the reporting function you can define your own reports, or use a drill-down approach to get more detailed information on specific process instances, activity instances, or inline human tasks. In addition, you can export the reported results for further external processing.

- If you are a business user, you can use Business Process Choreographer Explorer to work with your assigned tasks. For example, you can initiate business processes, services, and human tasks, and you can work on, edit, save, complete, or release human tasks. In addition, you can flag your absence and define substitutes.

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

Related tasks

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

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

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

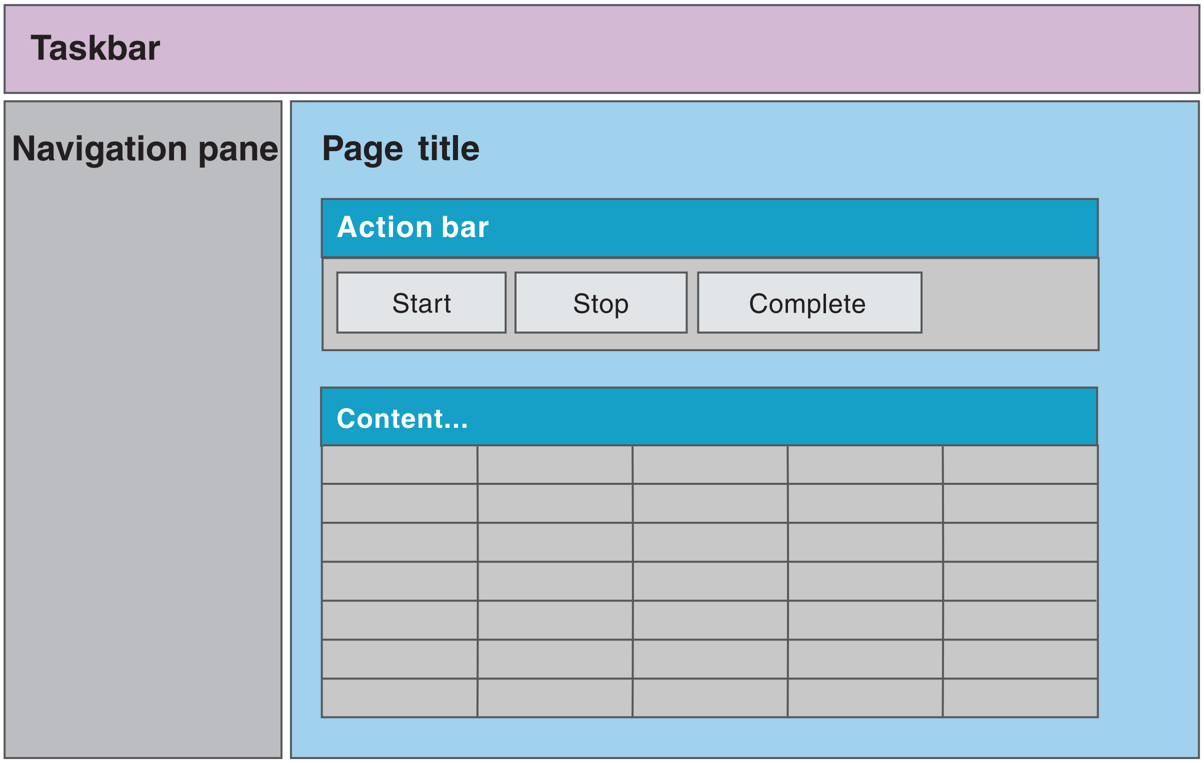
Working on your tasks
To work on a task, you must claim the task and then perform the actions that are needed to complete it.

Reporting on business processes and activities
During the processing of business processes and activities, events can be generated when the process, activity, or task changes state. These events are stored and made available for creating reports using Business Process Choreographer Explorer, for example, to analyze process performance issues, or to evaluate the reliability of a service that is called from an activity.

Business Process Choreographer Explorer user interface

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

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



The user interface has the following main areas.

Taskbar

For all users, the taskbar offers options for logging out of Business Process Choreographer Explorer and for accessing online help. In addition, the options **My Substitutes** and **Define Substitutes** are available for specifying absence settings. These options are available when substitution is enabled for the Human Task Manager in Business Process Choreographer and the Virtual Member Manager service is configured for WebSphere Application Server security.

My Substitutes
Select this option to specify substitutes for a user's tasks.

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

If you have system administrator rights, the taskbar also includes the following options:

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

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

Navigation pane

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

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

If the Reports tab is selected, the navigation pane contains links that you use to select the kind of report that you want to create, for example, you can view the data for an activity instance in a chart. Use the predefined lists and charts to get state and event information for runtime entities, for example, to get process and activity snapshot charts. The Reports tab is visible only if reporting is configured. The reporting function can be configured when you configure Business Process Choreographer Explorer, however it can also be configured later.

Page title

If the Views tab is selected, the workspace contains pages that you use to view and administer business process and human task related objects. You access these pages by clicking the links in the navigation pane, by clicking an action in the action bar, or by clicking links within the workspace pages. For information about a page click the **Help** icon i on the respective page.

If the Reports tab is selected, the workspace contains pages that you use to view predefined lists and charts, specify report definitions, and to view reports. You access these pages by clicking the links in the navigation pane, by clicking an action in the action bar, or by clicking links within the workspace pages. For information about a page click the **Help** icon i on the respective page.

Related reference

Configuring the Business Process Choreographer Explorer reporting function and event collector
Using the Business Process Choreographer Explorer reporting function is optional, however, before you can use it, you must setup the database and install the applications.

Business Process Choreographer Explorer Views tab

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

Available actions

The following actions are available in the navigation pane:

- Collapse and expand a group.

  Click the arrow beside an item in the navigation pane to expand or collapse the item.
- Navigate to a view.

  Click the view name to navigate to that view.

- Define a new search.

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

Additional actions are available from the pop-up menu depending on the view type. The **Show pop-up menu** icon ( ) indicates that a pop-up menu is available.

- To delete the view, click the **Delete** icon (✖).
- To modify the view, click the **Edit** icon (✎).
- To create a copy of the view and modify the copy, click the **Copy** icon ( ).
- To move the view up or down in the list, click the **Up** icon (⇧) or the **Down** icon (⇩).

View types

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

Predefined views in the default navigation pane

These groups of views are available in the navigation pane, and do not initially have a pop-up menu. When the navigation pane is changed using **Customize**, these predefined then have the **Predefined view** icon in front of them, which makes it possible to move them up or down.

Customized views and predefined views that were added to the navigation pane by the system administrator

Business users can click the view name and navigate to the view. For system administrators, pop-up menus are available.

- The predefined views are indicated by the **Predefined view** icon: . A system administrator can use the pop-up menu to change the position of these views in the navigation pane.
- The customized views are indicated by the **Custom view** icon: . A system administrator can delete, edit, copy, and move these views.

Personalized views

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

Predefined views in the navigation pane

The default navigation pane contains the following groups of views. The views that are shown in the navigation pane in your Business Process Choreographer Explorer might differ depending on whether your system administrator has added views to, or removed views from the navigation pane. All views show items, independent of additional filters, to which you are authorized. For example, you see only the terminated processes you are allowed to see. If no view is defined for a group of views, the group is not displayed.

Process Templates

The process templates group contains the following view:

Process Templates

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

Process Instances

The process instances group contains the following views:

Started By Me

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

Administered By Me

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

Critical Processes

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

Terminated Processes

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

Failed Compensations

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

Activity Instances

The activity instances group contains the following view:

Failed Activities

This view shows the activities that are in the failed state. Activities that are in the stopped state are not contained.

Task Templates

The task templates group contains the following view:

My Task Templates

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

Task Instances

The task instances group contains the following views:

My To-dos

This view shows a list of the task instances that you are authorized to work with. From this view, you can work on a task instance, release a task instance that you claimed, or transfer a task instance to another user. You can also change the priority of a task and change its business category.

All Tasks

This view shows all of the tasks for which you are the owner, potential owner, or editor. From this view, you can work on a task instance, release a task instance that you claimed, or transfer a task instance to another user. You can also change the priority of a task and change its business category.

Initiated By Me

This view shows the task instances that you initiated. From this view, you can work on a task instance, release a task instance that you claimed, or transfer a task instance to another user. You can also change the priority of a task and change its business category.

Administered By Me

This view shows the task instances that you are authorized to administer. From this view, you can act on the task instance, for example, to suspend and resume a process, to create work items for the task instance, or to display a list of the current work items for the task instance. You can also change the priority of a task and change its business category.

My Escalations

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

Business Process Choreographer Explorer Reports tab

Use the Reports tab of Business Process Choreographer Explorer to manage reports for specific processes and activities that were processed by Business Process Choreographer. You can select the kind of report that you want to create, such as process or activity reports. You can also store your own report definitions and add these to the navigation pane. Use the predefined lists and charts for a drill-down approach to get state and event information for runtime entities. For example, lists, process and activity snapshot charts, and process and activity instances by period charts are available. The Reports tab is visible only if reporting is configured. The reporting function can be configured when you configure Business Process Choreographer Explorer, however it can also be configured later.

Available actions

The following actions are available in the navigation pane:

- Collapse and expand a group.

  Click the arrow beside an item in the navigation pane to expand or collapse the item.
- Navigate to a predefined list or chart.

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

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

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

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

  – To delete the report definition, click the **Delete** icon (✖).

  – To edit the report definition, click the **Edit** icon (✏).

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

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

Predefined lists and charts in the navigation pane

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

Lists This group contains the following lists:

Processes

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

Activities

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

Users

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

Charts This group contains the following charts:

Process snapshot

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

Processes by period

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

Activity snapshot

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

Activities by period

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

Process and activity reports

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

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

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

Related tasks

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

Related reference

Configuring the Business Process Choreographer Explorer reporting function and event collector
Using the Business Process Choreographer Explorer reporting function is optional, however, before you can use it, you must setup the database and install the applications.

Starting Business Process Choreographer Explorer

Business Process Choreographer Explorer is a Web application that can be installed as part of the configuration of the business process container. Before you can start using Business Process Choreographer Explorer from a Web browser, you must have installed the business process container, human task container, and the Business Process Choreographer Explorer application, and the application must be running. The event collector application must be installed and running in order to use the reporting function.

About this task

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

Procedure

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

   The URL takes the following form. The value of the URL depends on how the virtual host and context root were configured for your installation. In addition, you can extend the URL to go directly to the details of a process, task, or escalation.

   ```
   http://app_server_host:port_no/context_root?oid_type=oid
   ```

   For example:

   ```
   http://hostname:9080/bpc?piid=PI:90030109.7232ed16.d33c67f6.beb30076
   ```

   Where:

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

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

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

oid_type
Optional. The type of object that you want display. This parameter can take one of the following values:

aiid	Activity instance ID
piid	Process instance ID
ptid	Process template ID
tkiid	Task instance ID
tktid	Task template ID
esiid	Escalation instance ID

oid Optional. The value of the object ID.

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

Results

If you specified an object ID, the details page for the object is displayed. If you did not specify an object ID, the initial Business Process Choreographer Explorer page is displayed. By default, the initial page is the My To-dos view.

Customizing Business Process Choreographer Explorer

Business Process Choreographer Explorer provides a user interface for administrators to manage business processes and human tasks, and for business users to work with their assigned tasks. Because this is a generic interface, you might want to customize the interface for a specific Business Process Choreographer Explorer instance to address the business needs of user groups that are assigned to this instance. Furthermore, during configuration (or later) users can choose to add the reporting function to create reports on processes and activities and to retrieve statistical information on events.

About this task

You can customize the user interface in various ways.

Customizing the Business Process Choreographer Explorer interface for different user groups

The navigation pane in the default Business Process Choreographer Explorer user interface contains a set of links to predefined views. The My To-dos view is the default view of the Views tab that is shown after you log in. If you have one of the system administrator roles of Business Process Choreographer, using **Customize** in the taskbar, you can customize the links that are shown in the navigation pane and the view that your users see when they log in. Additionally, using **Define Views**, you can define views that your users see in the navigation pane with the information, filter and sort criteria, and actions that you want in the views.

Before you begin

To customize the interface, you must be a system administrator of Business Process Choreographer.

About this task

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

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

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

Procedure

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

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

      After saving your changes, the predefined views appear with icons in front of them in the navigation pane, which allows you to move them up and down in the list.

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

   You can specify the information that is shown in the views for this Business Process Choreographer Explorer instance.
   a. Click **Define Views** in the taskbar.
   b. In the Search And Define Customized Views page, select the type of view that you want to customize, for example, process templates.
   c. In the Search For ... And Define Customized Views page, where ... is the type of view, for example Process Templates, specify the search criteria.

      Use the Process Criteria tab, the Task Criteria tab, and the Property Filters tab to limit the search results, for example, to a specific process template.

When defining instance views, you can also use the User Roles tab to limit the search results to users, groups, or roles.

d. Use the View Properties tab to select the list columns and list properties, such as ordering properties and the results threshold, to include in the view.

   In addition, in View Settings, you can specify the actions to add to the action bar in the view. To select the actions to be included in the view or search that you are about to run:

   - In Available Actions, select an action or actions, and click **Add**.
   - To remove an action, select the action in Actions for View, and click **Remove**.
   - The sequence of the actions in the action bar can be specified by moving the actions up and down in Actions for View.

   If this is a task, process, or activity instance view, click **View Settings** to specify the items that are included in the view for system administrators and system monitors.

   - For system administrators and system monitors, you can limit the search result to their own instances:
     - To show all items that match the search criteria in the view, select **All Instances**. All of the items are shown regardless of whether the system administrator has work items for these items.
     - To show only the items that the logged-on user has work items for, select **Personal Instances**.

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

   Use the Summary tab to check the settings that are currently set for the view.

The new view appears in your navigation pane. Users see the new view when they next log on to Business Process Choreographer Explorer. The views can be moved up or down in the navigation pane.

Defining views for process templates for business state machines:

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

Before you begin

To create customized views, you must have one of the system administrator roles.

About this task

Complete the following steps in Business Process Choreographer Explorer.

Procedure

1. Click **Define Views** in the taskbar.
2. In the Search and Define Customized Views page, select **Search For Process Templates And Define Customized Views**.
3. Click **Property Filters** → **Custom Property Filters**.
   a. Add a custom property with the following settings:
      - In the **Property Name** field, type `generatedBy`.
      - In the **Property Value** field, type `BusinessStateMachine`.

   b. Click **Add**.
   c. Add other custom properties as needed.
4. Click **View Properties** → **List Columns**.
   a. In the List Columns for Custom Properties, add a custom property with the following settings:
      - In the **Property Name** field, type `generatedBy`.
      - In the **Display Name** field, type a display name for the column, and click **Add**.
   b. Add other columns to or remove columns from the list of selected columns.
5. Type a display name for the query in the **View Name** field, and click **Save**.

Results

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

Defining views for process instances for business state machines:

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

Before you begin

To create customized views, you must have one of the system administrator roles.

About this task

Complete the following steps in Business Process Choreographer Explorer.

Procedure

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

Namespace field. These values are derived from the definition of the business state machine. Also provide a display name for the column in the **Display Name** field.

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

Variable Name
If the correlation set is initiated by incoming parameters, the variable name has the following format:

```
operation_name_Input_operation_parameter_name
```

where *operation_name* is the name of the operation for the transition out of the initial state.

If the correlation set is initiated by outgoing parameters, the variable name has the following format:

```
operation_name_Output_operation_parameter_name
```

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

b. Add other custom properties or query properties, or add columns to or remove columns from the list of selected columns.

5. Type a name for the query in the **View Name** field, and click **Save**.

Results

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

Personalizing the Business Process Choreographer Explorer interface

The navigation pane in the default Business Process Choreographer Explorer user interface contains a set of links to predefined views and views that are defined by your system administrator. Independent of your roles, you can add your own views to your navigation pane. For example, you can add a new view to monitor the progress of a specific task or process. You can specify the information shown, the filter and sort criteria, and also the actions provided in the view.

About this task

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

Procedure

1. In the section of the Views tab navigation pane, for example, Process Templates, where you want to define the new view, click the **New search** icon ( ).
2. In the Search For ... and Define Personalized Views page for the view, for example, Search For Process Templates And Define Personalized Views, specify the search criteria.

Use the Process Criteria tab, the Task Criteria tab, and the Property Filters tab to limit the search results, for example, to a specific process template. When defining instance views, you can also use the User Roles tab to limit the search results to users, groups, or roles.

3. Use the View Properties tab to select the list columns and list properties, such as ordering properties and the results threshold, to include in the view.

   In addition, in View Settings, you can specify the actions to add to the action bar in the view. To select the actions to be included in the view or search that you are about to run:

   - In Available Actions, select an action or actions, and click **Add**.
   - To remove an action, select the action in Actions for View, and click **Remove**.
   - The sequence of the actions in the action bar can be specified by moving the actions up and down in Actions for View.

   If this is a task, process, or activity instance view, click **View Settings** to specify the items that are included in the view for system administrators and system monitors. If you are a system administrator and or a system monitor, you can limit the search result to your own instances.

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

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

   Use the Summary tab to check the settings that are currently set for the view.

Results

The new view appears in your navigation pane.

Changing the appearance of the default Web application

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

Before you begin

Style sheet modification requires profound knowledge about cascading style sheets.

About this task

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

Procedure

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

Related concepts

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

Styles used in the Business Process Choreographer Explorer interface:

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

The `style.css` file contains styles for the following elements of the default user interface:

- "Banner"
- "Footer"
- "Menu bar" on page 34
- "Login page" on page 34
- "Navigator" on page 34
- "Content panels" on page 34
- "Command bar" on page 35
- "Lists" on page 35
- "Details panel" on page 35
- "Message data" on page 35
- "Tabbed panes" on page 35
- "Search pages" on page 36
- "Error details" on page 36

This file is in the following directory:

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

Banner

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

Footer

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

Menu bar

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

Login page

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

Navigator

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

Content panels

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

Command bar

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

Lists

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

Details panel

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

Message data

Style name	Description
.messageData	The division container around a message.
.messageDataButton	Button style for Add and Remove buttons in the message form.
.messageDataOutput	For rendering read-only text.
.messageDataValidInput	For message values that are valid.
.messageDataInvalidInput	For message values that are not valid.

Tabbed panes

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

Style name	Description
.tabHeaderSimple	The tab header of a nested tabbed pane.
tabHeaderProcess	The tab header of a nested tabbed pane for process filters.
.tabHeaderTask	The tab header of a nested tabbed pane for task filters.
.tabPaneSimple	The division container that encloses a nested tabbed pane.

Search pages

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

Error details

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

Getting started with Business Process Choreographer Observer

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

About this task

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

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

Related concepts

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

Business Process Choreographer Explorer Reports tab
Use the Reports tab of Business Process Choreographer Explorer to manage reports for specific processes and activities that were processed by Business Process Choreographer. You can select the kind of report that you want to create, such as process or activity reports. You can also store your own report definitions and add these to the navigation pane. Use the predefined lists and charts for a drill-down approach to get state and event information for runtime entities. For example, lists, process and activity snapshot charts, and process and activity instances by period charts are available. The Reports tab is visible only if reporting is configured. The reporting function can be configured when you configure Business Process Choreographer Explorer, however it can also be configured later.

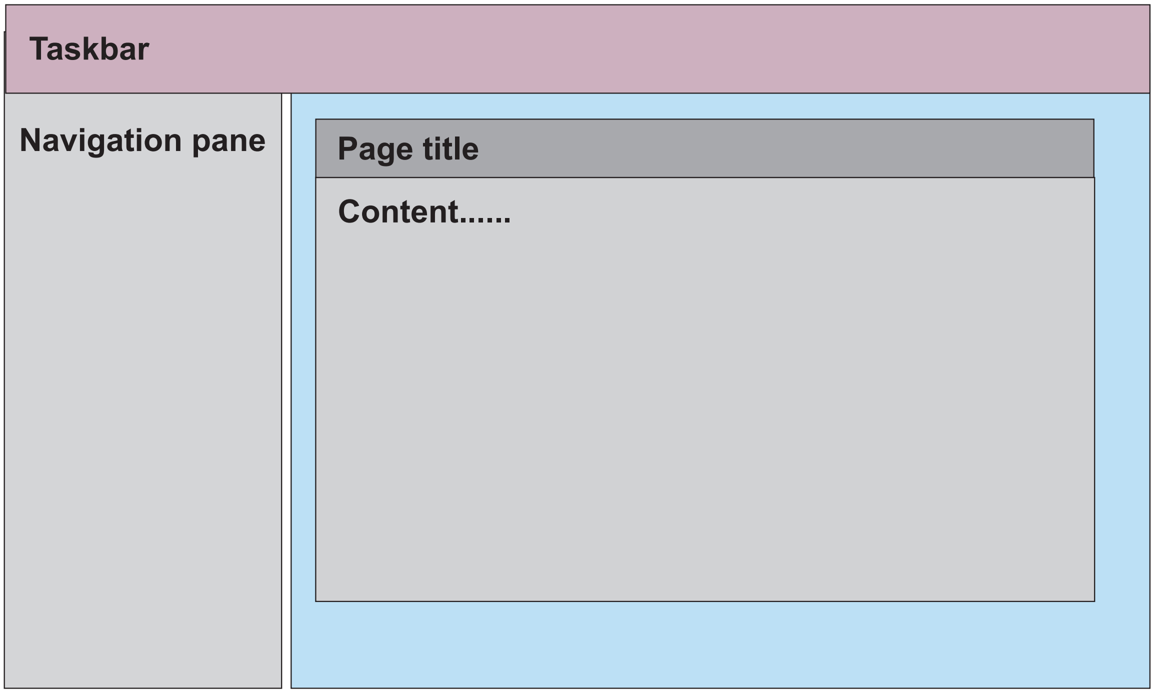
Related tasks

Reporting on business processes and activities
During the processing of business processes and activities, events can be generated when the process, activity, or task changes state. These events are stored and made available for creating reports using Business Process Choreographer Explorer, for example, to analyze process performance issues, or to evaluate the reliability of a service that is called from an activity.

Business Process Choreographer Observer user interface

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

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



The user interface has the following main areas.

Taskbar

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

Navigation pane

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

Workspace

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

Related tasks

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

Business Process Choreographer Explorer Reports tab

Use the Reports tab of Business Process Choreographer Explorer to manage reports for specific processes and activities that were processed by Business Process Choreographer. You can select the kind of report that you want to create, such as process or activity reports. You can also store your own report definitions and add these to the navigation pane. Use the predefined lists and charts for a drill-down approach to get state and event information for runtime entities. For example, lists, process and activity snapshot charts, and process and activity instances by period charts are available. The Reports tab is visible only if reporting is configured. The reporting function can be configured when you configure Business Process Choreographer Explorer, however it can also be configured later.

Available actions

The following actions are available in the navigation pane:

- Collapse and expand a group.

  Click the arrow beside an item in the navigation pane to expand or collapse the item.
- Navigate to a predefined list or chart.

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

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

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

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

  - To delete the report definition, click the **Delete** icon ( ).
  - To edit the report definition, click the **Edit** icon ( ).
  - To copy the report definition, click the **Copy** icon ( ).
  - To export the report result, click the **Export** icon ( ).
  - To run a report asynchronously, click the **Asynchronous Report** icon ( ).
    - After the asynchronous report completes successfully, the **Asynchronous Report Completed** icon ( ) is displayed in the navigation pane. Click the name of the report to view your results.
    - If the asynchronous report does not complete successfully, the **Asynchronous Report Failed** icon ( ) is displayed.

Predefined lists and charts in the navigation pane

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

Lists This group contains the following lists:

Processes

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

Activities

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

Users

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

Charts This group contains the following charts:

Process snapshot

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

Processes by period

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

Activity snapshot

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

Activities by period

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

Process and activity reports

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

Process reports

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

Activity reports

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

Related tasks

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

Related reference

Configuring the Business Process Choreographer Explorer reporting function and event collector
Using the Business Process Choreographer Explorer reporting function is optional, however, before you can use it, you must setup the database and install the applications.

Chapter 3. Administering servers

Use the administrative interfaces to create, start, and stop servers. Servers extend the ability of application servers, allowing them to handle Service Component Architecture (SCA) modules. Other server processes, such as deployment managers and node agents, can be used to manage servers.

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

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

Creating a new server

During the installation process, the product creates a default server. Most installations require several servers to handle the application serving needs of the production environment. You can use the command-line tool or the administrative console to create additional process servers.

Before you begin

Determine if you want to include the new server in a cluster. If this server is going to be part of a cluster, you must create the server with the Create a new cluster wizard instead of the Create a new application server wizard.

About this task

Important: This task creates a managed server. If you want a stand-alone server, do not follow these steps. Instead, create a stand-alone server profile.

To create a new managed server, perform the following steps.

Procedure

1. Follow the instructions in Creating application servers, selecting the **defaultProcessServer** template or a suitable user-defined template from the Select a server template page.
2. **Optional:** If the server will run applications that contain business processes or human tasks, configure Business Process Choreographer.

What to do next

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

Managing the administrative architecture

After you install and configure a deployment environment, use the administrative tools to monitor and control the resources in the deployment environment, including deployment managers, node agents, and clusters.

Starting deployment managers

The deployment manager is a server process. You must start the deployment manager before you can use its administrative console to manage the cell.

About this task

Perform the following steps to start and stop a deployment manager.

Procedure

1. Start the deployment manager with one of the following actions:
   - Windows From the **Start** menu, select **IBM WebSphere** → **Process Server 6.2** → **Profiles** → *profile_name* → **Start the deployment manager**.
   - In the First steps console, click **Start the deployment manager**.
   - Use the startManager command.
2. Verify that the deployment manager started successfully by checking the `install_root/profiles/profile_name/logs/server_name/startServer.log` log file for the message `Server server_name open for e-business; process id is nnnn`.

What to do next

You can now start the administrative console and manage the cell.

Stopping a deployment manager

Stop the deployment manager server process when performing certain maintenance activities such as migrating to a new version of the product or uninstalling the product. You can stop the deployment manager at any time without affecting the operation of the servers in its domain.

Before you begin

The deployment manager must be running.

About this task

To stop a deployment manager, perform the following steps.

Procedure

1. Stop the deployment manager with one of the following actions:
   - Windows From the **Start** menu, click **IBM WebSphere** → **Process Server 6.2** → **Profiles** → *profile_name* → **Stop the deployment manager**.
   - In the First steps console, click **Stop the deployment manager**.

- From the administrative console, click **System administration** → **Deployment manager** → **Stop** → **OK**. The administrative console closes before the server stops running.
- Use the stopManager command.

2. Verify that the deployment manager stopped successfully by checking the `*install_root*/profiles/*profile_name*/logs/*server_name*/startServer.log` log file for the message `Server *server_name* stop completed`.

Starting node agents

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

Before you begin

Before you can start and stop a node, you must federate the node into a cell.

About this task

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

To start a node agent, perform the following steps.

Procedure

1. Verify that the node agent is not currently running:
   a. Start the administrative console on the deployment manager.
   b. Click **System administration** → **Node agents** and verify that the node agent is stopped.
2. Use the the startNode command to start the node agent.
3. Verify that the server started successfully by checking the `*install_root*/profiles/*profile_name*/logs/*server_name*/startServer.log` log file for the message `Server nodeagent open for e-business; process id is nnnn`.

Example

- To start the node agent in the default profile, type `startNode`
- To list the options, type `startNode -help`
- To start the node agent in the Custom03 profile, type `startNode -profileName Custom03`
- To start the node agent in the Custom03 profile and write trace information to the log file called *install_root*/profiles/Custom03/logs/startServer.log, type `startNode -logfile -profileName Custom03`

What to do next

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

Stopping a node agent

Use the administrative tools when you need to stop a node agent (for example, to change the system clock). Node agents are administrative agents that represent a node to your system and manage the servers on that node.

Before you begin

Stop all servers that are managed by the node agent before you stop the node agent.

About this task

To stop a node agent, perform the following steps.

Procedure

1. From the administrative console of the deployment manager, click **System administration** → **Node agents**.
2. Select the node agent from the list on the Node Agent collection page.
3. Click **Stop**.

What to do next

Restart your node agent.

Restarting a node agent

Use the administrative tools to restart a stopped node agent.

About this task

To restart a stopped node agent, perform the following steps.

Procedure

1. From the administrative console of the deployment manager, click **System administration** → **Node agents**.
2. Select the node agent from the list on the Node Agent collection page.
3. Click **Restart**.

Starting and stopping deployment environments

You can start or stop deployment environments based on IBM-supplied patterns directly from the administrative console. You cannot manage custom deployment environments with this procedure.

Before you begin

- Verify that deployment environments exist on this deployment manager.
- Navigate to the administrative console of a deployment manager **Servers** → **Deployment Environments**.

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

To start or stop a deployment environment, the deployment environment must exist.

About this task

Follow these steps when you want to start or stop a deployment environments based on IBM-supplied patterns.

Note: To start or stop a custom deployment environment, you must start and stop its clusters individually.

Procedure

1. Select the check boxes next to the names of the deployment environments to start or stop.
2. Take one of the following actions:

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

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

Results

The display refreshes to indicate the status of the deployment environments.

Related information

➡ Using the Administration Thin Client

➡ Deployment environment status information

➡ Deployment environment function status

➡ Deployment environment status

➡ startDeploymentEnv command

➡ stopDeploymentEnv command

Starting the deployment environment using the command line

You can start the deployment environment using the wsadmin command.

Before you begin

Ensure the wsadmin client can connect to the deployment manager for the deployment environment.

Required security role for this task: When security and role-based authorization are enabled, you must use a user ID and password with administrator or operator authority to perform this task.

About this task

To start the deployment environment with the wsadmin command, perform the following steps.

Procedure

1. Open a command window.

2. At the command prompt, enter the wsadmin command to enter the command environment. The wsadmin command is located in either the <WPS>/profiles/<dmgr profile>/bin directory or the <WPS>/bin directory.
3. Enter the startDeploymentEnv command to start the deployment environment.
4. If administrative security is on, enter your user ID and password when prompted.

Example

This example starts the deployment environment (**MyDepEnv**) on the host (**myDmgr**) with administrative security enabled.

Note: If you are running the wsadmin client from the deployment manager bin folder, you do not need to include the `-host` and `-port` parameters in the command.

```
wsadmin -connType SOAP -host myDmgr -port 8879 -user dmgradmin -password dmgrpass
> $AdminTask startDeploymentEnv {-topologyName myDepEnv}
```

The `-connType` parameter specifies the type of connection to be used; the default argument is `SOAP`. If you are using a SOAP connection, including this parameter is optional.

The `-host` parameter specifies the host used for the SOAP or RMI connection. The default value for `-host` is the local host. If the node is running on the local host, you don not need to include this parameter.

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

Stopping the deployment environment using the command line

You can stop the deployment environment using the wsadmin command.

Before you begin

Ensure the wsadmin client can connect to the deployment manager for the deployment environment.

Required security role for this task: When security and role-based authorization are enabled, you must use a user ID and password with administrator or operator authority to perform this task.

About this task

To stop the deployment environment with the wsadmin command, perform the following steps.

Procedure

1. Open a command window.
2. At the command prompt, enter the wsadmin command to enter the command environment. The wsadmin command is located in either the <WPS>/profiles/<dmgr profile>/bin directory or the <WPS>/bin directory.
3. Enter the stopDeploymentEnv command to stop the deployment environment.
4. If administrative security is on, enter your user ID and password when prompted.

Example

This example stops the deployment environment (**MyDepEnv**) on the host (**myDmgr**) with administrative security enabled.

Note: If you are running the admin client from the deployment manager bin folder, you do not need to include the -host and -port parameters in the command.

```
wsadmin -connType SOAP -host myDmgr -port 8879 -user dmgradmin -password dmgrpass
> $AdminTask stopDeploymentEnv {-topologyName myDepEnv}
```

The -connType parameter specifies the type of connection to be used; the default argument is SOAP. If you are using a SOAP connection, including this parameter is optional.

The -host parameter specifies the host used for the SOAP or RMI connection. The default value for -host is the local host. If the node is running on the local host, you don not need to include this parameter.

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

Starting a cluster

You can start all the servers in a cluster (cluster members) in one action. When you start a cluster you automatically enable workload management.

Before you begin

- Ensure that the node agents are running.
- Verify that all resources required by applications deployed to the cluster are available.
- Start all prerequisite subsystems.

About this task

When you request that all members of a cluster start, the cluster state changes to `partially started` and each server that is a member of that cluster launches, if it is not already running. After all members of the cluster are running, the cluster states changes to `running`.

The **Ripplestart** option first stops and then starts each server in turn.

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

If you use a deployment environment pattern of *Remote Messaging* or *Remote Messaging and Remote Support*, there can be multiple clusters that depend on one another. If such a case exists, start the infrastructure and the clusters as follows to avoid potential startup problems:

1. Infrastructure startup sequence:
   a. Database, Lightweight Directory Access Protocol (LDAP), and Web servers
   b. Deployment manager (if needed)
   c. Node agents
2. Cluster startup sequence:
   a. Messaging infrastructure cluster
   b. Support cluster (CEI)
   c. Application deployment cluster

To start a cluster, perform the following steps.

Procedure

1. From the administrative console of the deployment manager, click **Servers** → **Clusters**.
2. Select the cluster you want to start.
3. Start the cluster.

   Note: If you use a deployment environment pattern of *Remote Messaging* or *Remote Messaging and Remote Support* make sure that you have started infrastructure components and that you start the clusters in the proper sequence, as described in About this task.

   - If your servers are stopped, click **Start**. This option calls the node agent for each server to launch the server process. If a call to a node agent for a server fails, the server does not start.
   - If your servers are running, click **Ripplestart** to first stop and then restart each cluster member.

     Note: The **Ripplestart** option restarts servers in sequence and ensures that at least one server in the cluster is online to handle requests.

Stopping a cluster

You can stop all the servers that are members of the same cluster at the same time by stopping the cluster.

Before you begin

Make sure there is no work in progress; performance monitoring infrastructure counters can indicate whether all queued work is complete. In addition, prevent new work from starting by disabling HTTP and IIOP traffic on the cluster members and quiescing the service integration buses.

Procedure

1. From the administrative console of the deployment manager, click **Servers** → **Clusters**.
2. Select the cluster you want to stop.
3. Click **Stop** or **Immediate Stop** to stop the cluster.
   - **Stop** halts each server in such a way that the server can finish work in progress. This option allows failover to another cluster member.
   - **Immediate Stop** halts each server quickly, ignoring any current or waiting tasks.

Chapter 4. Administering deployment environments

Through the administrative console on the deployment manager you administer the deployment environments defined on the deployment manager. You can also create, delete, import, and export deployment environments from the administrative console.

Before you begin

Verify that a deployment manager is started and log in to the administrative console.

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

About this task

Administer deployment environments when you need to update the deployment environments managed by a deployment manager. The **Deployment Environments** administrative console page is the starting point for all tasks related to the management and definition of deployment environments defined to a particular deployment manager.

Procedure

1. In the administrative console, click **Servers > Deployment Environments**.
2. To display the components of a deployment environment, click its name.
3. For existing environments, select the check box next to the deployment environments to manage and click one of the following buttons:

Function	Task
Start or **Stop**	Start and stop deployment environments
Remove	Remove resources from a deployment environment. This option does not delete the resources.
Export	Export deployment environments

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

What to do next

Manage deployment environment entities.

Related information

- Updating the deployment environment topology
- Configuring host aliases
- Configuring authentication aliases for a deployment environment
- Configuring custom deployment environments
- Configuring deferred configurations for a deployment environment
- Configuring deployment environments using the command line

Modifying the deployment topology

Use the Deployment Topology page to manage the topology configuration for your IBM-supplied patterns. Managing the configuration can involve adding and replacing nodes, as well as changing the number of cluster members.

Before you begin

Navigate to the administrative console of a deployment manager **Servers** → **Deployment Environments** → ***deployment_environment_name*** → **Additional Properties** → **Deployment Topology**.

About this task

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

By adding nodes you can increase the overall work capacity of the system.

- Select an objective and perform the associated actions.

Objective	Actions
Add an unnamed, empty node	Click **Unnamed node** and then **Add**.
Add a new node	Click **New node**, enter the node name, and then click **Add**.
Add an existing node	Click **Existing node**, select the node name from the list, and then click **Add**.
Replace a node in the deployment environment	Click **Select** next to the node to replace, click either **Unnamed node**, **New node**, or **Existing node** depending on how you are replacing the selected node, and then click **Replace Selected**.
Change the number of cluster members involved in each function	Type the number in the entry field underneath the columns labeled: • Application Deployment Target • Messaging Infrastructure • Supporting Infrastructure **Remember:** You must have at least one cluster member assigned for each function.

- Click **Apply** to keep the updates and remain on the Deployment Topology page. Click **OK** to keep the updates and return to the previous page.

What to do next

Either save the changes or discard them.

Managing deployment environment resources

You can manage the resources of your deployment environment to address changing requirements over time.

Before you begin

- Verify that deployment environments exist on this deployment manager.
- On the administrative console of the deployment manager navigate to **Servers > Deployment Environments**.
- You must completely stop any nodes you are removing from the deployment environment before removing those nodes.

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

About this task

As your deployment environment needs evolve, you can manage its resources to address new demands and processing requirements.

In managing the resources of a deployment environment, you can do the following:

- Add or remove servers and clusters.
- Change which nodes participate in specific functions.
- Change the configuration of data sources.
- Change authentication aliases.
- Obtain information on how to configure databases or tables if you previously deferred that operation.

Procedure

1. Select the deployment environment for which you want to manage resources by clicking the name. The system displays the Deployment Environment Configuration page which lists:
   - **Deployment Environment**
   - **Deployment Environment Pattern**
   - **Description**
   - **Deployment Environment Status**
   - **Deployment Environment Functions**
   - Links to the configuration pages
2. Select the configuration area of the deployment environment to manage. Select each link until you complete your changes.

Configuration area	Available actions
Additional Properties	**Deployment Topology** To change the configuration of a deployment environment based on IBM-supplied patterns. **Deferred Configuration** To determine any manual steps needed to complete the configuration of this deployment environment.
Related Items	**Data Sources** To change the data source configuration for the various components within the deployment environment. **Authentication Aliases** To change the authentication alias or password for components within the deployment environment.

3. Complete the configuration by choosing the option for the result needed.

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

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

What to do next

Manage the deployment environment.

Related information

Updating the deployment environment topology

Configuring custom deployment environments

Configuring a data source for your deployment environment

Configuring authentication aliases for a deployment environment

Configuring deferred configurations for a deployment environment

Editing the data source configuration

After you create a deployment environment, you can edit the data source configuration. The Data Sources page lists all of the data sources in your deployment environment, and you can perform multiple edits on this page.

Before you begin

- Verify that deployment environments exist on this deployment manager.
- Navigate to the administrative console of a deployment manager **Servers** → **Deployment Environments** → *deployment_environment_name* → **Related Items** → **Data Sources**.

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

About this task

The Data Sources page allows you to edit the collection of all the data sources in your deployment environment. Although you can edit data sources on this page, you cannot add a new data source here. The number of text fields might differ depending on the component and data source provider for each data source.

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

Procedure

1. On the Data Sources page, select the component that contains the data source to edit.
2. Make any required changes.
3. Click **Apply** or **OK** to save the changes.

Related information

Configuring a data source for your deployment environment

Editing your database provider

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

Before you begin

- Verify that deployment environments exist on this deployment manager.

- Navigate to the administrative console of a deployment manager **Servers** → **Deployment Environments** → ***deployment_environment_name*** → **Related Items** → **Data Sources**.

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

About this task

Use this procedure when you need to make changes to the configuration of a database provider used by a data source. Some sections of the Database Provider Configuration page, such as **Component Specific Properties**, have a different number of text boxes depending on the database provider.

Procedure

1. On the Data Sources page, select a data source for the data source provider you want to edit.
2. Click **Edit Provider** to open the Database Source Provider Configuration page.
3. Make any required changes.
4. Click **Apply** or **OK** to save your changes.

Related information

➡ Database specifications

➡ Common database specifications

Editing a data source in your deployment environment

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

Before you begin

- Verify that deployment environments exist on this deployment manager.
- Navigate to the administrative console of a deployment manager **Servers** → **Deployment Environments** → ***deployment_environment_name*** → **Related Items** → **Data Sources**.

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

About this task

Use this procedure to update the data sources used by a deployment environment through the Data Sources page. You cannot add a new data source in the Data Sources page. You can edit data source information by either clicking the data source name or selecting the component. Some text boxes are unavailable and you cannot change these values.

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

Procedure

1. On the Data Sources page, select the component that contains the data source you want to change and click **Edit**.
2. Edit the relevant information.
3. Click **Apply** or **OK** to save your changes.

Stopping and restarting the deployment manager

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

Before you begin

Verify that a deployment manager is started and log in to the administrative console.

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

Procedure

1. Choose a method to stop the deployment manager.

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

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

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

What to do next

Verify that the application deployment target cluster can start.

Stopping and restarting a cluster member

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

Before you begin

1. Prevent new work from entering the cluster member:

- If you are using the IBM® HTTP Server, change the plugin_cfg.xml file to remove the cluster member for HTTP traffic. If you are using another HTTP server, follow the directions for your HTTP server to remove the cluster member.
- For IIOP traffic, set the runtime weight to zero for the cluster member.
- Quiesce the service integration bus.

2. Verify that work that is destined for the cluster member is complete. Either wait a period of time or use Performance Monitoring Infrastructure counters to determine when the cluster completes all of the queued work.

About this task

Some configuration changes require you to stop and restart server processes before the configuration change takes effect. This involves stopping and restarting of the deployment manager, cluster member, and node agent.

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

Procedure

1. In the administrative console, navigate to **Servers** → **Application servers**.
2. Select the servers or cluster members to be stopped and click **Stop**.
3. Wait for the servers or cluster members to stop.
4. Select the servers or cluster members to be restarted and click **Start**.
5. Wait for the servers or cluster members to start.

   Note: Alternatively, you can stop and restart cluster members from the command line using the stopServer and startServer commands for your operating system or from the administrative console cluster panel by selecting **Servers** → **Clusters** → ***cluster_name*** → ***cluster_member_name***.

Starting and stopping deployment environments

You can start or stop deployment environments based on IBM-supplied patterns directly from the administrative console. You cannot manage custom deployment environments with this procedure.

Before you begin

- Verify that deployment environments exist on this deployment manager.
- Navigate to the administrative console of a deployment manager **Servers** → **Deployment Environments**.

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

To start or stop a deployment environment, the deployment environment must exist.

About this task

Follow these steps when you want to start or stop a deployment environments based on IBM-supplied patterns.

Note: To start or stop a custom deployment environment, you must start and stop its clusters individually.

Procedure

1. Select the check boxes next to the names of the deployment environments to start or stop.
2. Take one of the following actions:

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

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

Results

The display refreshes to indicate the status of the deployment environments.

Related information

Using the Administration Thin Client

Deployment environment status information

Deployment environment function status

Deployment environment status

startDeploymentEnv command

stopDeploymentEnv command

Exporting deployment environment definitions using the administrative console

Exporting deployment environment definitions helps you speed the implementation of deployment environments by minimizing the configuration on each deployment manager. You can use the exported deployment environment on other deployment managers as a template for the deployment environment. You can also replicate the same deployment environment configuration on a large scale.

Before you begin

- Define at least one deployment environment on a deployment manager.
- Log in to the administrative console of the deployment manager from which you are exporting the deployment environment definitions.
- On the administrative console of the deployment manager navigate to **Servers > Deployment Environments**.

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

About this task

If you are implementing a number of deployment environments based on the same designs, you can export the deployment environment definitions to use as templates for deployment environments on other deployment managers.

Procedure

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

Number of deployment environments to export	Action
One	At the prompt, enter the name of the exported file. The default name is *deployment_environment_name*.xml. To change the default name, specify the full file path.
Multiple	At the prompt, enter the output directory in which to place the exported compressed file that contains the deployment environment definitions. By default, the system names the compressed file *first_env_name*.zip. To change the default name, specify the full file path. **Note:** You cannot directly import a compressed file, you must extract the deployment environment definitions to the target file system.

3. Verify that the system created the files.

What to do next

You can import the exported files to other deployment managers.

Exporting deployment environment definitions using the command line

You can export deployment environment definitions using the wsadmin command. You can use the wsadmin command to perform the same definition-export task that you perform in the administrative console. This capability allows you to use a script to export large numbers of deployment environment definitions from a deployment manager freeing the administrative console for other tasks and enables you to replicate working configurations to other deployment managers.

Before you begin

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

Required security role for this task: When security and role-based authorization are enabled, you must use a userid and password with administrator or operator authority to perform this task.

About this task

You can use the command line to export deployment environment definitions in the following situations:

- You must export multiple deployment environment definitions and prefer to use the command line.
- You prefer to use the command line to export one deployment environment definition.
- You must export a large number of deployment environment definitions; using wsadmin reduces the time for performing the task.

Procedure

1. Open a command window.

   The wsadmin command can be found at either the <WPS>/profiles/<dmgr profile>/bin directory, or the <WPS>/bin directory.
2. At the command prompt, enter the wsadmin command to enter the wsadmin environment.
3. Use the exportDeploymentEnvDef command to export the deployment environment definition from the deployment manager to an output file. The file name is in the form *depEnvName*.xml

   Note: If administrative security is on, you will be prompted for a user ID and password, if you do not supply it in the command.

Example

This example exports the deployment environment **myDepEnv** on the host **myDmgr** with administrative security enabled.

Note: If you are running the admin client from the deployment manager bin folder, you do not need to include the `-host` and `-port` parameters in the command.

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

The `-connType` parameter specifies the type of connection to be used; the default argument is `SOAP`.

Note: As the default is `SOAP`, you do not need to give explicitly if SOAP is the connection type that is being used.

The `-host` parameter specifies the host used for the SOAP or RMI connection. The default value for `-host` is the local host.

Note: If the node is running on the local host, you don not need to specify `-host`

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

Related information

exportDeploymentEnvDef command

Importing deployment environment definitions using the administrative console

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

Before you begin

- On the administrative console of the deployment manager navigate to **Servers > Deployment Environments**.
- You must have a copy of an exported deployment environment definition from another deployment manager.
- You must be able to access the deployment environment definition file from the deployment manager into which you are importing the deployment environment definition.
- The deployment manager that imports the deployment environment definition must support at least all of the functions that are defined in the deployment environment definition. For example, you can import a deployment environment definition that was created on a WebSphere Enterprise Service Bus deployment manager into a WebSphere Process Server deployment manager but not vice versa.

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

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

About this task

Importing an existing deployment environment definition to create a new one can minimize the amount of time you spend configuring a deployment environment. If an existing environment is similar to the one you want to create, export it and then import it into the deployment manager you are configuring.

Procedure

1. Click **Import** in the Deployment Environments page to launch the Deployment Environment Configuration wizard.

   The wizard starts with **Load an external deployment environment definition** selected.
2. Click **Browse** to open a file dialog and select the deployment environment definition file to import or type the full path to it.
3. Click **Next** to load the configuration and launch the Import deployment environment wizard.

   The wizard displays the Select Nodes page, unless all the node names match currently federated notes. If all nodes match, the wizard displays the Database page.

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

4. Optional: From the list of possible nodes on the Select Nodes page, select the nodes to include in the deployment environment and click **Next**.

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

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

   On this page, define the database information for the components that are included in this deployment environment. Where possible, the wizard supplies default information for the parameters, but change those values to match the values that you defined when you planned the environment.
6. Optional: Define the Business Process Choreographer configuration and then click **Next**.

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

   Note: This page displays only if the value **WPS** is selected for **Runtime Capability**.
7. Optional: On the Business Rules Manager page, specify the context root for the Business Rules Manager and click **Next**.

   Note: This page displays only if the value **WPS** is selected for **Runtime Capability**.

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

Results

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

What to do next

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

Related tasks

Exporting deployment environment definitions using the administrative console
Exporting deployment environment definitions helps you speed the implementation of deployment environments by minimizing the configuration on each deployment manager. You can use the exported deployment environment on other deployment managers as a template for the deployment environment. You can also replicate the same deployment environment configuration on a large scale.

Importing deployment environment definitions using the command line

You can import deployment environment definitions using the wsadmin command. You can use the wsadmin command to perform the same definition-import task that you perform in the administrative console. This capability allows you to use a script to import many deployment environment definitions to a deployment manager freeing the administrative console for other tasks and enables you to replicate working configurations to other deployment managers.

Before you begin

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

Required security role for this task: When security and role-based authorization are enabled, you must use a userid and password with administrator or operator authority to perform this task.

About this task

Use the command line to import deployment environment definitions in the following situations:

- You must import multiple deployment environment definitions and prefer to use the command line.
- You prefer to use the command line to import one deployment environment definition as a template for multiple deployment environments.
- You must import a large number of deployment environment definitions; using wsadmin reduces the time for performing the task.

Procedure

1. Open a command window.

   The wsadmin command can be found at either the <WPS>/profiles/<dmgr profile>/bin directory, or the <WPS>/bin directory.
2. Copy the deployment environment definition file you are importing to the system.
3. Enter the wsadmin command to enter the wsadmin environment

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

   Note: If administrative security is on, you will be prompted for a user ID and password, if you do not supply it in the command.

Example

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

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

The `-connType` parameter specifies the type of connection to be used; the default argument is `SOAP`.

Note: As the default is `SOAP`, you do not need to give explicitly if SOAP is the connection type that is being used.

The `-host` parameter specifies the host used for the SOAP or RMI connection. The default value for `-host` is the local host.

Note: If the node is running on the local host, you don not need to specify `-host`

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

What to do next

Optiional: Validate the imported deployment environments.

Related information

Managing node agents

importDeploymentEnvDef command

Removing deployment environments

Removing a deployment environment removes the management entity of the deployment environment. Deleting the deployment environment does not remove or change the configuration of the servers, nodes, and clusters that make up the deployment environment. Deleting deployment environments might be the final phase of moving a deployment environment from one deployment manager to another.

Before you begin

- On the administrative console of the deployment manager navigate to **Servers > Deployment Environments**.
- Verify that deployment environments exist on this deployment manager.
- For recover purposes, consider exporting the deployment environment definition.

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

About this task

When you no longer need to manage the resources of a specific deployment environment as a group, remove that deployment environment definition from the deployment manager.

Procedure

1. On the Deployment Environment page, select the check box next to the deployment environments to remove and click **Remove**.

   The system removes the deployment environment from the display.
2. Click **Save** to save this change to the master configuration or **Discard** to prevent the update of the master configuration.

Related tasks

Exporting deployment environment definitions using the administrative console
Exporting deployment environment definitions helps you speed the implementation of deployment environments by minimizing the configuration on each deployment manager. You can use the exported deployment environment on other deployment managers as a template for the deployment environment. You can also replicate the same deployment environment configuration on a large scale.

Chapter 5. Administering applications and application services

Applications for WebSphere Process Server involve similar administration tasks and interfaces as J2EE applications for WebSphere Application Server, with some additional tasks specifically relating to service applications, service modules, WebSphere MQ destinations, and other resources.

Administering service applications and service modules

Use the administrative tools to view and manage service applications and their associated service modules.

Before you begin

Deploy your service modules to the runtime environment.

About this task

A service module is a Service Component Architecture (SCA) module that provides services in the run time. When you deploy a service module to WebSphere Process Server, you build an associated service application that is packaged as an Enterprise Archive (EAR) file.

Service modules are the basic units of deployment and can contain components, libraries, and staging modules used by the associated service application. Service modules have exports and, optionally, imports to define the relationships between modules and service requesters and providers. WebSphere Process Server supports modules for business services and mediation modules. Both modules and mediation modules are types of SCA modules. A mediation module allows communication between applications by transforming the service invocation to a format understood by the target, passing the request to the target and returning the result to the originator. A module for a business service implements the logic of a business process. However, a module can also include the same mediation logic that can be packaged in a mediation module.

Related tasks

Deploying a module
You can deploy a module or a mediation module, as generated by WebSphere® Integration Developer, into a production WebSphere Process Server environment using these steps.

Versioning in service applications

Service applications support versioning. You can develop and deploy one or more versions of a module and its artifacts into the runtime environment for use by specific clients.

What can be versioned?

A module can have a version number, as can the SCA import and export bindings in a module. SCA bindings inherit their version information from the module they are associated with.

Note: At this time, SCA bindings are the only binding type that can be versioned. Versioning is optional for 6.2.x modules. Modules developed and deployed with WebSphere Integration Developer and WebSphere Process Server 6.1.x do not have versions and continue to function with their current behavior. Refer to the migration topics for more information.

Libraries can also be versioned. Modules that use a library have a dependency on a specific version of that library, and libraries can also have dependencies on specific versions of other libraries. See the WebSphere Integration Developer Information Center for details about versioning libraries.

Considerations for deploying versioned modules

You can deploy a versioned module into the 6.2.x run time and administer it from the SCA Modules pages within the administrative console.WebSphere Process Server supports the following versioned deployment scenarios

- Installing a versioned module to a server or cluster in a cell
- Installing the same version of a module once to each of one or more servers or clusters in a cell
- Installing different versions of a module on the same server or cluster

Deploying a new version of a module does not replace any previous versions of the module. Previous versions of cell-scoped application artifacts (in this case, business rules) are overwritten.

If you want to update an application (for example, to make minor corrections or improvements) without changing the version, that updated application and its artifacts will replace the existing application and artifacts, with the exception of any defined security policies. All security policy artifacts are preserved during an application update.

In order to preserve versioning information, the installation process automatically changes the module name (via the serviceDeploy or createVersionedSCAModule command) to ensure it is unique within the cell. This change is accomplished by adding the version number, a unique cell ID, or both to the original module name.

```
moduleName_vversionValue_uniqueCellID
```

Considerations for binding versioned modules

After you have deployed multiple versions of a module on a server or multiple instances of a module across clusters, consider how to bind specific versions of modules to clients (which may or may not be versioned).

- Static binding: If you are using static binding, simply use the existing administrative tools to bind a versioned module to a client. You must specify the module version number in the static binding.
- Dynamic binding: To use dynamic binding with versioned modules, use a mediation flow component that contains the module version metadata (versionValue and versionProvider) and service-version-aware routing. Note that in order to use service-version-aware routing to dynamically bind versioned modules, all modules must be registered with WebSphere Service Registry and Repository (WSRR).

Service application features of the administrative interfaces

WebSphere Process Server allows you to use the administrative console to view and change aspects of service applications and service modules.

Service applications provide services, and have an associated service module (also called aService Component Architecture (SCA) module).

Viewable module details

After you have deployed an EAR (Enterprise ARchive) file containing an SCA module, you can view SCA module details. You can list all your SCA modules, and their associated applications, and you can view details about a particular SCA module.

The SCA module details you can view include some of the following:
- SCA module name.
- Associated application.
- SCA module imports:
  - Interfaces.
  - Bindings.
- SCA module exports:
  - Interfaces.
  - Bindings.
- SCA module properties.

Modifiable module details

After you have deployed an EAR file containing an SCA module you can change the following SCA module details using the administrative console, without having to redeploy the EAR file.
- Import bindings of type SCA:
  - Changing import bindings lets you change service interactions.
  - SCA bindings connect SCA modules to other SCA modules. One SCA module can interact with a second SCA module, and can be changed to interact with another SCA module.
  - Web service bindings connect SCA modules to external services using SOAP.
- Import bindings of type Web service (WS):
  - Changing import bindings lets you change service interactions.
  - WS import bindings allow SCA modules to access web services. A WS import binding calls a service located at a specified endpoint. You can change the end point such that the binding calls the service at an alternative end point, or even an entirely different service with compatible interfaces.
- Export and import bindings of types JMS, WebSphere MQ JMS, generic JMS, WebSphere MQ, and HTTP have attributes that you can modify.
- Mediation module properties:
  - Mediation module properties belong to the mediation primitives with which they are associated. However, the WebSphere Process Server administrative console displays some of them as Additional Properties of an SCA module. The integration developer must flag a mediation primitive property as Promoted in order for it to be visible from WebSphere Process Server.
  - Changing mediation module properties lets you change the behavior of your mediations. The mediation changes that you can make depend upon the properties that have been promoted.

Note: An export with no binding specified is interpreted by the runtime as an export with an SCA binding.

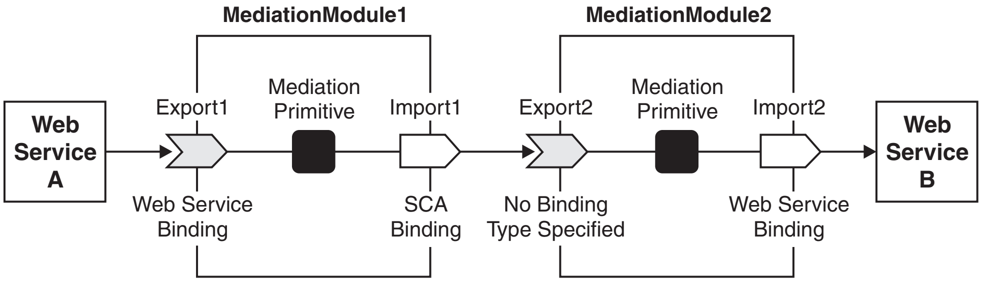


Figure 2. Example showing one mediation module interacting with another mediation module. Mediation Module1 connects to Mediation Module2

Administering service modules

You can list the service modules that have been deployed to WebSphere Process Server, view information associated with individual service modules, and make changes to some import bindings.

About this task

After deploying service applications, use the administrative console to list and administer all of the associated service modules, including mediation modules.

Procedure

1. Open the administrative console.
2. Click **Applications** → **SCA Modules** to list the available service modules.

Results

The content pane displays the service modules that have been deployed to WebSphere Process Server. You can also see the applications that the modules are associated with, and whether those applications are running.

Displaying details of a service module

You can display information on service modules that have been deployed to WebSphere Process Server.

About this task

To display details about the deployed service module use the administrative console to complete the following steps.

Procedure

1. In the navigation pane, expand **Applications** → **SCA Modules**, to display the SCA modules.
2. In the content pane, click the SCA module to choose an SCA module.

Results

The content pane displays the SCA module name and description; the name of the associated enterprise application; expandable lists of imports and exports; and a module properties link.

Displaying details of the application for a service module

You can display details about the application used to deploy a service module to WebSphere Process Server.

About this task

The application used to deploy a service module defines a range of configuration properties that affect the use of the 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 service module, use the administrative console to complete the following steps.

Procedure

1. Expand **Applications** → **SCA Modules** in the navigation pane to display the SCA modules.
2. In the column labeled **Application**, click the application name, to choose an SCA module.

Results

The content pane displays the application details page that provides the application's configuration properties and, if appropriate, local topology.

What to do next

From this page, you can review and, if needed, change the configuration properties for the application and link to additional console pages, as described in Configuring an application.

Starting and stopping service modules

You can start a service module that has a status of Stopped or stop one that is running that has a status of Started. To change the status of a service module, start or stop the application used to deploy the module.

Before you begin

Before you can start or stop the application for a service module, you must have deployed the module into WebSphere Process Server.

About this task

To use the services of a service module and associated components, start the associated application. By default, the application starts automatically when the server starts.

You can manually start and stop applications using the following administrative tools:

- Administrative console
- wsadmin startApplication and stopApplication commands
- Java programs that use ApplicationManager or AppManagement MBeans

To start or stop a service module, use the administrative console you complete the following steps.

Procedure

1. Expand **Applications** → **SCA Modules**, in the navigation pane, to list the SCA modules.
2. Select the check box for the SCA module that you want to start or stop.
3. Click the Start or Stop button

Option	Description
Start	Runs the application and changes the state of the application to *Started*. The status is changed to *partially started* if not all servers on which the application is deployed are running.
Stop	Stops the processing of the application and changes the state of the application to *Stopped*.

4. Click **Stop** or select the application you want to restart, then click **Start** to restart a running application.

Results

The status of the application changes and a message stating that the application started or stopped displays at the top the page.

What to do next

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 service module properties

You can display the properties of service modules that have been deployed to WebSphere Process Server.

About this task

You might want to check that property values are what you expect before running a service application.

To display the properties of deployed service modules, use the administrative console to complete the following steps.

Procedure

1. In the navigation pane, expand **Applications** → **SCA Modules** to display the SCA modules
2. Click the required SCA module, in the content pane, to choose an SCA module.
3. Click **Module Properties**, under Additional Properties, in the content pane, to list the SCA module properties.

4. Optional: Expand the group whose properties you want to view. If properties belong to a group they are displayed inside an expandable section; if they do not belong to a group you can view them immediately.

Results

The content pane displays the updatable properties for the SCA module in a table that shows property names, types and values. Only property values are updatable from the administrative console: to change property groups, names, and types you use WebSphere Integration Developer. A message is displayed if there are no properties that you can update.

Related concepts

Mediation modules
Mediation modules are Service Component Architecture (SCA) modules that can change the format, content, or target of service requests.

Related reference

Promotable properties

Changing service module properties

You can change the value of some service module properties.

About this task

You might want to change property values if the runtime environment changes.

To change the values of service module properties, use the administrative console to complete the following steps.

Procedure

1. Expand **Applications** → **SCA Modules** in the navigation pane to list the SCA modules.
2. Click a SCA module in the content pane to choose a SCA module.
3. Under Additional Properties, select **Module Properties**, in the content pane, to display the SCA module properties. This displays the module properties that you can update. Property groups, names, types, and values are displayed in the content pane, but you can only update property values. To change property groups, names, and types use WebSphere Integration Developer.
4. Optional: Expand the group whose properties you want to update. If properties belong to a group they are displayed inside an expandable section; if they do not belong to a group you can view them immediately.
5. Click a property value from the Properties table to choose a property value.
6. Enter a value that conforms to the property type to change a property value.
7. Click **OK** to save your changes. Then save your changes to the master configuration.

Results

The property values are changed. Generally, mediation flows use property changes immediately, unless the changes occur in a deployment manager cell. If changes occur in a deployment manager cell they take affect on each node in the cell after that node has been synchronized. Mediation flows that are in-flight at the time of the property value change continue to use previous values.

Related concepts

Mediation modules
Mediation modules are Service Component Architecture (SCA) modules that can change the format, content, or target of service requests.

Related reference

Promotable properties

Working with imports

You can list the imports of service modules that have been deployed to WebSphere Process Server. You can also display import interfaces and change the details of import bindings.

About this task

To list the imports of service modules that you have deployed, use the administrative console to complete the following steps.

Procedure

1. Expand **Applications** → **SCA Modules**, in the navigation pane to list the SCA modules.
2. Select the SCA module that you are interested in.
3. Expand **Imports**, in the content pane, under Module components to list the SCA module imports.

Results

The content pane displays a list of imports. If there are no imports an information message is displayed.

Related concepts

Imports and import bindings
Imports define interactions between Service Component Architecture (SCA) modules and service providers. SCA modules use imports to permit components to access external services (services that are outside the SCA module) using a local representation. Import bindings define the specific way that an external service is accessed.

Displaying an import interface:

You can display the import interfaces of service modules that have been deployed to WebSphere Process Server.

About this task

To display the import interfaces of service modules that you have deployed, use the administrative console to complete the following steps.

Procedure

1. Expand **Applications** → **SCA Modules**, in the navigation pane to list the SCA modules.
2. Select a SCA module.

3. Expand **Imports**, under Module components, in the content pane, to list the SCA module imports.
4. Expand the import you require, under Module components, in the content pane, to display import details.
5. Expand **Interfaces** to display import interfaces.
6. Select an interface.

Results

The content pane displays the WSDL (Web Services Description Language) interface.

Displaying an import binding:

You can display Web service, SCA, JMS or Adapter types of import bindings, after you deploy service modules to WebSphere Process Server.

About this task

To display the import bindings of service modules, use the administrative console to complete the following steps.

Procedure

1. Expand **Applications** → **SCA Modules**, in the navigation plane, to list the SCA modules.
2. Select an SCA module.
3. Expand **Imports**, under Module components, in the content pane, to list the SCA module imports.
4. Expand the import you require, under Module components, in the content pane, to display import details.
5. Expand **Binding** to display import bindings.
6. Select a binding.

Results

The content pane displays the import binding details.

Related concepts

Working with bindings
At the core of a service-oriented architecture is the concept of a *service*, a unit of functionality accomplished by an interaction between computing devices. An *export* defines the external interface (or access point) of a module, so that SCA components within the module can provide their services to external clients. An *import* defines an interface to services outside a module, so the services can be called from within the module. You use protocol-specific *bindings* with imports and exports to specify the means of transporting the data into or out of the module.

Changing an SCA import binding:

Using the administrative console, you can change the details of SCA import bindings. For information about changing the details of other types of binding, see the section on Bindings.

About this task

To change SCA import bindings of service modules, use the administrative console to complete the following steps.

Procedure

1. Expand **Applications** → **SCA Modules**, in the navigation pane, to list the SCA modules.
2. Choose an SCA module.
3. Expand **Imports**, in the content pane, under Module components, to list the SCA module imports.
4. Expand the import you require, in the content pane, under Module components, to display the import details.
5. Expand **Binding** to display import bindings.
6. Choose an SCA import binding. SCA import bindings are indicated using the identifier [SCA].
7. Choose a new target SCA module. Select a module from the **Target** drop-down menu. Selecting a different SCA module changes the exports and export interfaces that are displayed.
8. Select an export from the **Export** drop-down menu.
9. Save your changes to the master configuration.

Results

The SCA import binding is changed for the selected SCA module import.

WebSphere Process Server issues a warning for each import interface that is not satisfied by an export interface. WebSphere Process Server compares the WSDL (Web Services Description Language) port type names of import and export. If the port type names are not the same a warning is issued, but you are allowed to ignore the warning. However, if the port type names match WebSphere Process Server assumes that the operations provided are equivalent and no warning is issued.

Related concepts

Working with bindings
At the core of a service-oriented architecture is the concept of a *service*, a unit of functionality accomplished by an interaction between computing devices. An *export* defines the external interface (or access point) of a module, so that SCA components within the module can provide their services to external clients. An *import* defines an interface to services outside a module, so the services can be called from within the module. You use protocol-specific *bindings* with imports and exports to specify the means of transporting the data into or out of the module.

Changing a Web service import binding:

Using the administrative console, you can change the endpoint URL of a Web service import binding. For information about changing the details of other types of binding, see the section on Bindings.

About this task

To change web service import bindings of service modules, use the administrative console to complete the following steps.

Procedure

1. Expand **Applications** → **SCA Modules**, in the navigation pane, to list the SCA modules.
2. Choose an SCA module.
3. Expand **Imports**, in the content pane, under, Module components, to list the SCA module imports.
4. Expand the import you require, in the content pane, under Module components, to display import details.
5. Expand **Binding** to display import bindings.
6. Choose a web service import binding. Web service import bindings are indicated using the identifier [Web service].
7. Change the endpoint URL. Ensure the endpoint is a well-formed URL.
8. Save your changes to the master configuration.

Results

The web service import binding is changed for the selected SCA module import.

The changes take effect after you update the master configuration and restart the SCA module.

Related concepts

Working with bindings
At the core of a service-oriented architecture is the concept of a *service*, a unit of functionality accomplished by an interaction between computing devices. An *export* defines the external interface (or access point) of a module, so that SCA components within the module can provide their services to external clients. An *import* defines an interface to services outside a module, so the services can be called from within the module. You use protocol-specific *bindings* with imports and exports to specify the means of transporting the data into or out of the module.

Working with exports

You can list the exports of service modules that have been deployed to WebSphere Process Server. You can also display export interfaces and export bindings.

About this task

To list the exports of deployed service modules, use the administrative console to complete the following steps.

Procedure

1. Expand **Applications** → **SCA Modules**, in the navigation pane, to list the SCA modules.
2. Select the particular SCA module that you are interested in.
3. Expand **Exports**, in the content pane, under, Module components, to list the SCA module exports.

Results

The content pane displays a list of exports is displayed. If there are no exports, an information message is displayed.

Related concepts

Exports and export bindings
Exports define interactions between Service Component Architecture (SCA) modules and service requesters. SCA modules use exports to offer services to others. Export bindings define the specific way that an SCA module is accessed by service requesters.

Displaying an export interface:

You can display the export interfaces of service modules that have been deployed to WebSphere Process Server.

About this task

To display the export interfaces of service modules that you have deployed, use the administrative console to complete the following steps.

Procedure

1. Expand **Applications** → **SCA Modules** in the navigation pane to list the SCA modules.
2. Select an SCA module.
3. Expand **Exports**, in the content pane, under Module components, to list the SCA module, exports.
4. Expand the export you require, in the content pane, under Module components, to display export details.
5. Expand **Interfaces** to display export interfaces.
6. Select an interface.

Results

The content pane displays the WSDL (Web Services Description Language) interface.

Displaying an export binding:

You can display some types of export bindings after you deploy service modules to WebSphere Process Server.

About this task

To display the export bindings of service modules, use the administrative console to complete the following steps.

Procedure

1. Expand **Applications** → **SCA Modules** In the navigation pane to list the SCA modules.
2. Select an SCA module.
3. Expand **Exports**, in the content pane, under Module components, to list the SCA module exports.
4. expand the export you require, in the content pane, under Module components, to display export details.
5. Expand **Binding** to display export bindings.

6. Select a binding.

Administering 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 5. 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.

Table 5. Buttons for administering enterprise applications (continued)

Button	Resulting action
Export	Opens the Export Application EAR files page so you can export an enterprise application to an EAR file. Use the Export action to back up a deployed application and to preserve its binding information.
Export DDL	Opens the Export Application DDL files page so you can export DDL files in the EJB modules of an enterprise application.

For more information on administering applications, see the WebSphere Application Server Information Center.

Administering the throughput of SCA requests

For each SCA module deployed on WebSphere Process Server, requests being processed are held on queue points and in the data store for messaging engines. You can display the data for SCA requests, and take any appropriate action to manage the throughput of SCA requests.

About this task

When an SCA module is running in enterprise service bus, requests normally flow through the enterprise service bus without needing to be managed. Sometimes, you might want to check the throughput of a request, check the contents of a request, or if some problem has occurred, delete a request. You might also want to take other actions such as to monitor the overall throughput of requests, or change the reliability setting for requests.

Requests are handled as messages by the service integration technologies of the underlying WebSphere Application Server. For this reason, actions to manage requests are managed by using the WebSphere Application Server tasks to act on service integration messages.

This topic provides an overview of the main tasks that you might consider using, and links into the WebSphere Application Server tasks for more detailed information.

- Listing messages on a message point

  SCA requests that are being processed are held on queue points of the SCA.SYSTEM.bus. You can list the SCA requests either through a queue destination for a component of the SCA module, or through the messaging engine that hosts a queue point; for example: **Service integration** → **Buses** → **SCA.SYSTEM.localhostNode01Cell.Bus** → **Destinations** → **StockQuoteService_Export** → **Queue points** → **StockQuoteService_Export@localhostNode01.server1-SCA.SYSTEM.localhostNode01Cell.Bus** → **Runtime** → **Messages**
- Resolving locked messages on a message point

  If a problem occurs, an SCA request might remain locked on the queue point where it is being processed. You can display the message **State** property that indicates whether or not the request is locked, and if appropriate take further action to resolve the problem.
- Deleting messages on a message point

Under exceptional circumstances, you might need to delete one or more messages that exist on a message point for a selected bus destination or messaging engine. You should not normally need to delete messages on a message point. This task is intended as part of a troubleshooting procedure.

- Viewing data in the data store for a messaging engine.

  A messaging engine maintains 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.
- Changing message reliability for a 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.

Viewing data in a data store

A messaging engine maintains both volatile and durable data in its data store, including messages, transaction states, and communication channel states. You can use the database tools to view data in the data store for a messaging engine.

Before you begin

Before you can use the ij tool to view data in an embedded Derby data store for a messaging engine, you must have stopped the messaging engine.

About this task

Volatile data is lost when a messaging engine stops, in either a controlled or an uncontrolled manner. Durable data is available after the server restarts.

In some cases, you might want to view the data in a data store; for example, to examine the messages being processed by the messaging engine.

You can use the database tools for the data store to view data in the data store for a messaging engine. For example, if the messaging engine uses the embedded Derby database, you can use the ij tool to view request messages.

Procedure

1. Start the ij tool. On Windows® complete the following sub-steps:
   a. Open a command window
   b. Change directory to *profile_root*\derby\bin\embedded
   c. Type `ij.bat`

   On non-Windows platforms, complete the following sub-steps:
   a. Open a command window
   b. Change directory to *profile_root*/derby/bin/embedded
   c. Type `./ij.sh`
2. Open the data store for the messaging engine. Use the ij tool to complete the following sub-steps:
   a. Connect to the required database file.

      For a messaging engine, the database is stored in the directory *profile_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:

```
profile_root\profiles\default\databases\com.ibm.ws.sib\localhostNode01.server1-SCA.SYSTEM.localhostNode01Cell.Bus
```

b. Use the ij tool to issue SQL commands and view data.
   1) Change directory to *install_root*/derby/bin/embedded
   2) Type `./ij.sh`
   3) Type `protocol 'jdbc:derby:' ;`
   4) Type `connect 'profile_root/profiles/profile_name/databases/com.ibm.ws.sib/database_name' ;`

c. Optional: To display more help about using ij, type `help ;` at the `ij>` prompt.

Changing message reliability for a bus destination

Messages have a quality of service attribute that specifies the reliability of message delivery. You can select a reliability to suit your requirements for assured delivery, and system performance.

About this task

The administrator can specify the reliability setting on bus destinations, or the reliability can be specified by individual producers (typically under application control through an API call). The administrator can specify whether the default reliability for the destination can be overridden by a producer, and the maximum reliability that attached producers can request.

To browse or change the message reliability setting of a destination, use the administrative console to complete the following steps:

Procedure

1. Click **Service integration** → **Buses** in the navigation pane.
2. Click the name of the bus on which the destination exists in the content pane.
3. Click **Destinations**
4. Click the destination name. This displays the details page for the destination.
5. Review the reliability properties. The following properties control the message reliability for the destination:

   Default reliability
   The reliability assigned to a message produced to this destination when an explicit reliability has not been set by the producer.

   Maximum reliability
   The maximum reliability of messages accepted by this destination.

   These properties can have values from the following list:

   Best effort nonpersistent
   Messages are discarded when a messaging engine stops or fails. Messages may also be discarded if a connection used to send them becomes unavailable and as a result of constrained system resources.

   Express nonpersistent
   Messages are discarded when a messaging engine stops or fails. Messages may also be discarded if a connection used to send them becomes unavailable.

Reliable nonpersistent
Messages are discarded when a messaging engine stops or fails.

Reliable persistent
Messages may be discarded when a messaging engine fails.

Assured persistent
Messages are not discarded.

For more information about using these properties to control message reliability, see Message reliability levels.

6. Review whether producers can override the default reliability setting.

   Enable producers to override default reliability
   Select this option to enable producers to override the default reliability that is set on the destination.

7. Optional: Change the destination properties to suit your needs.

   You can further refine the configuration of a destination by setting other properties to suit your needs, as described in Configuring bus destinations.

8. Click **OK**.
9. Save your changes to the master configuration.

Doing more with service applications and service modules

You can use the WebSphere administrative console not only to manage service modules themselves, but also the resources used by the modules and the applications that contain the modules. You also can use commands to do these tasks.

About this task

The routine tasks for managing service modules are described in "Administering service modules" on page 70. For more advanced tasks, see the subtopics below.

Managing resources for service modules

Service modules uses resources provided by the service integration technologies of WebSphere Application Server. Service 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 service modules, you can use the WebSphere administrative console, commands, and scripting tools.

For more information about managing resources for service modules, see the related topics.

Service integration technologies

Service integration resources, such as bus destinations, enable a service module to use service integration technologies. Queue destinations are used by the SCA runtime exploited by the service module as a robust infrastructure to support asynchronous interactions between components and modules. When you install a service 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 service 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***/exportlink/***export_name*

This is the destination used by the export to send asynchronous requests to the module. Requests are routed to the component target linked to the export.

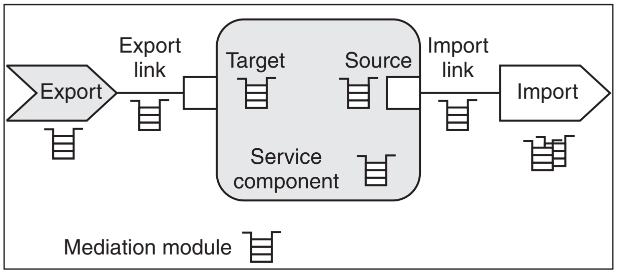
Queue sca/*module_name***/importlink/***import_name*

This is the destination used by the import to send asynchronous requests out of the module. Requests are routed to the module export linked to the import.

Queue sca/*module_name***/import/sca/dynamic/import/scaimport [for SCA binding]**

Queue sca/*module_name***/import/sca/dynamic/import/wsimport [for Web service binding]**

Queue sca/contextStore/*module_name*



For each of the destinations, a queue point is also created, and defined on the messaging engine of the relevant bus member.

You can deploy and use service 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 service module to use asynchronous messaging as a method of communication based on the Java Message Service (JMS) programming interface. The JMS support used depends on the JMS binding of the module. For example, a module with a JMS binding uses a JMS connection factory configured on the default messaging provider provided by the underlying WebSphere Application Server, while a module with a WebSphere MQ JMS binding uses a JMS connection factory configured on WebSphere MQ as the JMS provider. To manage use of the Java Message Service, you can administer the following resources:

JMS connection factory

A JMS connection factory is used to create connections to the associated JMS provider of JMS destinations, for both point-to-point and publish/subscribe messaging. Use connection factory administrative objects to manage JMS connection factories for the provider.

JMS queue

A JMS queue is used as a destination for point-to-point messaging. Use JMS queue destination administrative objects to manage JMS queues for the provider.

JMS topic
A JMS topic is used as a destination for publish/subscribe messaging. Use topic destination administrative objects to manage JMS topics for the provider.

JMS activation specification
A JMS activation specification is associated with one or more message-driven beans and provides the configuration necessary for them to receive messages.

JMS listener port
A JMS listener port defines the association between a connection factory, a destination, and a message-driven bean. This enables deployed message-driven beans associated with the port to retrieve messages from the destination.

Common Event Infrastructure (CEI)

CEI resources enable a service 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
This profile defines a JMS queue entry into the event bus. It defines the JNDI names for a JMS queue and queue connection factory.

Related tasks

Deploying a module
You can deploy a module or a mediation module, as generated by WebSphere® Integration Developer, into a production WebSphere Process Server environment using these steps.

Administering the Common Event Infrastructure

Related reference

Promotable properties

Managing service integration in applications

This set of topics provides information about the service integration technologies. Service integration is implemented as a group of messaging engines running in application servers (usually one engine to one server) in a cell.

A service integration bus is a form of managed communication that supports service integration through synchronous and asynchronous messaging. A bus consists of interconnecting messaging engines that manage bus resources. The members of a service integration bus are the application servers and clusters on which the messaging engines are defined.

Service Integration Bus Browser:

The Service Integration Bus Browser provides a single location for browsing and performing day-to-day operational tasks on service integration buses.

Examples of day-to-day operations include browsing service integration buses, viewing runtime properties for messaging engines, or managing messages on message points. The browser is not intended as a bus configuration tool.

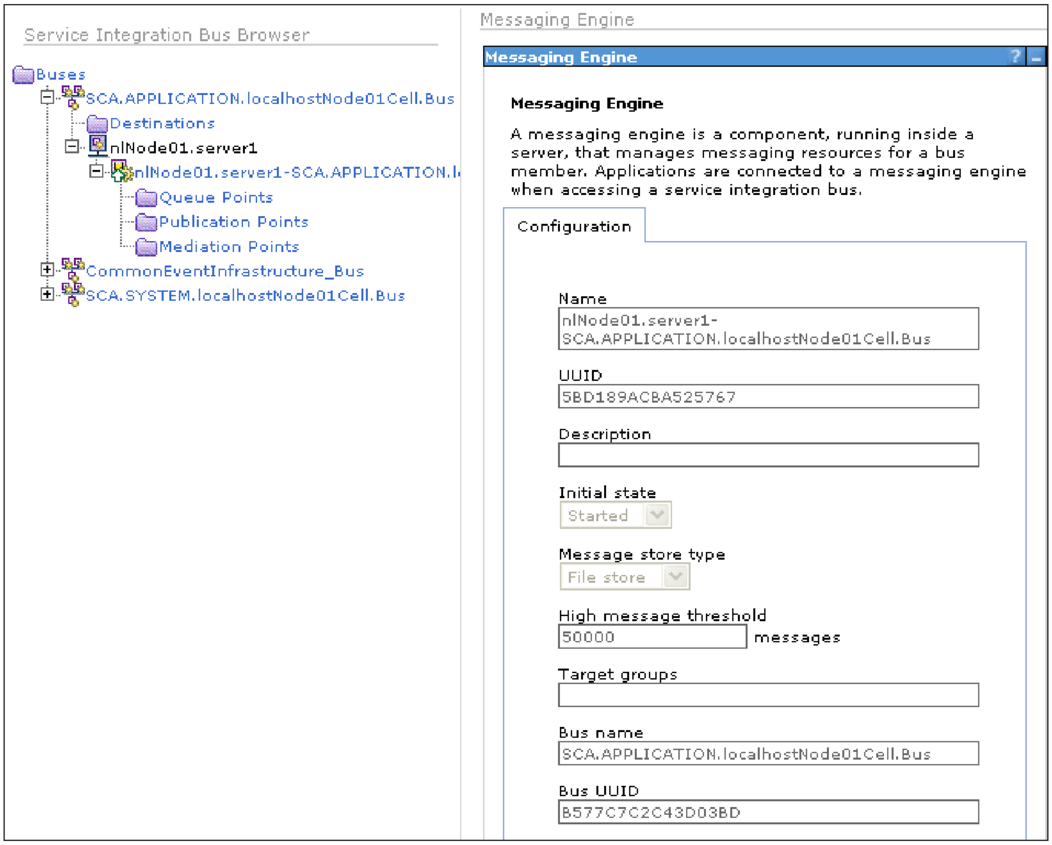
When you access the Service Integration Bus Browser by clicking **Service Integration** → **Service Integration Bus Browser**, two panes open to the right of the standard console navigation pane:

Navigation tree pane
This pane contains a navigation tree that allows you to browse the service integration buses configured on the system.

Content pane
This pane contains collection and detail pages for the buses and their individual components, such as messaging engines, queue points, destinations, publication points, and mediation points.

Note that not all pages can be edited when accessed from a link in the navigation tree. See the online help for the browser for more detail, including how to access versions of the page that can be edited.

When you click an item in the navigation tree pane, its corresponding collection or detail page opens in the content pane.



Service Integration Bus Browser
Buses
SCA.APPLICATION.localhostNode01Cell.Bus
Destinations
nlNode01.server1
nlNode01.server1-SCA.APPLICATION.l
Queue Points
Publication Points
Mediation Points
CommonEventInfrastructure_Bus
SCA.SYSTEM.localhostNode01Cell.Bus
Messaging Engine
Messaging Engine
?
Configuration

Messaging Engine

A messaging engine is a component, running inside a server, that manages messaging resources for a bus member. Applications are connected to a messaging engine when accessing a service integration bus.

Name
nlNode01.server1-SCA.APPLICATION.localhostNode01Cell.Bus

UUID
5BD189ACBA525767

Description

Initial state
Started

Message store type
File store

High message threshold
50000 messages

Target groups

Bus name
SCA.APPLICATION.localhostNode01Cell.Bus

Bus UUID
B577C7C2C43D03BD

Table 6 lists and describes the icons associated with each item in the navigation tree.

Table 6. Icons in the service integration bus browser

Icon	Description
📁	Indicates a collection of buses, destinations, queue points, publication points, or mediation points, depending on where in the navigation tree it appears.
🖧	Indicates a service integration bus.
⚙	Indicates a messaging engine.
🖥	Indicates a service integration bus member.

Related tasks

➜ Learning about service integration buses

Using commands to manage service applications

You can manage service applications using commands. The commands can be used within scripts.

Before you begin

Use the wsadmin tool to run service application commands.

About this task

You can use the wsadmin tool in different ways. You can use the tool interactively, as an individual command, or with a script. Running multiple commands in a script is useful if you are administering multiple machines.

WebSphere Process Server includes commands that display SCA modules and their imports and exports, and change the details of import and export bindings

Note: Jacl syntax for wsadmin scripts is deprecated and support will be removed in a future release. Jython is the strategic syntax for wsadmin scripts.

Procedure

1. List the SCA administration commands. `$AdminTask help SCAAdminCommands`
2. Display detailed help about a given command. `$AdminTask help command_name`

Example

```
$AdminTask help listSCAModules
```

Administering service modules using commands:

You can list the service modules that have been deployed to WebSphere Process Server from the command line. You can also view information associated with individual service modules and make changes to some import bindings.

Before you begin

Use the wsadmin tool to run WebSphere Process Server commands.

About this task

You can run commands individually or in a script. Running multiple commands in a script is useful if you are administering multiple hosts, or producing regular reports.

Listing service modules using commands:

You can use a command to list the service modules that have been deployed to WebSphere Process Server.

Before you begin

Start the wsadmin tool, which is used to run the required commands.

About this task

Perform the following steps to list all deployed service modules.

Procedure

List the deployed SCA modules. `$AdminTask listSCAModules`

Results

Lists the SCA modules that have been deployed to WebSphere Process Server, and the applications they are associated with. The output is returned in the format: `module name:application name`. This makes it easier for scripts to parse the output and extract names, for use in subsequent commands.

Displaying details of a service module using commands:

You can use a command to display attributes of service modules.

Before you begin

Start the wsadmin tool, which is used to run the required commands.

About this task

To show the description of a particular service module, you need to know the module name.

The listSCAModules command lists SCA modules that have been deployed to WebSphere Process Server.

Procedure

1. List the deployed SCA modules. `$AdminTask listSCAModules`
2. Display the details of a particular SCA module. `$AdminTask showSCAModule {-moduleName moduleName}`

   Note: In addition to specifying the module name, you have the option of specifying the application name. Providing an application name improves performance.

Results

The name and description of the specified SCA module is displayed.

Example

```
$AdminTask showSCAModule {-moduleName myModule -applicationName
myApplication}
```

Displaying the properties of a service module using commands:

You can use a command to show the properties for a specified service module.

Before you begin

Start the wsadmin tool, which is used to run the required commands.

About this task

To show the properties of a particular service module, you need to know the module name.

The listSCAModules command lists SCA modules that have been deployed to WebSphere Process Server.

Procedure

1. List the deployed SCA modules. `$AdminTask listSCAModules`
2. Display the properties of a particular SCA module. `$AdminTask showSCAModuleProperties {`**`-moduleName`** *`moduleName`*`}`

   Note: In addition to specifying the module name, you have the option of specifying the application name. Providing an application name improves performance.

Results

The properties of the specified SCA module are displayed.

Example

```
$AdminTask showSCAModuleProperties {-moduleName myModule -applicationName
myApplication}
```

Changing a service module property using commands:

You can use a command to change a property value for a specified service module.

Before you begin

Start the wsadmin tool, which is used to run the required commands.

About this task

The listSCAModules command lists all SCA modules that have been deployed to WebSphere Process Server.

Procedure

1. List the deployed SCA modules. `$AdminTask listSCAModules`
2. List the properties for a particular SCA module. `$AdminTask showSCAModuleProperties {`**`-moduleName`** *`moduleName`*`}`

   Note: In addition to specifying the module name, you have the option of specifying the application name. Providing an application name 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 application name. Providing an application name improves performance.

Results

The property value for the specified SCA module property is changed.

Example

```
$AdminTask modifySCAModuleProperty {-moduleName myModule -applicationName
myApplication -propertyName myPropertyName -newPropertyValue
myNewPropertyValue}
```

Listing imports using commands:

You can use a command to list the imports of any service module deployed to WebSphere Process Server.

Before you begin

Use the wsadmin tool to run the following commands.

About this task

Use the wsadmin tool to list the imports of a service module, by completing the following steps.

To list the imports of a service module, you need to know the name of the module.

The listSCAModules command lists all SCA modules that have been deployed to WebSphere Process Server, and the listSCAImports command lists the imports for a particular SCA module. It is possible for an SCA module not to have any imports.

Procedure

1. List the deployed SCA modules. `$AdminTask listSCAModules`
2. List the imports of a particular SCA module. `$AdminTask listSCAImports {-moduleName moduleName}`

   Note: In addition to specifying the *moduleName*, you have the option of specifying the *applicationName*. Providing an *applicationName* improves performance.

Results

Lists the imports for a particular SCA module.

Example

```
$AdminTask listSCAImports {-moduleName myModule -applicationName
myApplication}
```

Displaying details of an import using a command:

You can use a command to display import details of a service module deployed to WebSphere Process Server.

Before you begin

Use the wsadmin tool to run the following commands.

About this task

Use the wsadmin tool to display the import details of a service module by completing the following steps.

To show the details of a particular service module import, you need to know the module name and the import name.

The listSCAModules command lists all SCA modules that have been deployed to WebSphere Process Server, and the listSCAImports command lists the imports for a particular SCA module. It is possible for an SCA module not to have any imports.

Procedure

1. List the deployed SCA modules. `$AdminTask listSCAModules`
2. List the imports for a particular SCA module. `$AdminTask listSCAImports {-moduleName moduleName}`

   Note: In addition to specifying the *moduleName*, you have the option of specifying the *applicationName*. Providing an *applicationName* improves performance.
3. Show the details of a particular SCA module import. `$AdminTask showSCAImport {-moduleName moduleName -import importName}`

   In addition to specifying the *moduleName* and *importName*, you have the option of specifying the *applicationName*.

   Note: Providing an *applicationName* improves performance.

Results

Displays the import details for a particular SCA module import.

Example

```
$AdminTask showSCAImport {-moduleName myModule -applicationName
myApplication -import myImport}
```

Displaying an import binding using a command:

You can use a command to display the import bindings of a service module deployed to WebSphere Process Server.

Before you begin

Use the wsadmin tool to run the following commands.

About this task

Use the wsadmin tool to display the import bindings of a particular service module, by completing the following steps.

To show the import bindings of a particular service module import, you need to know the module name and the import name.

The listSCAModules command lists all SCA modules that have been deployed to WebSphere Process Server, and the listSCAImports command lists the imports for a particular SCA module. It is possible for an SCA module not to have any imports.

Procedure

1. List the deployed SCA modules. `$AdminTask listSCAModules`
2. List the imports for a particular SCA module. `$AdminTask listSCAImports {-moduleName moduleName}`

In addition to specifying the *moduleName*, you have the option of specifying the *applicationName*

Note: Providing an *applicationName* improves performance.

3. Show the import binding for a particular import. `$AdminTask showSCAImportBinding {`**`-moduleName`**` `*`moduleName`*` `**`-import`**` `*`importName`*`}`

   In addition to specifying the *moduleName* and *importName*, you have the option of specifying the *applicationName*.

   Note: Providing an *applicationName* improves performance.

Results

Displays the import binding for a particular SCA module import.

Example

```
$AdminTask showSCAImportBinding {-moduleName myModule -applicationName
myApplication -import myImport}
```

Changing an SCA import binding using commands:

You can use a command to change the SCA import bindings of service modules deployed to WebSphere Process Server.

Before you begin

Use the wsadmin tool to run the following commands.

About this task

An SCA binding connects one Service Component Architecture (SCA) module to another SCA module.

You might change an SCA import binding if you wanted a particular service module to call a different service 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 service module import, you need to know the names of the source and target service modules, and the specific import and export.

The listSCAModules command lists all SCA modules that have been deployed to WebSphere Process Server. The listSCAImports command lists all imports for a particular SCA module and the listSCAExports command lists all exports for a particular SCA module.

Procedure

1. List the deployed SCA modules. `$AdminTask listSCAModules`
2. List the imports for a particular SCA module. `$AdminTask listSCAImports {`**`-moduleName`**` `*`moduleName`*`}`

Note: In addition to specifying the *moduleName*, you have the option of specifying the *applicationName*. Providing an *applicationName* improves performance.

3. Modify an SCA module import binding, of type SCA. `$AdminTask modifySCAImportSCABinding` **`-moduleName`** *`moduleName`* **`-import`** *`importName`* **`-targetModule`** *`targetModuleName`* **`-targetExport`** *`targetExportName`*

   You also have the option of specifying the *applicationName* and *targetApplicationName*.

   Note: Providing an *applicationName* and a *targetApplicationName* improves performance.

Results

Changes the SCA import binding for a particular SCA module import.

WebSphere ESB issues a warning for each import interface that is not satisfied by an export interface. WebSphere Process Server compares the WSDL port type names of import and export, if they are not the same then a warning is issued. However, if the port type names do match, then WebSphere Process Server assumes that the operations provided are equivalent and no warning is issued.

Example

```
$AdminTask modifySCAImportSCABinding {-moduleName myModule -applicationName
myApplication -import myImport -targetModule myTargetModule
-targetApplicationName myTargetApplication -targetExport myTargetExport}
```

Changing an import Web service binding using commands:

You can use a command to change the Web service import bindings of service modules deployed to WebSphere Process Server.

Before you begin

Use the wsadmin tool to run the following commands.

About this task

You might change an import Web service binding if you wanted a particular service module to call a different Web service. If you change an import binding you must ensure that the import and export match (that is, that the operations provided are equivalent). This might involve reviewing the WSDL. Use the wsadmin tool to complete the following steps.

To modify the Web service binding of a particular service module import, you need to know the specific URL for the target endpoint.

The listSCAModules command lists all SCA modules that have been deployed to WebSphere Process Server. The listSCAImports command lists all imports for a particular SCA module and the listSCAExports command lists all exports for a particular SCA module.

Procedure

1. List the deployed SCA modules. `$AdminTask listSCAModules`

2. List the imports for a particular SCA module. `$AdminTask listSCAImports {`**`-moduleName`** *`moduleName`*`}`

   In addition to specifying the *moduleName*, you have the option of specifying the *applicationName*.

   Note: Providing an *applicationName* improves performance.
3. Modify an import binding, of type WS. `$AdminTask modifySCAImportWSBinding` **`-moduleName`** *`moduleName`* **`-import`** *`importName`* **`-endpoint`** *`targetEndpointName`*

Results

Changes the import Web service binding for a particular Web service import.

WebSphere ESB issues a warning for each import interface that is not satisfied by an export interface. WebSphere Process Server compares the WSDL port type names of import and export, if they are not the same then a warning is issued. However, if the port type names do match, then WebSphere Process Server assumes that the operations provided are equivalent and no warning is issued.

Example

```
$AdminTask modifySCAImportWSBinding {-moduleName myModule -applicationName
myApplication -import myImport -endpoint http://myTargetEndpoint}
```

Listing exports using commands:

You can use a command to list the exports of any service module deployed to WebSphere Process Server.

Before you begin

Use the wsadmin tool to run the following commands.

About this task

Use the wsadmin tool to list the exports of a service module by completing the following steps.

To list the exports of a particular service module, you need to know the module name.

The listSCAModules command lists all SCA modules that have been deployed to WebSphere Process Server, and the listSCAExports command lists the exports for a particular SCA module. It is possible for an SCA module not to have any exports.

Procedure

1. List the deployed SCA modules. `$AdminTask listSCAModules`
2. List the exports of a particular SCA module. `$AdminTask listSCAExports {`**`-moduleName`** *`moduleName`*`}`

   In addition to specifying the *moduleName*, you have the option of specifying the *applicationName*.

   Note: Providing an *applicationName* improves performance.

Results

Lists the exports for a particular SCA module.

Example

```
$AdminTask listSCAExports {-moduleName myModule -applicationName
myApplication}
```

Displaying details of an export using commands:

You can use a command to display export details of a service module deployed to WebSphere Process Server.

Before you begin

Use the wsadmin tool to run the following commands.

About this task

Use the wsadmin tool to display the export details of a service module, by completing the following steps.

To show the details of a particular service module export, you need to know the module name and the export name.

The listSCAModules command lists all SCA modules that have been deployed to WebSphere Process Server, and the listSCAExports command lists the exports for a particular SCA module. It is possible for an SCA module not to have any exports.

Procedure

1. List the deployed SCA modules. `$AdminTask listSCAModules`
2. List the exports for a particular SCA module. `$AdminTask listSCAExports {-moduleName moduleName}`

   Note: In addition to specifying the *moduleName*, you have the option of specifying the *applicationName*. Providing an *applicationName* improves performance.
3. Show the details of a particular SCA module export. `showSCAExport -moduleName moduleName -export exportName`

   In addition to specifying the *moduleName* and *exportName*, you have the option of specifying the *applicationName*.

   Note: Providing an *applicationName* improves performance.

Results

Displays the export details for a particular SCA module export.

Example

```
$AdminTask showSCAExport {-moduleName myModule -applicationName
myApplication -export myExport}
```

Displaying an export binding using commands:

You can use a command to display the export bindings of a service module deployed to WebSphere Process Server.

Before you begin

Use the wsadmin tool to run the following commands.

About this task

To display the export bindings of a service module, use the wsadmin tool to complete the following steps.

In order to show the export bindings of a particular service module export, you need to know the module name and the export name.

The listSCAModules command lists all SCA modules that have been deployed to WebSphere Process Server, and the listSCAExports command lists the exports for a particular SCA module.

Procedure

1. List the deployed SCA modules. `$AdminTask listSCAModules`
2. List the exports for a particular SCA module. `$AdminTask listSCAExports {-moduleName moduleName}`

   In addition to specifying the *moduleName*, you have the option of specifying the *applicationName*.

   Note: Providing an *applicationName* improves performance.
3. Show the export binding for a particular export. `$AdminTask showSCAExportBinding {-moduleName moduleName -export exportName}`

   Note: In addition to specifying the *moduleName* and *export*, you have the option of specifying the *applicationName*. Providing an *applicationName* improves performance.

Results

Shows the export binding for a particular SCA module export The information displayed depends upon the type of binding. If an export has no binding specified then the runtime assumes that the binding is of type SCA.

Example

```
$AdminTask showSCAExportBinding {-moduleName myModule -applicationName
myApplication -export myExport}
```

Changing WebSphere MQ JMS destinations of deployed SCA modules

As your business environment evolves, it is sometimes necessary to change which WebSphere MQ destinations a Service Component Architecture (SCA) module uses.

Before you begin

You must know which SCA modules use the WebSphere MQ JMS destinations you are changing. By default, these destinations are created when you install an SCA module into a server or cluster. The destinations are of the form:

- For a one-way export:
  - *modulename.exportname*_**MQEXPORT_CF**
  - *modulename.exportname*_**MQ_RECEIVE_D**
- In addition, for a two-way export:
  - *modulename.exportname*_**MQ_SEND_D**
- For a one-way import:
  - *modulename.importname*_**MQIMPORT_CF**
  - *modulename.importname*_**MQ_SEND_D**
- In addition, for a two-way import:
  - *modulename.importname*_**MQ_RECEIVE_D**

This task assumes that you are using the administrative console to change the configuration.

Restrictions: When changing the destinations keep the following in mind:

- You must configure queue destinations with the target client set to **MQ**.
- You must use a TCP/IP client connection to connect to WebSphere MQ. See "WebSphere MQ Intercommunication" for details.
- You cannot use channel compression.
- You must set any necessary channel exits for correct data handling: click **WebSphere MQ messaging provider** → **WebSphere MQ connection factory objects** → **Custom Properties** and set the following custom properties:
  - **SENDEXIT** to the value com.ibm.ws.sca.internal.mq.exit.MQInternalSendExitImpl.
  - **RECEXIT** to the value com.ibm.ws.sca.internal.mq.exit.MQInternalReceiveExitImpl.
  - Optional: **SENDEXITINIT** to a text string of your choice. Setting this property helps you identify the exit in a trace.
  - Optional: **RECEXITINIT** to a text string of your choice. Setting this property helps you identify the exit in a trace.
- You cannot use the same connection factories for WebSphere MQ and Java Message Service (JMS) MQ bindings as the exits will inhibit the JMS MQ bindings.

About this task

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

Procedure

1. Stop all applications that use the destinations you are changing. Use the steps described in "Administering enterprise applications."
2. Allow the destination to complete processing the messages in progress.
3. Display the WebSphere MQ JMS destinations. Navigate to this page in the administrative console using **Resources > JMS Providers > WebSphere MQ > WebSphere MQ queue destinations**
4. Select the scope of the destinations you are changing.

   If the SCA modules containing the destinations are installed on single servers, select **Server scope**.

If the SCA modules containing the destinations are installed on clusters, select **Cluster scope**.

5. Select the destination to change from the list.
6. Change the fields on the next page to the new values.
7. Repeat steps 5 and 6 for each destination you are changing.
8. Save the configuration changes.

What to do next

Restart the applications you stopped in step 1 on page 98.

Service Component Architecture modules and WebSphere MQ

SCA modules and WebSphere MQ queues can be connected to provide services to one another.

Service Component Architecture (SCA) modules can communicate with WebSphere MQ applications much in the same way as they do other SCA modules. A module that wants to send requests to a WebSphere MQ application uses an import configured with the correct response and request queues associated with that application. Similarly, an SCA module can provide services to a WebSphere MQ application using an export configured with the appropriate application request and response queues. You define the connections between the SCA modules and the WebSphere MQ queues when you build your module.

From the WebSphere MQ queue manager perspective, the SCA module looks as if it were a normal MQ client. From the SCA module end, the WebSphere MQ queue looks like any other service. You can even further shield the SCA module from the WebSphere MQ queues by using a mediation module between the SCA module and the WebSphere MQ queue and let the mediation transform the original SCA request to the correct format for the target queue and handle the response when it becomes available.

Restrictions: When configuring WebSphere MQ for imports and exports, keep in mind the following:

- You must configure queue destinations with the target client set to **MQ**.
- You must use a TCP/IP client connection to connect to WebSphere MQ. See “WebSphere MQ Intercommunication” for details.
- You cannot use channel compression.
- You must set any necessary channel exits for correct data handling: click **WebSphere MQ messaging provider** → **WebSphere MQ connection factory objects** → **Custom Properties** and set the following custom properties:
  - **SENDEXIT** to the value com.ibm.ws.sca.internal.mq.exit.MQInternalSendExitImpl.
  - **RECEXIT** to the value com.ibm.ws.sca.internal.mq.exit.MQInternalReceiveExitImpl.
  - Optional: **SENDEXITINIT** to a text string of your choice. Setting this property helps you identify the exit in a trace.
  - Optional: **RECEXITINIT** to a text string of your choice. Setting this property helps you identify the exit in a trace.
- You cannot use the same connection factories for WebSphere MQ and Java Message Service (JMS) MQ bindings as the exits will inhibit the JMS MQ bindings.

Working with targets

Targets provide additional flexibility by allowing you to modify processing by changing the target configured for a reference.

A component can call a component in another module to minimize the time and cost of building an application. Targets provide additional flexibility: to allow your installed applications to benefit from advances in processing or other changes, you can use the administrative console to change the endpoint of a cross-module invocation, without rewriting or redeploying the application.

To take advantage of the flexibility provided, you must understand how the system names the targets. The link from the calling module must connect to the correct target.

Target names

Target names are derived from how the calling component invokes the target. The names have the following format:

Invocation type
Name format

Synchronous
A name that follows the Java Naming and Directory Interface (JNDI) format, for example:

folder/export/*fullpath_to_target*/*target_component_name*

Asynchronous
A name with the format

folder/*calling_component_name*/
full_path_to_target_component/*target_component_name*

Multiple destinations
This name is the same as an asynchronous invocation but the target sends a message to multiple destination components.

Related tasks

"Changing import 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 import targets

Changing the target of a reference provides applications with the flexibility of taking advantage of advances in components as they happen without recompiling and reinstalling the application.

Before you begin

Before changing the target for a reference you must:

- Make sure the new target uses the same data object type
- Know whether the module is synchronously or asynchronously invoking the target
- Know whether the reference targets a single or multiple services

About this task

Change the target of an import from a module when another service with the same interface as the original target provides new or improved functionality that your module can use.

Procedure

1. Stop the module that contains the reference that you are changing.
   a. Using the administrative console, display the Service Component Architecture (SCA) modules.

      Navigate to this panel using **Applications > SCA Modules**

   b. Select your module and press **Stop**. The display updates to show the application as stopped.
2. Change the target destination of the reference.

   How you make the change depends on how the module invokes the target.

Type of invocation	How to change
Single target service	1. Using the administrative console, display the SCA Modules. Navigate to the panel using **Applications > SCA Modules**. 2. From the displayed list, select the module that contains the import that references the target to change. 3. Expand the list of imports by clicking the plus sign (+) next to **Imports**. 4. Select the import to change from the list. 5. In the **Target** area, select the **Module** from the list. 6. After the **Export** list refreshes, select the export for the new target. 7. Save the change by clicking **OK**.
Multiple target services	1. Display the buses on the system on which the module resides. Navigate to the panel using **Service Integration > Buses**. 2. Select the **SCA.System.*cellname*.Bus** 3. Display the destination targets for the bus by clicking **Destinations**. 4. Select the destination that represents the import that connects the calling module to the targets. This identifier will contain the word import. 5. Display the list of properties by clicking **Context properties**. 6. Select the property to change by clicking on the targets property in the list. 7. Change the **Context value** field to the new destination targets. 8. Return to the **Context properties** panel by clicking **OK**. 9. Save the change by clicking **OK**.

3. Save your changes. Click **Save** when prompted.

What to do next

Start the module and make sure the module receives the expected results.

Deleting J2C activation specifications

The system builds J2C application specifications when installing an application that contains services. There are occasions when you must delete these specifications before reinstalling the application.

Before you begin

If you are deleting the specification because of a failed application installation, make sure the module in the Java Naming and Directory Interface (JNDI) name matches the name of the module that failed to install. The second part of the JNDI name is the name of the module that implemented the destination. For example in sca/SimpleBOCrsmA/ActivationSpec, **SimpleBOCrsmA** is the module name.

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

About this task

Delete J2C activation specifications when you inadvertently saved a configuration after installing an application that contains services and do not require the specifications.

Procedure

1. Locate the activation specification to delete.

   The specifications are contained in the resource adapter panel. Navigate to this panel by clicking **Resources > Resource adapters**.

   a. Locate the **Platform Messaging Component SPI Resource Adapter.**

      To locate this adapter, you must be at the **node** scope for a standalone server or at the **server** scope in a deployment environment.

2. Display the J2C activation specifications associated with the Platform Messaging Component SPI Resource Adapter.

   Click on the resource adapter name and the next panel displays the associated specifications.

3. Delete all of the specifications with a **JNDI Name** that matches the module name that you are deleting.

   a. Click the check box next to the appropriate specifications.

   b. Click **Delete.**

Results

The system removes selected specifications from the display.

What to do next

Save the changes.

Deleting SIBus destinations

Service integration bus (SIBus) destinations are used to hold messages being processed by SCA modules. If a problem occurs, you might have to remove bus destinations to resolve the problem.

Before you begin

If you are deleting the destination because of a failed application installation, make sure the module in the destination name matches the name of the module that failed to install. The second part of the destination is the name of the module that implemented the destination. For example in sca/SimpleBOCrsmA/component/test/sca/cros/simple/cust/Customer, **SimpleBOCrsmA** is the module name.

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

About this task

Delete SIBus destinations when you inadvertently saved a configuration after installing an application that contains services or you no longer need the destinations.

Note: This task deletes the destination from the SCA system bus only. You must remove the entries from the application bus also before reinstalling an application that contains services (see Deleting J2C activation specifications in the Administering section of this information center.

Procedure

1. Log into the administrative console.
2. Display the destinations on the SCA system bus.
   a. In the navigation pane, click **Service integration** → **buses**
   b. In the content pane, click **SCA.SYSTEM.*cell_name*.Bus**
   c. Under Destination resources, click **Destinations**
3. Select the check box next to each destination with a module name that matches the module that you are removing.
4. Click **Delete.**

Results

The panel displays only the remaining destinations.

What to do next

Delete the J2C activation specifications related to the module that created these destinations.

Administering enterprise applications

Use the console's Enterprise Application page (viewed by clicking **Applications > Enterprise Applications**) to view and administer enterprise applications installed on the server.

To view the values specified for an application's configuration, click the application name from the list. The application details page opens and displays the application's configuration properties and, if appropriate, local topology. From this page, you can modify existing values and link to additional console pages for configuring the application.

To administer an enterprise application, select it by clicking the check box next to its name and then use one of the following buttons:

Table 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.
Export DDL	Opens the Export Application DDL files page so you can export DDL files in the EJB modules of an enterprise application.

For more information on administering applications, see the WebSphere Application Server Information Center.

Administering the Application Scheduler

Application Scheduler allows an administrator to schedule the starting and stopping of applications that are installed on WebSphere Process Server. Use the Application Scheduler panel in the administrative console to control the scheduling of any installed application.

Additionally, you can generate scheduler entries during the migration of a WebSphere InterChange Server repository that includes WebSphere InterChange Server scheduler entries. (See the topics on Migrating from WebSphere InterChange Server and the reposMigrate command). Use the Application Scheduler panel in the administrative console to administer these migrated scheduler entries as well.

In a stand-alone server environment, the Application Scheduler is automatically installed. When you create the stand-alone server profile, the Application Scheduler is installed and configured on that server.

In a Network Deployment environment, the Application Scheduler is automatically installed for every managed server and cluster member created; no additional action is needed.

In WebSphere InterChange Server, an application that contained collaboration objects or connectors could be started, paused, and stopped at the component level (that is, a component could be stopped while the remainder of the application was allowed to continue). In WebSphere Process Server, scheduling of events is provided through the Application Scheduler. The Application Scheduler allows you to start and stop processes at the application level.

Accessing the Application Scheduler

Access the Application Scheduler either programmatically using the Application Scheduler MBean interface or through the Application Scheduler panels of the administrative console.

Accessing the Application Scheduler using the Application Scheduler MBean interface

Use the command line to invoke the Application Scheduler MBean.

About this task

Perform the following to invoke Application Scheduler MBean.

Procedure

1. Set the properties SOAP_HOSTNAME and SOAP_PORT in the class com.ibm.wbiserver.migration.ics.Parameters.

   This class is in the migration-wbi-ics.jar file in the *WAS_HOME*\lib directory. `SOAP_HOSTNAME` is the name of the host where Application Scheduler is running. `SOAP_PORT` is the port where the Application Scheduler is running.

   ```
   Parameters.instance.setProperty(Parameters.SOAP_HOSTNAME, "localhost");
   Parameters.instance.setProperty(Parameters.SOAP_PORT, "8880");
   ```

   Note: If security is turned on, you must specify a user ID and password in the soap properties file found at the location *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, which invokes the correct Mbean based on its `name`, `type`, `server name` and `node name`.

```
protected static final String WEBSPHERE_MB_QUERY_CONSTANT = "WebSphere:*";
protected static final String NAME_QUERY_CONSTANT = ",name=";
protected static final String WBI_SCHED_MB_NAME = "Scheduler_AppScheduler";
protected static final String TYPE_QUERY_CONSTANT = ",type=";
protected static final String WBI_SCHED_MB_TYPE = "WASScheduler";
protected static final String SERVER_QUERY_CONSTANT = ",process=";
serverName = "<server1>";
protected static final String NODE_QUERY_CONSTANT = ",node=";
nodeName = "<myNode>";

String queryString = new StringBuffer(WEBSPHERE_MB_QUERY_CONSTANT)
   .append(NAME_QUERY_CONSTANT)
   .append(WBI_SCHED_MB_NAME)
   .append(TYPE_QUERY_CONSTANT)
   .append(WBI_SCHED_MB_TYPE)
   .append(SERVER_QUERY_CONSTANT)
   .append(serverName)
   .append(NODE_QUERY_CONSTANT)
   .append(nodeName).toString();

MBeanUtil mbs = new MBeanUtil(queryString.toString());
```

3. Call Mbean methods using the invoke() method of the MbeanUtil instance and pass it the name of the method.

Example

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, such as `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 createSchedulerEntry method of the Mbean. Pass it the scheduler entry `entry1` as a parameter along with the name of the ScheduleEntry class.

```
mbs.invoke(schedulerExtMBName, "createScheduleEntry", new Object[]{entry1},
     new String[]{"com.ibm.wbiserver.scheduler.common.ScheduleEntry"});
```

Finally, read all the Schedule entries, including the one that was just added, by calling the readAllScheduleEntries method.

```
     result = mbs.invoke("readAllScheduleEntries", null, null);
 }
 catch (MigrationException e)
 {
     e.printStackTrace();
 }
```

Displaying scheduler entries using the administrative console

Use the Application Scheduler panel of the administrative console to create, modify, or delete scheduler events.

Before you begin

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

Procedure

1. Select **Servers** → **Application Servers** → *ServerName*.
2. Select **Application Scheduler** under the **Business Integration** subheading.
3. Select the scope (cell, node, server) of the entries to display.
   Scheduler entries are normally defined at the server scope.

Results

The existing scheduled events for that scope are listed.

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

Creating a scheduled event

The administrative console provides a panel for creating new scheduled events.

Before you begin

To create a new scheduled event, you must be at the Application Scheduler collection panel in the administrative console for the server.

About this task

You might need to create an event to fit a specific need. To create a new scheduled event, follow these steps.

Note: The fields with an "*" on the panel are required fields.

Procedure

1. Click **New**. The Add panel opens.
2. Configure the scheduled event.
   a. Select the **Group Application**.

b. Select the **Status**.

c. Type in the **Initial Date** with the following format: *Abbrv month, dd, yyyy*. For example, type **Apr 15, 2005** for April 15, 2005.

d. Type in the **Initial Time** using a 12-hour format (*hh:mm*), and then type either `am` or `pm` and the time zone.

   Note: After you have moved from this field, the **Next Fire Time** is automatically calculated.

e. Select the **Action**.

f. Optional: Fill in the **Recurrence** parameters.

   - **Start-by period**

     If the Application Scheduler or Process Server is not running at the time an event is scheduled to fire, the start-by period parameter defines a length of time or window (in minutes) commencing at the scheduled firing time of the event, during which an event will fire if the Application Scheduler or process server resumes operation. However, if the Application Scheduler or process server does not resume operation until after the Start-by-period has expired, the next fire time is calculated and the event will fire at that time.

     For example, suppose you set the start-by period to 60 - Minutes for an event that is scheduled to fire at midnight but the server happens to be down at that time. Provided that the server comes back online before 1:00 a.m., the event will fire.

   - Whether the scheduled entry should recur at a specified time.
     - One or more times a minute, hour, day, month, or year.
     - A certain day (Sunday through Saturday) of a certain week (first, second, third, fourth, or last) of every one or more months.
     - The last day of every one or more months.

3. Click **Apply** or **OK** to set the event.

   Note: To create another event, click **Reset** to clear the panel.

Results

Application Scheduler creates and displays a new scheduled event in the Application Scheduler panel.

Event status and action descriptions

Each event must have a status and an action.

Status

The **Status** field shows the 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.

Status	Description
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 that either the task has been purged or the information used to query for that task is invalid.
Running	A task is in the midst of firing. **Note:** This status should be seen rarely because 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 to do with the event. There are only two actions available for an event:

- **Start Application** - starts all applications that are under the system deployment manager.
- **Stop Application** - stops all applications that are under the system deployment manager.

Modifying a scheduled event

Modify migrated or existing scheduled events from the administrative console.

Before you begin

To modify a scheduled event, you must be at the Application Scheduler collection panel in the administrative console for the server.

Procedure

1. Click the **Schedule Entry Id** of the event that you want to modify. The Event panel opens.
2. Modify any of the following fields:

   Note: Because all applications on the server are listed, you must be careful when changing the status of an existing event. You may stop an application that is running on the server.

   - **Group Application**
   - **Status**
   - **Initial Date** with the following format (*Abbrv month, dd, yyyy*)
   - **Initial Time** using a 12-hour format (*hh:mm*)
   - **Action**

   Optional: You can also fill in the **Recurrence** parameters.
3. Click **Apply** or **OK** to set the modifications for the event.

Note: If you modify a scheduled event, the server assigns a new Schedule Entry ID. The server deletes the currently scheduled event and schedules a new event with the new ID.

Results

The panel displays the modified event with the new ID in the Application Scheduler collection panel.

Deleting a scheduled event

Application Scheduler provides a panel for deleting scheduled events.

Before you begin

To delete a scheduled event, you must be at the Application Scheduler collection panel in the administrative console for the server.

About this task

As events become obsolete, you can delete them from the list of events in the collection panel. Follow these steps to delete a scheduled event.

Procedure

1. In the **Select** column, select the Schedule Entry to be deleted.
2. Click **Delete**.

Results

The Schedule Entry is deleted.

Administering relationships

The relationship manager is a tool for manually controlling and manipulating relationship data to correct errors found in automated relationship management or provide more complete relationship information. In particular, it provides a facility for retrieving as well as modifying relationship instance data.

How the relationship manager works

The relationship manager allows you to configure, query, view, and perform operations on relationship runtime data, including roles and their data. You create relationship definitions with the relationship editor. At run time, instances of the relationships are populated with the data that associates information from different applications. This relationship instance data is created when the maps or other WebSphere Process Server components run and need a relationship instance. The relationship service exposes a set of application programming interfaces (API's) to retrieve relationship metadata and to create, retrieve, and manipulate the instance data. The data is stored in the relationship tables that are specified in the relationship definition. The relationship manager provides a graphical user interface to interact with the relationships and relationship instances.

For each relationship instance, the relationship manager can display a hierarchical listing of its roles. Each role in the relationship has instance data, properties, and key attributes. The relationship tree also provides detailed information about each of the roles in the relationship instance, such as the type of entity, its value, and the date it was last modified. A relationship instance ID is automatically generated when the relationship instance is saved in the relationship table. The relationship manager displays this instance ID at the top level of the relationship tree.

Uses of the relationship manager

You can use the relationship manager to manage entities at all levels: the relationship instance, role instance, and attribute data and property data levels. For example, you can use the relationship manager to:

- Browse and inspect the values for existing relationships
- Create and delete relationship instances
- Modify the contents of a relationship instance, such as adding and deleting role instances
- Edit the data of a relationship role instance like role properties and logical state
- Activate and deactivate role instances
- Get role instances, given the key attribute, start and end date, and property value
- Salvage a situation when problems arise. For example, when corrupt or inconsistent data from a source application has been sent to the generic and destination application relationship table, you can use the relationship manager to rollback the data to a point in time when you know the data is reliable

For more information on relationships, see the WebSphere Integration Developer Information Center and the topics on the relationship service in the WebSphere Process Server Information Center.

Viewing relationships

Perform this task to view a list of relationships in the system, including the relationship name, display name, and static and identity attributes.

Before you begin

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

About this task

To view the list of relationships in the system, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.

   The information is displayed in table format. Each relationship type is a link.

   Tip: You can customize the number of rows to display at one time. Click **Preferences** and modify the **Maximum row** field value. The default is 25.

Viewing relationship details

Perform this task to view detailed information for the selected relationship, including the relationship name, display name, associated roles with their attributes, property values, and static and identity attributes.

Before you begin

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

About this task

To view detailed information for the selected relationship, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.
4. You can view the relationship details in two ways:
   a. Click the relationship name.
   b. Select the radio button next to the relationship name and click **Details**.

   The relationship details include role attributes, which are displayed in table format and include the display name, object name, and managed attribute setting for the role.

   To return to the list of relationships, click **Relationships** from the path at the top of the page or click **Back**.

Viewing role details

Perform this task to view detailed information for the selected role, including the relationship name, role name, display name, property values, keys, role object type, and managed attribute setting.

Before you begin

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

About this task

To view detailed information for the selected role, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the Relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.

4. Click a relationship name to open the Relationship Detail page.
5. Under **Role schema information**, click an associated role name to open the Role Detail page.

What to do next

To return to the Relationship Detail page, click **Relationship Detail** from the path at the top of the page or click **Back**.

Querying relationships

Use this task to perform relationship-based instance queries.

Before you begin

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

About this task

Select a query option (**All**, **By ID**, **,** **By property**, or **By role**) to get all or a subset of the instance data for a relationship. The return is the result set of that query and is displayed in table format with each row representing one relationship instance.

To query relationships, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.
4. Select the radio button next to the relationship name and click **Query**.
5. Click one of the query option tabs and specify the search criteria.

Option	Description
All tab	Get a list of all instances in the relationship. You can select to display all activated, all inactivated, or all activated and inactivated relationship instance data.
By ID tab	Get relationship instances in the range of the starting and ending instance identifiers. If you leave one field blank, the query returns only the single instance. The query returns all of the roles for the instances it finds.
By property tab	Get relationship instances by specific property values.
By role tab	Get relationship instances based on a role name, key attribute value, date range during which the role was created or modified, or specific property value.

6. After you have specified the query parameters, you have the following options:

- Click **OK** to display the result data from the query.
- Click **Cancel** to discard any changes made and return to the list of relationships.

Querying relationship data using database views

You can use views in a database to query relationship data without using the relationship manager.

You can use your database views to directly query relationship data stored on the database. When you create a new relationship database table, a corresponding SQL view is automatically created. These views are essentially encapsulations of the relationship data stored in database tables. You can use these views to populate, query relationship data, or both by:

- using SQL statements with a DB client (for example, with the DB2® command center)
- using JDBC to run SQL statements with a Java program

In either case, you can use the SQL views in the same manner as you would for tables. You can use this technique as an alternative method to the Relationship Manager application to directly populate large sets of application-specific data by using SQL statements into your relationship database(s). You can also use this technique to import data from a flat-text file into a database table

Relationship database SQL views are created based on data contained in tables located elsewhere in the data source. The view will exist even when the database table itself is empty. Each view is has its own unique name which follows this convention: ″V_″+*relationship_display_name*+″_″*role_display_name*+″_″+*uuid* (notice that the variables are concatenated using an underscore character ″_″). Both display names are limited to 20 alphanumeric characters, while the uuid is a number generated from the combination of both display names. Consequently, each view name should be unique within a data source. An example of this naming convention can be shown if we use these variables:

- *relationship_display_name* = SAMPLECUSTID
- *role_display_name* = MYCUSTOMER
- *uuid* = 80C (this number is generated automatically by the server)

The resulting view name would be ″V_SAMPLECUSTID_MYCUSTOMER_80C″. For a given relationship, you should have two corresponding views containing the same relationship display name but different role display names and uuids.

Note: For Oracle databases, the naming convention differs in this regard: only the first ten characters of the *relationship_display_name* and *role_display_name* are used.

Each view will contain the columns (including the associated properties of type, value, and nullable) listed in the following table:

Table 8. Relationship database view columns

Name	Data type	Value	Nullable?
INSTANCEID	Integer	The ID number used to correlate instance data between different applications.	No

Table 8. Relationship database view columns (continued)

Name	Data type	Value	Nullable?
ROLE_ATTRIBUTE_COLUMNS • Dynamic relationship - defined in business object • Static relationship - DATA	• Dynamic relationship - defined in business object • Static relationship - Varchar	The column name and type depends on the role definition. Column names are based on the key attribute names, while. column types are database data types that is mapped based on key attribute type defined in role definition.	No
STATUS	Integer	0-4 0 – created 1 – updated 2 – deleted 3 – activated 4 – deactivated **Note:** When populating instances through views, ensure that the value for this column is 0.	Yes
LOGICAL_STATE	Integer	• 0 = activated • 1 = deactivated Ensure that you set the proper value when you populate the database with data.	No
LOGICAL_STATE_TIMESTAMP	Timestamp	Date and time when the logical state column data was last updated.	Yes
CREATE_TIMESTAMP	Timestamp	Date and time when the role instance was created.	Yes
UPDATE_TIMESTAMP	Timestamp	Date and time when the role instance was last updated.	Yes
ROLEID	Integer	ID number used to identify a role instance	No

Example

This example presented here is an identity relationship that includes three sets of data from three enterprise applications:

- Clarify
- SAP
- Siebel

The data is correlated using the WebSphere Process Server relationship service. Each application contains similar customer information, with an identity relationship to correlate the information between each application.

The following three tables show the data as it is stored within each database:

Table 9. Clarify customer

Given Name	Last Name	Home Phone	ID
Jessica	Reed	111 111 11111	clarify_1
Tara	McLean	333 333 33333	clarify_2

Table 10. SAP customer

First Name	Last Name	Home Phone	ID
Jessica	Reed	111 111 11111	sap_10
Tara	McLean	333 333 33333	sap_8

Table 11. Siebel customer

Full Name	Home Phone	ID
Jessica Reed	111 111 11111	siebel_6
Tara McLean	333 333 33333	siebel_8

The customer business object definition names and elements (created in WebSphere Integration Developer for each database) are shown in the following table:

Table 12. Business object definitions for customer on each database

ClarifyCustomer		SapCustomer		SiebelCustomer	
Element	Type	Element	Type	Element	Type
`givenName`	`string`	`firstName`	`string`	`fullName`	`string`
`lastName`	`string`	`lastName`	`string`		
`homePhone`	`string`	`homePhone`	`string`	`homePhone`	`string`
`clarifyId`	`string`	`sapId`	`string`	`siebelId`	`string`

An identity relationship is defined to correlate the customer information between each database. This relationship, called `ID` in this example, uses the business object elements `clarifyId`, `sapId`, and `siebelId`. These elements are used because they contain the ID data for each database, and that data is unique for each customer. The following table describes the roles that are used to correlate different databases in the relationship to a common ID used by WebSphere Process Server:

Table 13. ID relationship definition

Relationship name	Role name	Business object name	Key
ID	GenCustomer	GenCustomer	genId
	ClarifyCustomer	ClarifyCustomer	clarifyId
	SapCustomer	SapCustomer	sapId
	SiebelCustomer	SiebelCustomer	siebelId

The full relationship name is `http://CustomerModule/ID`. The full role names are

- `http://CustomerModule/ClarifyCustomer`
- `http://CustomerModule/SapCustomer`
- `http://CustomerModule/SiebelCustomer`

You can correlate the data within the business objects contained in all three databases by using the defined relationship. The customer ID data from each database is correlated with the customer data from the other databases by sharing instance IDs. For example, Tara McLean is identified by `clarify_3 ID` in Clarify, `sap_8` in SAP, and `siebel_8` in Siebel. A unique ID is generated by the WebSphere Process Server relationship service.

Note: You cannot manipulate relationship instance tables using the views with the Derby database. You can, however, use the views to browse the relationship table content.

You can define multiple relationship instances by using the views created in the Common database. The mapping of the view name (using the naming convention as previously described) to its corresponding relationship role is captured in the `RELN_VIEW_META_T` table in the Common database. The following table shows an example of the view names for the `ClarifyCustomer`, `SapCustomer`, and `SiebelCustomer` roles:

Table 14. RELN_VIEW_META_T table

VIEW_NAME	RELATIONSHIP_NAME	ROLE_NAME
V_ID_CLARIFYCUSTOMER_098	http://CustomerModule/ID	http://CustomerModule/ClarifyCustomer
V_ID_SAPCUSTOMER_515	http://CustomerModule/ID	http://CustomerModule/SapCustomer
V_ID_SIEBELCUSTOMER_411	http://CustomerModule/ID	http://CustomerModule/SiebelCustomer
V_USASTATE_ABBREVIATION_DE8	http://CustomerModule/USASTATE	http://CustomerModule/Abbreviation
V_USASTATE_CODE_B32	http://CustomerModule/USASTATE	http://CustomerModule/Code
V_USASTATE_NAME_933	http://CustomerModule/USASTATE	http://CustomerModule/FullName

The view column definition as described in table 1 will have a `ROLE_ATTRIBUTE_COLUMN` with the following properties:

Table 15. View column definition

Column Name	Data Type	Value	Description
KEY_ATTRIBUTE_NAME	depends on the key attribute type	Not null	This is where the role instance data is stored. For identity relationships, the column is named by the name of the key attribute. For example, SAPCUSTOMER_SAPID will use sapid as the key attribute name and sapcustomer as the business object name. One column is defined for each key attribute. For static relationships, the column is named DATA

The following table shows the show the views in the Common database for the ID relationships.

Table 16. View column definition

Clarify role view	SAP role view	Siebel role view
INSTANCEID	INSTANCEID	INSTANCEID
CLARIFYCUSTOMER_CLARIFYID	SAPCUSTOMER_SAPID	SIEBELCUSTOMER_SIEBELID
STATUS	STATUS	STATUS
LOGICAL_STATE	LOGICAL_STATE	LOGICAL_STATE
LOGICAL_STATE_TIMESTAMP	LOGICAL_STATE_TIMESTAMP	LOGICAL_STATE_TIMESTAMP
CREATE_TIMESTAMP	CREATE_TIMESTAMP	CREATE_TIMESTAMP
UPDATE_TIMESTAMP	UPDATE_TIMESTAMP	UPDATE_TIMESTAMP
ROLEID	ROLEID	ROLEID

Note: All of the column names in the views match, except the key attribute column names.

You must first know the name of the role runtime table view before you can run SQL against the view to manipulate role instance data. The following SQL script shows an example using DB2® Universal Database™. The example assumes that all the data from each database has been copied to the relationship database. You can copy the data using the SELECT INTO SQL statement:

```
//Create a table to store ID values from all three applications for each customer,
//and associate a unique instance ID with each customer. Use this table as a base
//source table to populate relationship tables.
CREATE TABLE joint_t (instanceid INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY,
clarify_id VARCHAR(10) NOT NULL,
sap_id VARCHAR(10) NOT NULL,
siebel_id VARCHAR(10) NOT NULL)

//Compare the name and home phone number across the three application tables.
//If a match is found, insert that person's ID value from each application table
//into the joint_t table. Associate the three ID values to a unique ID; this
//ID will be used later as the relationship instance ID.
INSERT INTO joint_t (clarify_id,sap_id,siebel_id)
SELECT A.ID, B.ID, C.ID
FROM clarifycustomer A,sapcustomer B, siebelcustomer C
WHERE A.homephone=B.homephone AND
B.homephone=C.homephone, AND
B.givenname=C.firstname AND
B.lastname=C.lastname AND
```

```
A.fullname=C.firstname CONCAT ' ' CONCAT C.lastname

//Create a sequence for each application; this sequence will be
//used later as a role ID in each role table.
CREATE SEQUENCE clarify_roleid MINVALUE 1 ORDER CACHE 100
CREATE SEQUENCE sap_roleid MINVALUE 1 ORDER CACHE 100
CREATE SEQUENCE siebel_roleid MINVALUE 1 ORDER CACHE 100

//Populate the role instance table for the CLARIFY role.
INSERT INTO V_ID_CLARIFYCUSTOMER_098 (instanceid, roleid,
  clarifycustomer_clarifyid, status, logical_state, logical_state_timestamp,
  create_timestamp, update_timestamp)
FROM joint_t

//Populate the role instance table for the SAP role.
INSERT INTO V_ID_SAPCUSTOMER_515 (instanceid, roleid, sapcustomer_sapid,
  status, logical_state, logical_state_timestamp, create_timestamp,
  update_timestamp)
SELECT instanceid NEXTVAL FOR sap_roleid, sap_id, 0, 0, current
  timestamp, current timestamp, current timestamp
FROM joint_t

//Populate the role instance table for the SIEBEL role.
INSERT INTO V_ID_SIEBELCUSTOMER_AFC (instanceid, roleid, siebelcustomer_siebelid,
  status, logical_state, logical_state_timestamp, create_timestamp, update_timestamp)
SELECT instanceid, NEXTVAL FOR siebel_roleid, sap_id, 0, 0, current timestamp,
  current timestamp, current timestamp
FROM joint_t
```

The joint_t table is created to temporarily store key values. You can delete the table when you are finished to save resources, if necessary. Alternatively, you can create a view table or a temporary table.

Related concepts

Relationships
Relationships are services used to model and maintain associations between business objects and other data.

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.

Viewing relationship instances

Perform this task to view a list of relationship instances that match the relationship query. The results display in table view and include the relationship instance ID and the property values associated with the instance.

Before you begin

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

About this task

To view a list of relationship instances that match the relationship query, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.
4. Select the radio button next to the relationship name and click **Query**.
5. Click one of the query option tabs (**All**, **By ID**, **By property**, or **By role**) and specify the search criteria. For descriptions of the query options, see "Querying relationships" on page 113
6. Click **OK** to open the Relationship Instances page.

Results

The list of relationship instances that match your query appears in table view, with each relationship instance shown in its own row. The total page and returned instance counts are displayed at the bottom of the page.

Tip: You can customize the number of rows to display at one time. Click **Preferences**, modify the **row** field value, and click **Apply**. The default is 25, with 1 being the minimum number of records to display at one time and all records being the maximum.

You can navigate through the pages, as follows:

- To view the next set of instances, click the forward arrow.
- To view the previous page of instances, click the back arrow.

Restriction: Filtering or sorting on a large relationship instance count might result in performance problems as it requires getting the full query result set from the server in order to do the sorting. For example, sorting the relationship instance data on a query that would return 20,000 relationship instances needs to sort on those 20,000 instances. The total count (bottom of page) gives an estimate of how many relationship instances you can expect and whether sorting or filtering on a large set of data might lead to long wait times.

For information on setting the query block size parameter to allow for customization of how many instances are read from the server at one time, see the help topic on configuring the relationship service.

Viewing relationship instance details

Perform this task to view detailed information for the selected relationship instance, including the relationship name, relationship instance ID, property values, participating roles, and role instance values (role instance ID, logical state, key attributes, and property values). You can view multiple roles concurrently.

Before you begin

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

About this task

To view detailed information for the selected relationship instance, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.
4. Select the radio button next to the relationship name and click **Query**.
5. Click one of the query option tabs (**All**, **By ID**, **By property**, or **By role**); specify the search criteria; and click **OK** to open the Relationship Instances page.
6. You can view the relationship instance details in two ways:
   - Click the relationship instance ID.
   - Select the radio button next to the relationship instance ID and click **Details**.

   To return to the list of relationship instances, click **Relationships Instances** from the path at the top of the page.

Editing relationship instance details

Perform this task to edit the property values for the selected relationship instance.

Before you begin

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

About this task

To edit the property values for the selected relationship instance, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.
4. Select the radio button next to the relationship name and click **Query**.
5. Click one of the query option tabs (**All**, **By ID**, **By property**, or **By role**); specify the search criteria; and click **OK** to open the Relationship Instances page.
6. Display the relationship instance details in one of two ways:
   - Click the relationship instance ID.
   - Select the radio button next to the relationship instance ID and click **Details**.
7. Modify the relationship instance property values, as necessary.

   Restriction: You can only edit the property values if they have been previously defined for the relationship instance.
   To delete the relationship instance, click **Delete** at the bottom of the page.

   From this page, you can also create new role instances or delete existing role instances by selecting them and clicking **Create** or **Delete**, respectively, below the role table. Clicking **Create** will open the New Role Instance page for entering key attribute values and property values for the new role instance. You can edit the property values of the role instance by clicking the selected role instance ID.

8. When you are finished making changes in the instance and within the roles of the instance, you have the following options:
   - Click **OK** to save the changes to the system immediately.
   - Click **Cancel** to discard any changes and return to the Relationship Instances page.

Creating new relationship instances

Perform this task to create a new relationship instance.

Before you begin

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

About this task

To create a new relationship instance, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.
4. Select the radio button next to the relationship name and click **Create** to open the New Relationship Instance page.
5. Add the property value information in the **Value** field if you want values other than the default values, and click **OK** to save the new relationship instance locally.

   Note: You must also create a role instance for the relationship instance, as you cannot have a relationship instance without a role instance.

Deleting relationship instances

Perform this task to delete a selected relationship instance.

Before you begin

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

About this task

To delete a selected relationship instance, perform the following steps.

Procedure

1. Ensure that the administrative console is running.

2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.
4. Select the radio button next to the relationship name and click **Query**.
5. Click one of the query option tabs (**All**, **By ID**, **By property**, or **By role**); specify the search criteria; and click **OK** to open the Relationship Instances page.
6. Select the radio button next to the ID of the relationship instance you want to delete.
7. Click **Delete**.

   The relationship instance is deleted immediately from the system.

Rolling back relationship instance data

Perform this task to roll back the relationship instance data to a specified date and time.

Before you begin

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

About this task

The following actions are performed during the rollback:
- Relationship instances which are created during the given period get deleted (hard delete) from the database.
- Relationship instances which are activated get deleted (hard delete) from the database.
- Relationship instances which are deactivated in the given time period get activated.

To roll back the relationship instance data, perform the following steps.

Procedure
1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the relationships page for the server you want to manage by clicking **Relationships** next to the relationship services MBean.
4. Select the radio button next to the relationship name and click **Rollback**.
5. Enter the time period for the rollback in the **From date** and **To date** fields.

   Important: Make sure the WebSphere Process Server server and the database server are set to the same time zone or the rollback will fail.
6. Click **OK**.

   All instance data in the relationship created later than the specified date and time will be marked as deactivated.

Viewing role instance details

Perform this task to view detailed information for the selected role instance, including the role name, role element, key attributes and property values, status, and logical state.

Before you begin

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

About this task

To view detailed information for the selected role instance, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.
4. Select the radio button next to the relationship name and click **Query**.
5. Click one of the query option tabs (**All**, **By ID**, **By property**, or **By role**); specify the search criteria; and click **OK** to open the Relationship Instances page.
6. Display the relationship instance details in one of two ways:
   - Click the relationship instance ID.
   - Select the radio button next to the relationship instance ID and click **Details**.
7. To view the details for the role instance, click its associated ID in the role instance table.

Editing role instance properties

Perform this task to edit the property values for the selected role instance.

Before you begin

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

About this task

To edit the property values for the selected role instance, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.

4. Select the radio button next to the relationship name and click **Query**.
5. Click one of the query option tabs (**All**, **By ID**, **By property**, or **By role**); specify the search criteria; and click **OK** to open the Relationship Instances page.
6. Display the relationship instance details in one of two ways:
   - Click the relationship instance ID.
   - Select the radio button next to the relationship instance ID and click **Details**.
7. In the role instance table, click the role instance ID to display the role instance details.
8. Edit the role instance property information, as necessary, and click **OK** to save these changes locally.

   Restriction: You can only edit the property values if they have been previously defined for the relationship instance.

Creating new role instances

Perform this task to create a new role instance for a relationship.

Before you begin

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

About this task

To create a new role instance for a relationship, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.
4. Select the radio button next to the relationship name and click **Query**.
5. Click one of the query option tabs (**All**, **By ID**, **By property**, or **By role**); specify the search criteria; and click **OK** to open the Relationship Instances page.
6. Display the relationship instance details in one of two ways:
   - Click the relationship instance ID.
   - Select the radio button next to the relationship instance ID and click **Details**.
7. Locate the role for which you want to create a new instance and click **Create** below the role table to open the New Role Instance page.
8. Enter the key attribute and role property values in their respective **Value** fields and click **OK** to save the new role instance locally.

   Restriction: You can only set the key attribute value when creating the role instance. You cannot change this information after you have applied the changes back to the database. However, you can edit the property values later.

Deleting role instances

Perform this task to delete a selected role instance of a relationship.

Before you begin

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

About this task

To delete a selected role instance of a relationship, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Open the relationships page for the server you want to manage by clicking **Relationships** next to that relationship services MBean.
4. Select the radio button next to the relationship name and click **Query**.
5. Click one of the query option tabs (**All**, **By ID**, **By property**, or **By role**); specify the search criteria; and click **OK** to open the Relationship Instances page.
6. Display the relationship instance details in one of two ways:
   - Click the relationship instance ID.
   - Select the radio button next to the relationship instance ID and click **Details**.
7. Locate the role from which you want to delete the role instance.
8. Click the radio button next to the role instance you want to delete and click **Delete** below the role table.

   The role instance is deleted locally.

Removing relationship instance data from the repository

An application that uses relationships has associated relationship schema and data in a repository. The repository is the database configured to hold the relationship instance data. When you uninstall such an application from a production server, the server does not remove the relationship schema and data from the repository. To do so, you need to remove the existing relationship schema manually.

Before you begin

Make sure that you uninstall the application that uses the relationship schema from all servers that access that schema.

About this task

When you install an application containing relationships, the server creates the corresponding database schema objects including tables, indexes, sequences, and stored procedures. At run time, the tables are populated with the relationship instance data. If you uninstall the application that contains relationships, the tables and instance data are not removed from the database. This design can present a problem if you attempt to reinstall the application after modifying the relationship or role definitions.

Note: If you use the Unit Test Environment (UTE) test server in WebSphere Integration Developer (WID), the relationship schema and data are removed from the repository when an application project is removed.

If you reinstall the application with the same relationship, the old schema is reused. However, if the relationship or role definition is modified in such a way that makes it incompatible with the existing schema, the relationship service throws an exception and terminates the installation of the relationship. The logs show the following exception and message:

```
RelationshipServiceException("table <tablename> already exists, but the
table schema is different from current role definition")
```

The solution for this problem is to remove the existing relationship schema artifacts manually, using the tools supplied by the database platform of your repository, and to reinstall the application.

To remove the existing relationship schema from the repository, perform the following steps.

Procedure

1. Locate the database. The database location depends on the database platform.

Option	Description
Database platform	Location
Derby	WASHOME\derby\databases\RepositoryDB
Other databases	The location is configured during installation and profile creation of the server. For example, if you configured the server automatically and selected the default database name, the name of the database is WPRCSDB. For DB2 on i5/OS®, the referenced container is a collection instead of a database. It is the collection name rather than the database name that is configured during installation and profile creation; and it is the collection rather than the database that is by default named WPRCSDB.

2. Delete the database artifacts making up a relationship: Using the tools for your database platform, perform the following steps to delete all database objects for a given relationship.
   a. Before removing any data from the database in the following steps, make a backup of the database.

      Note: For DB2 on i5/OS, make a backup of the collection before removing any data.
   b. Find the relationships tables. The following tables are created at the time the relationships are installed.

Table	Format
1 table for relationship properties	_<relname>_P_uniqueidentifier
1 table for generating instance IDs for each relationship (on Derby)	_<relname>_S_uniqueidentifier
1 table for role properties for each application-specific role	_<relname>_<rolename>_P_uniqueidentifier

Table	Format
1 table for each application-specific role (for static relationships 1 table for the generic role is also created)	_<relname>_<rolename>_RT_uniqueidentifier

Restriction: Only the first four characters of the relationship name are used. If the database holds tables for multiple relationships, you should distinguish relationship names within the first 4 characters.

c. Find the stored procedures. Stored procedure objects have the following format:

   _<relname>_RS_uniqueidentifier or
   <relname><rolename>_RS_uniqueidentifier

d. Find the sequences. Sequence objects have the following format:

   _<relname>_S_uniqueidentifier

   Restriction: Sequences are not supported under Derby.

e. Using the tools for your database platform, delete the following:

   1) tables
   2) stored procedures
   3) sequences (except for Derby)

Results

The relationship instance data is removed from the database repository.

What to do next

Now you can reinstall the application.

Tutorial: Relationship manager administration

The relationship manager can be used to add, modify, and remove instances of relationships, which correlate identifiers from different environments for the same item of data. This tutorial demonstrates the basic functions of the relationship manager.

This tutorial demonstrates the basic functions of the WebSphere Process Server relationship manager. Relationships are used to correlate identifiers from different environments for the same item of data. For example, in one environment, states are identified by two-letter abbreviations (AZ, TX). In another environment, different abbreviations are used (Ariz., Tex.). A relationship would be created to correlate ″AZ″ in the first environment to ″Ariz″ in the second environment.

The sample relationship referenced here correlates customer IDs. Many business applications maintain databases of customers, and most of these applications assign their own ID to each customer. In an enterprise environment, the same customer likely has a different ID in each business application. In this tutorial, a relationship is defined to correlate customer IDs. The relationship name is ″SampleCustID″. Two roles are defined for this relationship. One role is for the Customer Information System (CIS), and the other role is for the General Ledger (GL) application. This relationship was created by the relationship services sample along with the roles and a small amount of sample data.

The relationship manager is designed to add, modify, and remove role instances of a relationship instance as well as add, modify, and remove relationship instances. WebSphere Integration Developer should be used to create and deploy new relationship definitions. The definitions are stored as XML files that are deployed as part of a J2EE application to a particular server.

Objectives of this tutorial

After completing this tutorial, you will be able to change the values of relationship instances.

Time required to complete this tutorial

This tutorial requires approximately 10 minutes to complete.

Prerequisites

This tutorial uses a relationship that is created by the relationship services technical sample. Before following the steps of this tutorial, go to the samples gallery and perform the steps described in the relationship services sample to create the required relationship and roles.

Related tasks

➪ Installing and accessing the Samples Gallery
Samples of integration application artifacts are available in the Samples Gallery, an option to install when you install this product.

Example: Changing the values of a relationship instance

For a relationship instance, the values of key attributes can be changed on the Relationship Instances page of the administrative console. This example shows the use of that page to change a value for a relationship instance.

About this task

One of your customers has a customer ID of A004 in your CIS application. This same customer has a customer ID of 801 in your GL application. However, due to a data entry error, the relationship instance that correlates the customer IDs of this customer currently has a value of 901 instead of 801 for the GL customer ID. This tutorial takes you through the steps to correct this entry in the relationship.

Procedure

1. Open the 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. Click **Relationships** next to that relationship services MBean.

   A relationship named SampleCustID should be visible.
5. Select the radio button next to SampleCustID, then click **Query**.
6. Locate the relationship instance for the customer
   a. Click the query **By role** tab
   b. In the **Role name** field, select MyGLCustomer_0 from the drop-down list.
   c. In the **Value** field under **Key attributes**, enter`901`

d. Click **OK**

This locates the relationship instance for the requested customer and opens the Relationship Instances page.

7. Click the relationship instance ID.

   This displays 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, then click **Delete** below the role table.

   Note: It should not have any property values associated with it. If any other data appears, you need to look at the role instance and record any data you want to keep.

9. Click **Create** to open the New Role Instance page for creating a new role instance for this relationship instance.

10. Enter `801` in the **Value** field under **Key attributes**, then click **OK**.

    The new role instance is saved, and you should see a new role instance in the table.

Results

You now have the correct customer ID value in the relationship instance for the GL application.

Administering the relationship service

The relationship service maintains relationships and roles in the system. It manages relationship and role definitions and metadata and makes it possible to specify the definition of a relationship and manipulate the instances derived from the definition.

The relationship service makes it possible to capture relationships across different objects. Participants in the relationship are distinguished by the roles they serve. For instance, a Person object ″Joe″ can have an ownership relationship with a Car object ″Subaru with license plate XYZ 123.″ In this example, Joe participates in the relationship with the role ″owner″ while the car participates in the relationship under the role ″owned object.″

Relationship and role definitions

Relationships and roles are described in definitions that you design through the graphical interface of the relationship editor tool in WebSphere Integration Developer. The relationship definition is a template that describes what the relationship should look like, identifying the roles each participant in the relationship can assume. The role definition captures the structure and constraint requirements for the participants. Relationship definitions are stored as XML files that are deployed as part of a J2EE application to a particular server.

For detailed background and task information on creating relationships, identifying relationship types, and using the relationship editor, see the WebSphere Integration Developer Information Center.

How relationships work

At run time, when maps or other WebSphere Process Server components run and need a relationship instance, the instances of the relationships are either created or retrieved, depending on the scenario. The relationship and role instance data can be manipulated through three means:

- WebSphere Process Server component Java snippet invocations of the relationship service APIs
- Relationship transformations in the WebSphere Process Server business object mapping service
- Using the relationship manager tool

The relationship and role instance data is saved in relationship tables that are stored in the database in the default data source that you specify when you configure the relationship service.

The relationship service runs on each server at the cell level. The **Relationship Manager** home page **About** section shows the number of servers in the cell that are running relationship services; the **Relationships** section shows each server name that is running relationship services. Before working with relationship instances, you need to select the server that has the instances of the relationships and roles you want to manage.

For detailed information on using the relationship manager, see the topics on the relationship manager in the WebSphere Process Server Information Center.

The following topics describe the configuration tasks to perform for the relationship services for your WebSphere Process Server environment.

Viewing relationships managed by the relationship service

Perform this task to view a list of the existing relationships that this relationship service manages.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, any WebSphere security role can view this configuration.

About this task

To view the relationship list, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager.**
3. Click **Relationship Services configuration > Relationships**.

   The Relationship collection page displays. Each row shows the version and data source for the associated relationship.

   Tip: To customize the number of rows that display at one time, click **Preferences**. Modify the **Maximum rows** field value and click **Apply**. The default is 25. The total relationship count managed by this relationship service is displayed at the bottom of the page.

What to do next

To see the configuration properties for a relationship, click the relationship name in the relationship collection table.

Viewing relationship properties

Perform this task to view the configuration properties that the relationship service manages at both the relationship service level–as it applies to the relationship service–and at the individual relationship level–as it applies to individual relationships.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, any WebSphere security role can view this configuration.

About this task

To view the configuration properties, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Click **Relationship Services configuration > Relationships**.
4. In the relationship collection table, click the name of the relationship whose properties you want to view.

   The configuration tabbed page displays, showing the name, version, and data source currently in use for the relationship (read-only).

   Note: The version is used for migration purposes. If the old relationship data needs to coexist in the new system, then the old infrastructure version will be set to the old version. Otherwise, it will be set to the current version.
5. To return to the Relationship collection page, click **Back**.

Chapter 6. Administering Business Process Choreographer

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

Chapter 7. Configuring and administering the Common Event Infrastructure

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

Chapter 8. Administering service components

Use the topics in this section to manage service components.

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

Administering business state machines

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

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

- correlation set properties
- display states

Correlation set properties

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

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

Display states

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

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

Finding business state machine instances

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

Before you begin

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

About this task

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

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

Procedure

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

What to do next

Perform your administrative tasks.

Viewing display states

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

Before you begin

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

About this task

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

Procedure

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

What to do next

Perform your administrative tasks.

Administering business rules and selectors

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

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

Considerations for modules containing business rules and selectors

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

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

Considerations for changing business rules or selectors

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

Considerations for replacing a module containing business rules or selectors

When you replace a module that contains business rules or selectors, the server overwrites the copies of the business rules and selectors in the repository. When you replace a module, any changes that you made dynamically are lost. To prevent that loss, export the business rules and selectors used by the module, re-import them into your development environment, and rebuild the module before replacing the module on your production system.

If you have made changes to the business rules or selectors implemented by one module, other modules running in the server may need the current copies of the business rules or selectors. If this is the case, you will have to configure different repositories so that the updated module has no effect on the other modules when you install that module in the server. The topic “Configuring the environment” describes configuring the databases.

Considerations for deleting a module containing business rules or selectors

When you delete a module that contains business rules or selectors from the server, the server does not remove the business rules and selectors from the repository. It keeps these artifacts because it cannot determine if another application or module requires the rules.

If you determine that there is no requirement for a business rule or selector, remove it from the repository. "Removing business rule and selector data from the repository" describes how to clear out unneeded business rules or selectors.

Removing business rule and selector data from the repository

When you uninstall an application that uses business rules or selectors, the server does not remove these artifacts from the repository. Delete the unused artifacts from the database manually after you uninstall applications that use them. Remove the artifacts using the tools supplied by the database platform of your repository. The reason this is done is that business rules and selectors contain business logic which may have been updated when the application was installed, and we do not want to delete this important business data when the application is removed.

Before you begin

Make sure to uninstall all copies of applications that use the business rules or selectors that will be removed. You can back up business rule or selector artifacts before deleting them by exporting them out of the server using the administrative console or wsadmin command.

About this task

When you install an application containing business rule or selector artifacts, the server stores these artifacts in database tables so that you can dynamically update them without changing the application. This also allows other servers to share these artifacts. When you uninstall an application, the server does not automatically remove these artifacts from the database tables because the application may still be installed and running on another server. Deleting the artifacts from the database causes the other running copies of the application to fail when they try to use business rules or selectors.

To remove unneeded business rule and selector artifacts from the repository, perform the following steps.

Procedure

1. Locate the following database tables from which you will delete rows:

   BYTESTORE
   The main table that contains the business rule and selector artifacts

   BYTESTOREOVERFLOW
   The overflow table for the main table

   APPTIMESTAMP
   The table that holds a timestamp of installed applications that contain business rule and selector artifacts

   CUSTPROPERTIES
   The table that holds custom user-defined properties and system properties for a business rules group, rule set, or decision table.

2. Using the tools for your database platform, follow these steps to delete all business rule and selector artifacts for a given application:
   a. Find all of the rows in the BYTESTORE table where the **APPNAME** column is the same as the name of the application.
   b. Record the values of the primary key columns for all of the rows found. The primary key columns for the BYTESTORE table are **ARTIFACTTNS**, **ARTIFACTNAME**, and **ARTIFACTTYPE**.
   c. Delete the rows found in step 2a from the BYTESTORE table.
   d. For each set of primary key values recorded in step 2b, find the rows in the BYTESTOREOVERFLOW table that have the same values in the corresponding columns.

      Note: For a given set of primary key values, there may be zero, one, or more than one row in the BYTESTOREOVERFLOW table.
   e. Delete the rows found in step 2d from the BYTESTOREOVERFLOW table.
   f. For each set of primary key values recorded in step 2b, find the rows in the CUSTPROPERTIES table that have the same values in the corresponding columns.
   g. Delete the rows found in step 2f from the CUSTPROPERTIES table.
   h. Delete the row in the APPTIMESTAMP table where the **APPNAME** column equals the name of the application.

Results

You have removed the unneeded business rules and selector artifacts from the database tables.

Overview of business rules

Use business rules to control the behavior of a business practice.

What is a business rule?

A business rule is anything that imposes structure upon or controls the behavior of a business practice. A rule can enforce business policy, establish common guidelines within an organization, or control access in a business environment.

When to use a business rule

Use business rules to officiate over frequently changing business practices that can come from within a business or mandated from outside a business, such as regulatory agencies. Some typical uses for business rules are as follows:

- Determining current interest rates
- Calculating discounts for products
- Calculating sales tax
- Determining special groups such as senior citizens or preferred customers

How to use business rules

Develop and deploy business rules using the Eclipse-based business rules editors in WebSphere Integration Developer. Manage and modify business rule values using the Web-based business rules manager, which is an option of WebSphere Process Server. For more information about these tools, see the appropriate topics in the WebSphere Integration Developer Information Center and the WebSphere Process Server Information Center, respectively.

Displaying business rule components

Displaying business rule components is the first step in administering a business rule group. From the display you can export or import any or all of the business rule groups or display the tables that comprise the business rule groups.

Before you begin

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

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

About this task

To determine which business rule groups exist in your server, perform the following steps.

Procedure

1. From the administrative console, click **Servers > Application servers**.
2. Click *servername* to select the server from the server list that displays business rules.
3. Click **Business rules** under Business Integration.

Results

The console displays a list of all the business rule components defined with a description of each group.

Exporting business rules using the administrative console:

Export business rule components when you have made changes to the business rule tables. This will create a file that you can import into your development environment, thereby keeping the development artifacts synchronized with the actual production system artifacts.

Before you begin

Before starting this task, you need to display your business rule components as described in “Displaying business rule components.” Click **Servers > Application servers >** ***servername*** **> Business rules > Business rules**.

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an administrator or a configurator to perform this task. When security is not enabled, you must log in to the administrative console with a user ID.

About this task

To export business rules using the administrative console, perform the following steps.

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

Procedure

1. Select the check boxes next to one or more business rule groups and click **Export**.

   The browser displays a list of HTML links to the business rule groups you chose. This is the Business rules export page. Each business rule group has a file extension of `.zip`.
2. Download the files to your system by clicking each file name. When the system prompts you to save the file, click **OK**.

   Note: If you choose to, you can rename the files as you download them.
3. Click **Back** to return to the list of business rule groups.

Results

The system saves the files where you specified. You can then copy them to your test system.

What to do next

You must import the files into your WebSphere Integration Developer environment. For more information, see the WebSphere Integration Developer Information Center.

Importing business rules using the administrative console:

Import business rules in order to update installed business rules without reinstalling an application.

Before you begin

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

Before importing business rules, make sure the following are true or the import will fail:

- The file has an extension of `,zip`.
- The compressed file was created by exporting the business rules from a server.
- The application that uses the business rules group has already been installed on a server in the cell.

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

About this task

Import business rules when you have made changes to business rules in use by installed applications and you are ready to bring those changes into another cluster or server. You can also use this facility to synchronize your development environment with changes in the production environment.

To import business rules using the administrative console, perform the following steps.

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

Procedure

1. Display the business rules on the server to which you are importing the business rules. Click **Servers > Application Servers >** *servername* **> Business rules > Business rules**.
2. Click **Import**.
3. Specify the path to the file on the Preparing for importing business rules page.

What to do next

Display the business rules to verify the changed rules.

Business rules manager

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

Business rules are designed and developed in WebSphere Integration Developer using if/then rule sets and decision tables to implement their operations. Business rules can also be created in WebSphere Business Modeler; however Modeler only supports the creation of business rule tasks, which become rule sets when exported out of Modeler. The rule sets and decision tables are set into templates. The templates control which aspects of a business rule you can modify and by exactly how much. They define the structure of if/then rules, condition cases, and actions for decision tables.

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

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

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

How the business rules manager works

The business rules manager is the main WebSphere Process Server tool that a business analyst uses for runtime rule authoring.

Use the business rules manager to perform the following tasks:

- Retrieve a copy of a business rule from the repository
- Browse and edit a business rule
- Publish a business rule to the repository

The following figure shows how the business rules manager calls and publishes rules.

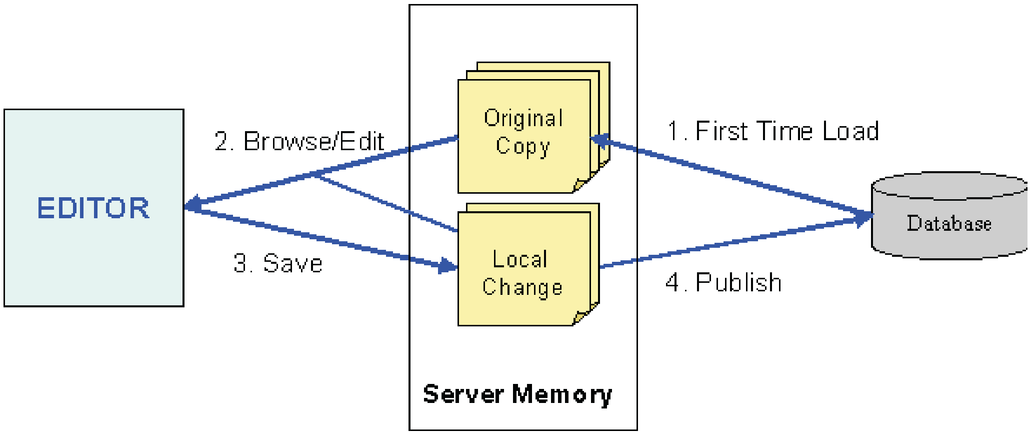


Figure 3. Business rules manager sequence of events

After you log on to the business rules manager, the following events occur when you modify a business rule.

1. When you select a business rule, the business rules manager accesses the business rule group from the repository and stores it in the server memory as an original copy.
2. The business rule group and rule logic are available for editing.
3. You can save changes to a rule set, decision table, and business rule group as a copy in the server memory.
4. You publish the local copy back to the data source. Alternatively, you can cancel the changes with no updates being performed.

Accessing the business rules manager

You access the business rules manager using a Web browser.

Before you begin

Make sure that both the server and client are configured correctly.

About this task

The default URL for accessing the business rules manager is as follows. The URL may vary according to the environment.

http://hostname:port/br

where “hostname” is the name (or IP address) of the current host system, and “port” is the port of the application server where the application was installed.

For example, in the stand-alone environment with only one server, the link is the following:

http://hostname:9080/br

Note: If administrative security is enabled, the preceding link will automatically be switched to a secure link. For example, in the stand-alone environment with only one server, it is https://hostname:9443/br.

If administrative security is not enabled, the Business Rule Groups page opens. If administrative security is enabled on the server, the Login page opens.

If administrative security is enabled, perform the following steps to log in.

Procedure

1. At the Login page type your **User ID**.
2. Type your **Password**.
3. Click **Login**.

Results

The initial page of the business rules manager opens with the existing business rule groups listed in the navigation pane.

What to do next

You can now browse and edit business rule operations and templatize business rules.

Business Rule Groups page and the business rules manager page layout

When the business rules manager opens, the Business Rule Groups page displays, which allows you to browse all of the business rule groups and their defined operations.

The Business Rule Groups page is the first level of navigation. Its page layout includes many elements generic to the other business rules manager pages.

Toolbar

The toolbar contains the following components:

Welcome
Displays the name of the user that is currently logged on.

User identification
Provides the name of the current user preceded with **Welcome User Name**.

Logout
Opens the Login page if administrative security is enabled.

Important: If you log out without publishing, a dialog box appears asking for confirmation.

Search
Opens the Search for Business Rule Groups page, which allows you to quickly locate or narrow a specified set of business rule groups that you want to work with.

Help
Provides access to business rules topics in the WebSphere Process Server Information Center.

Navigation pane

The navigation pane is the left pane. It provides access to the Publish and Revert page and the available business rule groups. The navigation tree enables you to drill down to the rule level you need.

Note: The navigation pane is not displayed on any page that is in the edit mode.

Important: If you retrieve business rule artifacts with a version number greater than the version number of the current model, the business rule artifacts, known as shells, will become flat text items in the navigation pane. As a result, you will not be able to expose the shells further. You should update your current WebSphere Process Server to the latest one, which has a version equal to or higher than the version of the shells.

Publish and Revert
Opens the Publish and Revert page where you can publish changes of business rule groups and rule schedules to the database or revert business rule groups or rule schedules to the original copy that was on the database.

Business Rule Groups
Opens the Business Rule Groups page, which is the top level of browsing. The business rule groups are listed in a navigation tree. You can expand or collapse a business rule group by clicking either the plus (+) or minus (-) next to its display name to show all of its associated rules. When you select a business rule group in the left pane navigation tree, all the child Rule Schedule pages (business rule operations) are listed in the right pane, including all the associated rule sets and decision tables. Clicking any of these opens a corresponding page for editing.

Content area

The content area is the right pane and is the main viewing and editing area. The content area contains a title section, general information section, and page-specific section.

Note: The information displayed in the content area depends on whether you are viewing a Business Rule Group page, Rule Schedule page, Rule Set page, Decision Table page, Publish and Revert page, or Search for Business Rule Groups page.

Title section

The title section includes the following information:

Path information
Provides the path to the page, such as the name of the business rule group and the Rule Schedule page in the following format:

```
BusinessRuleGroup01 > Table1_operation1
```

Example: `CalculateDiscountBRG > CalculateDiscount`

Rule title
Provides the resource display name and type of business rule in the following format:

```
Ruleset112 - Ruleset
```

Examples: `calculateDiscount-Rule Schedule`, `CalculateDiscountRS - Rule Set`

Function buttons

Enable various actions depending on the purpose of the particular page. Not all function buttons are available for a page, and some buttons appear in other sections of the content area. The following table lists the possible function buttons for a page.

Table 17. Function buttons

Button Name	Function
Add Property	Adds properties to a business rule group in the Business Rule Group page or to create a search query in the Search for Business Rule Groups page.
Back	Returns to the previous page.
Cancel	Discards any changes to the resource and returns to the previous page.
Copy	Copies either a decision table or rule set in order to create a new decision table or rule set. You must copy an existing decision table or rule set and then modify its values in order to make a new decision table or rule set.
Edit	Enables editing of the business rule group, rule schedule, rule set, or decision table.
Publish	Publishes the business rule group or rule schedule to the repository.
Revert	Cancels all changes to the rule that have been saved locally and reverts the rule to the original copy that resides in the server memory. Rules cannot be reverted after publishing.
Save	Validates and saves the changes to the local copy and goes back to the previous page. Note that the running state of the server has not been changed. See "Publish" for how to change the server's state.
Search	Initiates the search query on the Search for Business Rule Groups page and returns the business rule groups that match the query as search results on the same Search for Business Rule Groups page.
Sort	Sorts the properties on the business rule groups by the property names in alphabetical ascending order.

Messages field

Shows the status of an action that has been taken to the rule or that an error has occurred. The following are examples of status messages:

```
"calculateDiscount" has been temporarily saved.
```

```
 You may publish the changes from the "Publish and Revert" page.
```

General Information section

The General Information section contains the following information.

Note: The Business Rule Group page includes the General Information section for WebSphere Process Server 6.1 and later. The Search for Business Rule Groups page and the Publish and Revert page do not have this section.

Display Name
Gives the display name of the business rule group, rule set, or decision table for Websphere Process Server 6.1 and later. The display name is read-only in the browse mode but you can modify it in the edit mode on Business Rule Group, Rule Set, and Decision Table pages. Display names can be any string value and can include special characters. Display names of business rule artifacts of the same type do not need to be unique; however, the names of the business rule artifacts still need to be unique in use cases.

If the display name is set, it is used instead of the name value everywhere name values are used, including the navigation pane and when artifacts are displayed in detail. If the display name of a business rule artifact is not set, its name value is used instead. Selecting the **Synchronize with the name** check box synchronizes the display name with the corresponding name value of the target business rule group, rule set, or decision table. The new name takes effect on all pages of the business rules manager when you save the changes made in the edit page.

Last Published
Shows the last published date of the business rule group, rule schedule, rule set, or decision table.

Status
Shows whether the rule schedule, rule set, or decision table is in the edit mode or has been published.

Description
Provides a brief description of the business rule group, rule schedule, rule set, or decision table. You can edit the description in the edit mode of these pages.

Restriction: Do not use CDATA tags when editing the description fields for business rule group components and business rules in the business rules manager as they make business rule groups and business rules uneditable. If CDATA tags exist, open the business rule group or business rule with an XML editor and manually remove the CDATA tags from the description fields.

Page-specific information section

The content of the page-specific information section depends on whether you are viewing a Business Rule Group page, Rule Schedule page, Rule Set page, or Decision Table page. For specific information for each of these pages, see the individual topics.

For the Business Rule Groups page, the section includes the following information:

Business Rules Resources
Lists the display names of the rule schedules, rule sets, and decision tables.

Description
Provides either a brief description or the name of the resource.

Action
Shows the available actions for the corresponding business rule resource. It is initially empty; but when you expand the business rule group, an **Edit** button appears beside each rule.

Publish and Revert page:

The Publish and Revert page is for publishing locally saved changes for business rule groups and rule schedules to the repository. It is also for reverting business rule groups and rule schedules back to the original copy that was in the server memory before the business rule resource was saved locally.

The page-specific information section of the content area includes the following elements.

Changed Business Rules Resources section

This section provides a list of business rule groups and rule schedules available for publishing or reverting, with the following information:

Business Rule Resources
Lists the names of the changed business rule groups and rule schedules. Resources that are ready for publishing have a check box beside them to select or unselect for publishing.

Status Indicates if the resource is the original or has been changed locally.

Description
Provides a brief description of the resource.

Action
Indicates which resource can be reverted. The resource has a **Revert** button in the corresponding **Action** field.

Business Rule Group page:

The Business Rule Group page lists all the business rules resources associated with the business rule group.

You can browse this page or open the editing page for modifying the information for the business rule group or for the associated business rules resources, including adding, deleting, and modifying the custom properties of the business rule group.

The page-specific information section of the content area includes the following elements.

Properties section

This section provides the custom-defined properties for the business rule group.

Restriction: If the business rule group has no custom properties or its list of custom properties is empty, the Properties section will not display in the browse mode. Also, if the business rule group belongs to a version before WebSphere Process Server 6.1, the Properties section and **Edit** button for the business rule group will not display on the Business Rule Group page.

Name Specifies the name of the property. The name must be unique and cannot be empty. Each property can only be defined once in a business rule group.

Value Specifies the value of the property. Each property must have a defined value. It can be an empty string or zero in length, but not null. Setting a property to null is the same as deleting the property.

Business Rules Resources section

This section provides a list of rule schedules, rule sets, and decision tables associated with the business rule group.

Business Rules Resources
Lists the display names of the rule schedules, rule sets, and decision tables associated with the business rule group.

Description
Provides either a brief description or name of the business rule group, rule schedule, rule set, and decision table.

Action
Shows the available actions for the corresponding business rule listing. It is initially empty; but when you expand the group, an **Edit** button appears beside each rule.

Rule Schedule page:

The Rule Schedule page provides an interface for modifying the values of a business rule group in the scheduled rule logic entries. The information is displayed in table format.

From the Rule Schedule page, you can perform such tasks as browsing, modifying, adding, splitting, or deleting effective dates for a business rule. You can also create a new business rule by copying an existing one.

The page-specific information section of the content area includes the following elements.

Scheduled Rule Logic section

This section provides a list of effective business rules that are the building blocks of that rule and enables working with scheduled rule logic entries, such as adding and sorting them.

Note: You can specify the rule logic selection **Date/Time** value in the business rules manager with either local time (uses the time zone of the client running the Web browser) or Universal Time Coordinated (UTC) time.

Start Date/Time
Provides the options of either a specific date or ″no start date.″

Note: The ″no start date″ signifies that the target rule logic is effective for any date before the end date.

End Date/Time
Provides the option of either a specific date or ″no end date.″

Note: The ″no end date″ signifies that the rule logic is effective for the start date and any date after it.

Effective Rule Logic
Specifies the rule set or decision table that is effective in the corresponding time frame.

Action
Provides options for splitting and deleting scheduled rule logic entries.

Default Rule Logic
Provides a default rule logic if no other rule logic is applicable. It is selected when the date does not match any of the other scheduled rule logic entries.

Available Rule Logic section

This section provides a list of rule sets or decision tables that can apply to a particular business rule, with their associated descriptions and actions.

Rule Logic
Specifies the name of the rule set or decision table.

Description
Provides a brief description of the rule set or decision table.

Action
Provides options to facilitate editing or copying rules.

Rule Set page:

The Rule Set page lists the rule ″instances″ for a business rule, their execution order, and associated templates for that rule set.

From the Rule Set page you can browse or edit an existing rule instance using the templates, create a new rule instance from a selected template, specify the execution order of the rules, rename a rule or rule set, browse or edit a rule set display name or rule in a rule set, browse or edit a rule set or rule description or description of a template parameter, save the rule set as a working copy, or delete a rule.

The page-specific information sections of the content area include the following elements.

Rules section

This section provides a list of associated rules with the following information:

Name Provides the name of the rule. This field is visible in edit mode only.

Display Name
Provides the display name of the rule. It is set to the **Name** value if a display name was not specified. It is read-only in the browse mode and editable in the edit mode. The display name can be any string value and can include special characters. It does not need to be unique. Selecting the **Synchronize Name** check box in the **Action** field synchronizes the display name with the corresponding name.

Rule Lists the variables, constraints, range, and enumeration that defines the rule.

Description
Provides more information about each rule in the rule set. It is read only in the browse mode and editable in the edit mode.

Action
Enables reordering rules, deleting rules, and synchronizing the display name with the name by clicking the associated buttons. The actions are available in the edit mode only.

Templates section

This section facilitates creating a new rule in the edit mode using an existing template and includes fields for specifying the following information for the rule:

Template Name
Provides the name of the existing template.

Name Provides a text area for entering and modifying the name of the rule.

Display Name
Provides a text area for entering the display name of the rule. It is set to the **Name** value if a display name is not specified. The display name can be any string value and can include special characters. It does not need to be unique. Selecting the **Synchronize Name** check box synchronizes the display name with the name value of the rule. The new name goes into effect on all pages of the business rules manager when you save the changes made in the edit page.

Note: If the **Synchronize Name** check box is selected, the display name of the rule is disabled and cannot be modified.

Rule Provides a text area for specifying the variables, constraints, range, and enumeration that defines the rule.

Description
Provides more information about each template parameter. It is visible only when a rule set is in the edit mode and you move the mouse over the target template parameter. It is read-only.

Action
Enables adding the rule to the template, deleting the rule from the template or synchronizing the display name with the name value of the rule.

Decision Table page:

The Decision Table page contains the condition cases and actions, their orientation (rows and columns), and the templates associated with that decision table. You open the Decision Table page from the Rule Schedule page.

From the Decision Table page, you can browse or edit an existing condition or action using a template, add a new condition using the templates defined for that decision table, delete a condition, change the order of conditions, change the orientation, change the initialization action rule using the associated template, browse and edit decision table and initialization rule display names and descriptions, and save a decision table as a working copy.

The page-specific information sections of the content area include the following elements.

Initialization Rule section

This section shows the initialization rule of the decision table. The initialization rule displays only if the business rule definition was designed in WebSphere Integration Developer with an initialization action. The initialization rule is invoked directly before the decision table logic is issued and can be used to initialize variables and actions used in the decision table. In the edit mode there are fields for modifying the following information.

Name Provides the name of the initialization rule.

Display Name
Provides the display name of the rule. It is set to the **Name** value if a display name was not specified. The display name can be any string value, can include special characters, and does not need to be unique. Selecting the **Synchronize Name** check box in the **Action** field synchronizes the display name with the corresponding name. The new name goes into effect when you save the changes made in the edit page.

Note: If the **Synchronize Name** check box is selected, the display name of the rule is disabled and cannot be modified.

Rule Lists the variables, constraints, range, and enumeration that defines the initialization rule.

Description
Provides more information about each initialization rule. It is read-only in the browse mode and editable in the edit mode of the decision table.

Action
Enables synchronizing the display name with the name by selecting the **Synchronize Name** check box.

Decision Table section

This section provides the conditional cases, represented in the row and column headings, and the actions, represented as the intersection points of the conditional cases in the table. You can switch the orientation of condition rows from horizontal to vertical, or vice versa, using the **orientation** icon.

Otherwise
Shows the *otherwise* condition of this decision table. The *otherwise* condition is a special condition that will be entered by default if no other condition in the decision table is applicable. The *otherwise* condition displays only if it was specified in the decision table definition that was designed in WebSphere Integration Developer. You cannot add or remove the *otherwise* condition column from a decision table dynamically from the business rules manager.

Templates section

This section facilitates adding a new rule using an existing template.

Search for Business Rule Groups page:

The Search for Business Rule Groups page is for creating a search query to locate or narrow a specified set of business rules groups that you want to work with. You open the Search for Business Rule Groups page by clicking **Search** in the toolbar at the top of the business rules manager.

On the Search for Business Rule Groups page, you can search by the target namespace, business rule group name, custom properties or any combination of these; you can add one or many custom properties, sort custom properties by their names in alphabetical ascending order, move properties up or down inside the property table, or delete custom properties.

The content area of the Search for Business Rule Groups page includes a **Messages** field and page-specific information sections with the following elements.

Search Data section

This section contains the following elements:

Name Provides a text area for entering the name of the business rule group to search for. If you leave this value empty, it will not be included in the search context. The value you enter is used as both a name and a display name. Consequently, the search will look for business rule groups with either the names or the display names that match the entered name value. If you want to specifically search by either name or display name, but not both, you need to indicate such a search through property names.

Example: If you enter IBMSystemName for the name of a property and VIPGroup for the value of the property, the business rules manager will search for business rule groups with the names, but not display names, matching VIPGroup.

Target Namespace
Provides a text area for entering the URL of the business rule group. If you leave this value empty, it will not be included in the search context.

Properties section

This section opens when you click **Add Property** and contains the following elements:

Logical Operator
Provides a drop-down list for selecting ″And″, ″Or″, or ″Not″ to create a search query containing multiple properties.

Name Provides a text area for entering the name of the property. The name must be unique inside the Properties table of the search context and must not be empty.

Query Operator
Provides a drop-down list for selecting from four query operators for each search data field. The query operators are as follows.

Query Operator	Description
is equal to	Indicates that the value of a business rule group name, target name space, or property must match the specified string exactly.

Query Operator	Description
is like	Indicates that the query should look for business rule groups where the value of a business rule group name, target name space, or property is like the specified string. The string can contain wildcard characters. Use the percent character ('%') to specify a wildcard for any number of characters and use the underscore character ('_') to specify a single character wildcard. These wildcard characters must follow SQL syntax.
is not equal to	Indicates that the value of the business rule group name, target name space, or property must not match the specified string.
is not like	Indicates that the query should look for business rule groups where the value of a business rule group name, target name space, or property is not like the specified string. The string can contain wildcard characters as defined in the ″like″ operator.

Value Provides a text area for entering the property value. The value can be empty and is taken into the Search context.

Example: If the value of property PayMethod is left empty and its query operator is set to ″is not equal to,″ the Search will find all the business rule groups whose PayMethod property has the value set to a non-empty string.

Action

Enables moving a property up or down inside the property table and deleting custom properties.

Search Results section

This section contains the following elements:

Rule Groups

Lists the names of the business rule groups that the search query returned.

Status Shows the status of the business rule group returned from the runtime as a search result. The status can be one of the following four kinds of status.

Tip: Clicking on a result business rule group opens its business rule group page.

Status	Description
Same as Local	Indicates that a copy of the result business rule group already exists in the business rule manager and that its content and the content of the result business rule group are exactly the same. Thus, no further action is taken after the search.

Status	Description
Modified from Runtime	Indicates that a copy of the result business rule group already exists in the business rules manager. However, another user session modified the master copy, and so the contents of the local and result business rule groups are different. The business rules manager will automatically update the local copy to get new modifications from the runtime.
Modified in Local	Indicates that a copy of the result business rule group already exists in the business rules manager. However, it has been modified by the current user. The business rules manager will use the local copy for any further actions by the user.
New to Local	Indicates that a copy of the result business rule group does not exist in the business rules manager. In this case, the business rules manager will create a local copy of the result business rule group and will also display it in the navigation pane.

Description
Provides additional information for the business rule group.

Adding, deleting, and modifying business rule group properties

You can use custom properties on business rule groups for searches in order to retrieve subsets of business rule groups that you want to view and modify. You add new custom properties, delete or modify existing properties through the editing pages of business rule groups. The number of custom properties on a business rule group is unlimited.

Before you begin

You need to be in the edit mode for the business rule group.

Restriction: Properties support on business rule groups is available on 6.1 business rule groups and later.

About this task

To add, delete, or modify business rule group properties, perform the following steps.

Procedure

1. Select from the following options.

Option	Description
Option	**Steps**

Option	Description
Add a property to the rule	1. Click **Add Property**. 2. Specify a unique **Name**. The name cannot be empty. 3. Specify a unique **Value**. Each property can only be defined once in a business rule group and must have a defined value. The value can be an empty string or zero in length, but not null. Setting a property to null is the same as deleting the property.
Delete a property	In the **Action** field of the selected property, click **Delete**.
Modify a property	Enter the new name and value in the corresponding field.
Sort properties	Click **Sort** to sort the properties on the business rule groups by the property names in alphabetical ascending order.

2. Click **Save**.

Results

The business rules manager will validate the rules before sending the properties to the server.

Searching business rule groups

You can perform a search query on a business rule group to locate or narrow a specified set of business rule groups that you want to work with. You create a search query based on the name, target name space, custom properties, or any combination of these.

Before you begin

You need to be on the Search for Business Rule Groups page, which you can open by clicking **Search** in the business rules manager toolbar.

About this task

To create a search query, perform the following steps.

Procedure

1. In the **Name** field enter the name of the business rule group to search for. You can leave this value empty; however, it will not be included in the search context. The value you enter is used as both a name and a display name. Consequently, the search will look for business rule groups with either the names or the display names that match the entered name value. If you want to specifically search by either name or display name, but not both, you need to indicate such a search through property names.

   Example: If you enter IBMSystemName for the name of a property and VIPGroup for the value of the property, the business rules manager will search for business rule groups with the names, but not display names, matching VIPGroup.

2. In the **Target Namespace** field enter the URL of the business rule group. You can leave this value empty; however, it will not be included in the search context.
3. For each **Search Data** field select one of the following four query operators.

Option	Description
Query Operator	**Description**
is equal to	Indicates that the value of a business rule group name, target name space, or property must match the specified string exactly.
is like	Indicates that the query should look for business rule groups where the value of a business rule group name, target name space, or property is like the specified string. The string can contain wildcard characters. Use the percent character ('%') to specify a wildcard for any number of characters and use the underscore character ('_') to specify a single character wildcard. These wildcard characters must follow SQL syntax. **Examples:** 1. If you enter ″is like″ ″Discount″ for the business rule group name and ″http://calculateDiscounts″ as the target name space, the search will return all the business rule groups containing that string and with that URL. 2. If you enter ″is like″ ″%Discount%″ for the business rule group name, the search will return all the business rule groups with names such as AirlineTicketDiscount and MovieTicketDiscountRules.
is not equal to	Indicates that the value of the business rule group name, target name space, or property must not match the specified string.
is not like	Indicates that the query should look for business rule groups where the value of a business rule group name, target name space, or property is not like the specified string. The string can contain wildcard characters as defined in the ″like″ operator.

4. **Optional:** Click **Add Property** to add as many properties as needed for the search context.
   a. Specify the **Name**. It must be unique inside the Properties table of the search context and must not be empty.
   b. Specify the **Query Operator**.
   c. Specify the **Value**. The value can be empty and is taken into the search context.

      Example: If the value of property PayMethod is left empty and its query operator is set to ″is not equal to,″ the search will find all the business rule groups whose PayMethod property has the value set to a non-empty string.
   d. Click the up and down arrows in the **Action** field to order the properties.

Tip: You can combine the properties in the **Logical Operator** field using ″And″, ″Or″, or ″Not″ to create a search query containing multiple properties.

Example: To search for all the business rule groups in target namespace ″http://calculateDiscounts″ and with the DiscountedItem property containing string ″men T-Shirts″ and with the Ship Handling property set to value ″Free″, you would use the logical property ″And″.

Note: Adding, deleting, or modifying the properties on the Search for Business Rule Groups page only applies within the search context. It does not affect the properties of any rule object inside the business rules manager.

5. Click **Search**.

Results

The business rule groups that match the search query display in the **Search Results** section on the Search for Business Rule Groups page. The status of the business rule group returned from the runtime as a search result may be one of the following four kinds of status.

Status	Description
Same as Local	Indicates that a copy of the result business rule group already exists in the business rule manager and that its content and the content of the result business rule group are exactly similar. Consequently, no further action is taken after the search.
Modified from Runtime	Indicates that a copy of the result business rule group already exists in the business rules manager. However, another user session modified the master copy, and so the contents of the local and result business rule groups are different. The business rules manager will automatically update the local copy to get new modifications from the runtime.
Modified in Local	Indicates that a copy of the result business rule group already exists in the business rules manager. However, it has been modified by the current user. The business rules manager will use the local copy for any further actions by the user.
New to Local	Indicates that a copy of the result business rule group does not exist in the business rules manager. In this case, the business rules manager will create a local copy of the result business rule group and will also display it in the navigation pane, too.

Note: The synchronization of changes of the business rule groups occurs at the same time as the search results returned and is applied in the business rules manager context. This means that the next operation on an affected business rule group will work with the latest updates of the business rule group.

Example

Examples: Four business rule groups are installed with the following properties:

Business rule group 1
- **Name:** BRDCR002BRG2.brg
- **Target namespace:** http://BRDCR002BRG2/com/ibm/br/rulegroup
- **Properties:**
  - organization, 7GAA
  - department, accounting
  - ID, 0000047
  - ID_cert45, ABC
  - region, NorthRegion

Business rule group 2
- **Name:** BRDCR002BRG3.brg
- **Target namespace:** http://BRDCR002BRG3/com/ibm/br/rulegroup
- **Properties:**
  - organization, 7FAB
  - department, finance
  - ID, 0000053
  - ID_app45, DEF
  - region, NorthCentralRegion

Business rule group 3
- **Name:** BRDCR002BRG4.brg
- **Target namespace:** http://BRDCR002BRG4/com/ibm/br/rulegroup
- **Properties:**
  - organization, 7HAA
  - department, shipping
  - ID, 0000023
  - ID_app45, GHI
  - region, SouthRegion

Business rule group 4
- **Name:** BRDCR002BRG5.brg
- **Target namespace:** http://BRDCR002BRG5/com/ibm/br/rulegroup
- **Properties:**
  - organization, 8JAA
  - department, claims
  - ID, 00000567
  - region, SouthCentralRegion
  - manager, Joe Bean

Retrieve a business rule group by a single property.

Logical Operator	Name	Query Operator	Value
	department	is equal to	accounting

This returns business rule group 1.

Retrieve business rule groups by two properties using the ′%′ multi-character wildcard.

Logical Operator	Name	Query Operator	Value
	region	is like	%Region
AND	ID	is like	00000%

This returns business rule groups 1-4.

Retrieve business rule groups by using the ′_′ single-character wildcard.

Logical Operator	Name	Query Operator	Value
	ID	is like	00000_3

This returns business rule groups 2 and 3.

Retrieve business rule groups by using multiple ′_′ single-character wildcard.

Logical Operator	Name	Query Operator	Value
	region	is like	__uth%Region

This returns business rule groups 3 and 4.

Retrieve a business rule group by using a ′_′ single-character wildcard and not operator.

Logical Operator	Name	Query Operator	Value
	organization	is not like	7__A

This returns business rule group 4.

Retrieve a business rule group by using a ′%′ multi-character wildcard and not operator.

Logical Operator	Name	Query Operator	Value
	organization	is not like	7%

This returns business rule group 4.

What to do next

Click a result business rule group to open its business rule group page.

Working with scheduled rule logic entries

A scheduled rule logic entry identifies information for a rule, such as its effective dates and the if/then rule set or decision table associated with the rule.

About this task

Use the business rules manager to create, modify, or delete scheduled rule logic entries.

Creating scheduled rule logic entries:

You create scheduled rule logic entries from existing entries.

Before you begin

You need to be in the edit mode for the rule you want to create.

About this task

To create a new scheduled rule logic entry, perform the following steps.

Procedure

1. On the Rule Schedule page click **Add Selection Record**.

   A new scheduled rule logic entry is added at the bottom of the list with the **Start Date/Time** and **End Date/Time** fields set to **Jan 1**. A message displays in the **Messages** field indicating that the date/time field values are invalid.
2. Set the **Start Date/Time** field:
   a. Select the month from the drop-down list.
   b. Select the day from the drop-down list.
   c. Enter the year.
   d. Enter the time (in 24-hour format).
3. Set the **End Date/Time** field.
   a. Select the month from the drop-down list.
   b. Select the day from the drop-down list.
   c. Enter the year.
   d. Enter the time (in 24-hour format).

   Restriction: Only one rule logic can be in effect at any one point in time. Rule dates cannot have date/time ranges that overlap.

   Note: Gaps in date/time ranges are allowed. If you have specified a default rule logic, it is used during the gap. As a best practice, you should always specify a default rule logic.
4. Select the **Effective Rule Logic** from the drop-down list.
5. Click **Save** .

Results

A message displays in the **Messages** field indicating that the scheduled rule logic entry has been temporarily saved and that you can publish the changes from the Publish and Revert page.

Related tasks

"Deleting scheduled rule logic entries" on page 173
You can delete existing scheduled rule logic entries from the scheduled rule logic table. When a scheduled rule logic entry is deleted, the associated rule set or decision table definition remains with the rule group and is listed in the Available Rule Logic section of the page. The scheduled rule logic entry can be added back either as the default rule logic or with a specific date and time.

Modifying scheduled rule logic entries:

You can modify the date and time values of existing scheduled rule logic entries.

Before you begin

You need to be in the edit mode for the rule you want to modify.

About this task

To modify a scheduled rule logic entry, perform the following steps.

Procedure

1. On the Rule Schedule page edit the **Start Date/Time** of the scheduled rule logic entry:
   a. Select the month from the drop-down list.
   b. Select the day from the drop-down list.
   c. Enter the year.
   d. Enter the time (in 24-hour format).
2. Edit the **End Date/Time** of the scheduled rule logic entry:
   a. Select the month from the drop-down list.
   b. Select the day from the drop-down list.
   c. Enter the year.
   d. Enter the time (in 24-hour format).

   Restriction: Only one rule logic can be in effect at any one point in time. Rule dates cannot have date/time ranges that overlap.

   Note: Gaps in date/time ranges are allowed. If you have specified a default rule logic, it is used during the gap. As a best practice, you should always specify a default rule logic.
3. Click **Save**.

   Note: If the **Date/Time** fields are invalid, the fields will turn **red** and a message will display in the **Messages** field indicating that the dates/time field values are invalid.

Results

The scheduled rule logic entry is saved locally and is ready to be published to the repository. For more information, see "Publishing and reverting business rules" on page 174.

What to do next

For more information on setting business rule dates, see “Splitting dates in business rules.”

Related tasks

“Deleting scheduled rule logic entries” on page 173
You can delete existing scheduled rule logic entries from the scheduled rule logic table. When a scheduled rule logic entry is deleted, the associated rule set or decision table definition remains with the rule group and is listed in the Available Rule Logic section of the page. The scheduled rule logic entry can be added back either as the default rule logic or with a specific date and time.

Date/Time selections:

Business rules are selected by a date/time specification.

The date is defined either as part of the business rule group’s operation parameter or it is derived at run time. The dates are always in terms of UTC and are specific points in time. Only one rule logic can be effective for an operation at any point in time. When no rule logic is found to be in effect for any point in time, the default rule logic is used.

The business rule group supports the following date/time options, which you access by clicking the icon in the **Start Date/Time** and **End Date/Time** fields:

Specify Date/Time
Specifies a date manually.

Continuous
Uses an automatic date calculation that sets the end date to the earliest start date that is later than the scheduled rule logic entry. The continuous date selection is only available on the **End Date/Time** field.

Note: The continuous selection is used when date ranges of two scheduled rule logic entries are contiguous. A continuous attribute is set to the end date of the first scheduled rule logic entry. When this attribute is set, the start date of the second scheduled rule logic entry is set to the end date of the first scheduled rule logic entry so that you do not have to specify both dates.

No Start Date or No End Date
Does not set a starting or ending boundary, depending on which is selected.

Restriction: The business rule group only supports effective dates. If you need to perform another type of selection, use a selector component.

Splitting dates in business rules:

Splitting a date in a business rule provides a shortcut for modifying a business rule for another purpose.

Before you begin

You need to be in the edit mode for the rule you want to modify.

About this task

To split a scheduled rule logic entry, perform the following steps.

Procedure

1. Click **Split** next to the scheduled rule logic entry.

   A new scheduled rule logic entry is created with a start date of Jan 1; and its fields are in red. A message displays in the **Messages** field indicating that the date/time field values are invalid.
2. Select the start date/time for the new scheduled rule logic entry.

   The end date/time for the original scheduled rule logic entry changes from *continuous* to the start date/time of the new scheduled rule logic entry, and the end date/time of the new scheduled rule logic entry changes to the end date/time of the previous scheduled rule logic entry.
3. Modify the date/times of the new scheduled rule logic entry.
4. Modify the **Effective Rule Logic** to fit the needs of the new rule.

Rule sets

A rule set is a group of if/then statements or rules where the *if* is the condition and the *then* is the action of the rule. Rule sets are best suited for those business rules that have very few condition clauses.

If the condition is met, the action is performed. This may result in more than one action being performed by the rule set. The order of rule processing is determined by the order of the rule statements in the if/then rule set. Therefore, when you modify or add a rule, you need to be sure that it is in the correct sequence.

A rule set may have two kinds of rules–if/then rules and action rules:

- An if/then rule determines what action to take according to the condition of the incoming message.
- An action rule determines what action to take no matter what the incoming message is.

A condition in a rule contains a condition expression, which could be a simple string or an *and*, *or*, or *not*.

You create new rule sets or modify existing rule sets in the business rules manager using templates defined for that rule set. The templates provide the structure that determines how the rule set functions. Rule templates are not shared between rule sets.

Creating rule set entries:

You create a new rule set entry by copying an existing rule set entry and modifying its values.

About this task

To create a new rule set entry, perform the following steps.

Procedure

1. Click **Copy** next to the scheduled rule logic entry for the selected rule set.

   The edit page opens for the new entry, with a title Edit Mode:Copy_of_TableName-Ruleset.

2. In the **Name** field enter a unique name for the new rule set entry.
3. In the **Display Name** field enter a display name for the new rule set entry. The display name does not need to be unique for the rule set. It can be any string value and can contain special characters. If you do not specify a display name, the **Name** value will be used for the display name.

   Note: To synchronize the display name with the corresponding name of the rule set, select the **Synchronize with the name** check box.
4. In the **Description** field enter a short description for the new rule set entry.
5. Modify the values in each condition.

   Tip: To display the parameter settings for each value, place your cursor over a field. A rollover message shows the type of variable and its range.
6. Click the up or down arrow to place the rule in the correct sequence.
7. Click **Save**.

Results

A message displays in the **Messages** field indicating that the rule set entry has been temporarily saved and that you can publish the changes from the Publish and Revert page.

Related tasks

"Deleting scheduled rule logic entries" on page 173
You can delete existing scheduled rule logic entries from the scheduled rule logic table. When a scheduled rule logic entry is deleted, the associated rule set or decision table definition remains with the rule group and is listed in the Available Rule Logic section of the page. The scheduled rule logic entry can be added back either as the default rule logic or with a specific date and time.

Creating rules within rule sets from templates:

You create a new rule within a rule set using the rule templates associated with that rule set.

Before you begin

You need to be in the edit mode for the rule set.

About this task

To create a new rule from a template, perform the following steps.

Procedure

1. Click **New Rule from Template** to display the list of available templates for the rule.
2. Select a template and do the following:
   a. In the **Name** field enter the name of the new rule.
   b. In the **Display Name** field enter a display name for the new rule. The display name does not need to be unique for the rule. It can be any string value and can contain special characters. If you do not specify a display name, the **Name** value will be used for the display name.

Note: To synchronize the display name with the name value, select the corresponding **Synchronize Name** check box in the **Action** field. If the check box is selected, the display name of the rule is disabled and cannot be modified.

   c. Specify the values for the rule in the input fields or select the variables from the drop-down lists.
   d. Enter a description for the rule.
   e. Click **Add**.
3. Click the up or down arrows in the **Action** field to place the rule in the proper order.

   Note: The order of rule processing is determined by the order of the rule statements in the if/then rule set. Therefore, when you modify or add a rule, you need to be sure that it is in the correct sequence.
4. Click **Save**.

What to do next

The rule set is ready for publishing. For more information, see "Publishing and reverting business rules" on page 174.

Related tasks

"Deleting scheduled rule logic entries" on page 173
You can delete existing scheduled rule logic entries from the scheduled rule logic table. When a scheduled rule logic entry is deleted, the associated rule set or decision table definition remains with the rule group and is listed in the Available Rule Logic section of the page. The scheduled rule logic entry can be added back either as the default rule logic or with a specific date and time.

Modifying rules within rule sets using templates:

You modify a rule in a rule set using templates associated with that rule set.

Before you begin

You need to be in the edit mode for the rule set.

About this task

To modify a rule using an existing template, perform the following steps.

Procedure

1. Edit the value by typing over the existing value in the input field or by clicking the down arrow that appears in the field and selecting a value from the drop-down list.
2. If necessary, click the up or down arrows to place the rule in the proper order.

   Note: The order of rule processing is determined by the order of the rule statements in the if/then rule set. Therefore, when you modify or add a rule, you need to be sure that it is in the correct sequence.
3. Click **Save**.

What to do next

The modified rule set is ready for publishing. For more information, see "Publishing and reverting business rules" on page 174

Related tasks

"Deleting scheduled rule logic entries" on page 173
You can delete existing scheduled rule logic entries from the scheduled rule logic table. When a scheduled rule logic entry is deleted, the associated rule set or decision table definition remains with the rule group and is listed in the Available Rule Logic section of the page. The scheduled rule logic entry can be added back either as the default rule logic or with a specific date and time.

Decision tables

A decision table is a scheduled rule logic entry, in table format, that consists of conditions, represented in the row and column headings, and actions, represented as the intersection points of the conditional cases in the table. Decision tables are best suited for business rules that have multiple conditions. Adding another condition is done by simply adding another row or column.

Like the if/then rule set, the decision table is driven by the interaction of conditions and actions. The main difference is that in a decision table, the action is decided by more than one condition, and more than one action can be associated with each set of conditions. If the conditions are met, then the corresponding action or actions are performed.

Templates

You use templates to modify decision table values in the business rules manager. The templates are designed in WebSphere Integration Developer and contained in the business rule definition. The templates determine which aspects of a decision table you can modify and provide a list of valid values to choose from. You create new rows or columns in the table or new actions based on the templates defined for that decision table, and you modify existing conditions or actions that were created with the template. Decision table templates are not shared between decision tables.

Initialization action rules

Decision tables support the use of an initialization action rule, which runs before the decision table is executed and allows for pre-processing, such as for creating business objects or setting initial values. You can modify an initialization action rule in the business rules manager, provided that the business rule definition was designed in WebSphere Integration Developer with an initialization action.

Although only one initialization action rule can be created from a single template, the action rule can have multiple action expressions in it, so it can perform multiple actions. If an initialization rule template is defined for a particular decision table, it can only be used in that table.

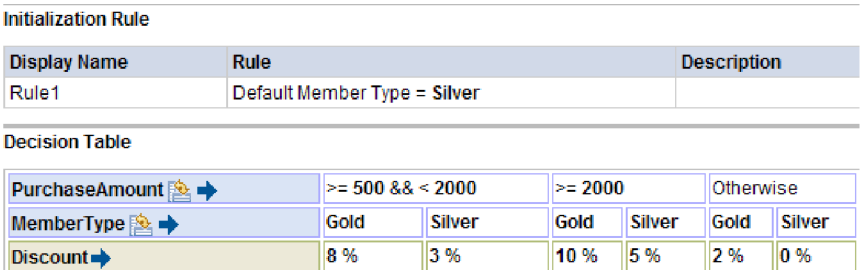
Otherwise conditions

The *otherwise* condition is a special condition that will be entered by default if no other condition in the decision table is applicable.

The *otherwise* condition will only display in the business rules manager if it is included in the decision table definition that was designed in WebSphere Integration Developer. You cannot add or remove it dynamically in the business rules manager.

However, you can incorporate actions defined with templates for the *otherwise* condition. The *otherwise* condition can be used zero or one time for any condition being checked.

The following figure shows a decision table with an *initialization action rule* that sets the default member type to Silver) and *otherwise conditions* that apply to gold and silver customers spending less than $500. The *conditions* PurchaseAmount and MemberType are along the first and second rows, and the *action* Discount is along the third row. The orientation of conditions and actions is shown by arrows.



Initialization Rule

Display Name	Rule	Description
Rule1	Default Member Type = Silver	

Decision Table

PurchaseAmount	>= 500 && < 2000		>= 2000		Otherwise	
MemberType	Gold	Silver	Gold	Silver	Gold	Silver
Discount	8 %	3 %	10 %	5 %	2 %	0 %

Figure 4. Decision table

The example shows that gold customers spending $500 - $1999 get an 8% discount while silver customers spending $500 - $2000 get a 3% discount. Gold customers spending $2000 or more get a 10% discount while silver customers spending $2000 or more get a 5% discount. Gold customers spending less than $500 get a 2% discount while silver customers spending less than $500 get a 0% discount.

Creating decision table entries:

You create a new decision table entry by copying an existing decision table entry and modifying its values.

About this task

To create a decision table entry, perform the following steps.

Procedure

1. Click **Copy** next to the scheduled rule logic entry for the selected decision table.

   The edit page opens for the new entry, with a title Edit Mode:Copy_of_TableName-Decision Table.
2. In the **Name** field enter a name for the new decision table entry.
3. In the **Display Name** field enter a display name for the new decision table entry. The display name does not need to be unique for the decision table. It can be any string value and can contain special characters. If you do not specify a display name, the **Name** value will be used for the display name.

   Note: To synchronize the display name with the name value, select the corresponding **Synchronize with the name** check box.

4. In the **Description** field enter a short description of the new decision table entry.
5. Modify the **values** in each condition.

   Tip: To display the parameter settings for each value, place your cursor over a field. A rollover message displays showing the type of variable and its range.
6. Click **Save**.

Results

A message appears in the message field indicating that the decision table entry has been temporarily saved and that you can publish the changes from the Publish and Revert page. For more information, see "Publishing and reverting business rules" on page 174.

Related tasks

"Deleting scheduled rule logic entries" on page 173
You can delete existing scheduled rule logic entries from the scheduled rule logic table. When a scheduled rule logic entry is deleted, the associated rule set or decision table definition remains with the rule group and is listed in the Available Rule Logic section of the page. The scheduled rule logic entry can be added back either as the default rule logic or with a specific date and time.

Special actions menu:

The Decision Table page has a **Special actions** menu to edit the values in a decision table or modify the structure and variables of a template.

The **Special actions** menu is available for any field that has the **Special actions** icon beside it when a decision table is in the edit mode. Clicking the **Special actions** icon for the field opens a list of available options for the field. The following table lists the possible options.

Note: Reordering the columns or rows only affects the visual presentation of the table and has no effect on the order in which the conditions and actions are processed.

Menu option	Description	Modifies condition	Modifies action
Add below	Adds a new condition value (row) below the present cell (orientation is vertical)	Yes	
Add to the right	Adds a new condition value to the right of the cell (orientation is horizontal)	Yes	
Change template	Allows modifications to the cell value	Yes	Yes
Move up	Moves the condition value or variable to the row above (orientation is vertical)	Yes	

Menu option	Description	Modifies condition	Modifies action
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 decision table entries:

You edit a decision table by directly entering the new value into the appropriate input field or by selecting a value from the field's list box options.

Before you begin

You need to be in the edit mode for the decision table you want to modify.

About this task

To modify the values of a decision table, perform the following steps.

Procedure

1. Edit the value by typing over the existing value in the input field or by clicking the down arrow that appears in the field and selecting a value from the drop-down list.

   Restriction:

   - The initialization rule will only be displayed in the decision table if it is included in the business rule definition that was designed in WebSphere Integration Developer. Only one initialization action rule can be associated with a single template, but the action rule can have multiple action expressions in it.
   - The *otherwise* condition will only be displayed in the decision table if it is included in the business rule definition that was designed in WebSphere Integration Developer. You cannot add or remove the *otherwise* condition in the business rules manager; however, you can incorporate actions defined with templates for the *otherwise* condition.
2. Click the **Special actions** icon beside the field to open a list of available options for the field, and select an action, as desired.

   Note: Selecting an option for reordering the columns or rows only affects the visual presentation of the table and has no effect on the order in which the conditions and actions are processed.

3. Click **Save**.

Results

The rule is modified locally and is ready to be published to the repository. For more information, see "Publishing and reverting business rules" on page 174.

Related tasks

"Deleting scheduled rule logic entries"
You can delete existing scheduled rule logic entries from the scheduled rule logic table. When a scheduled rule logic entry is deleted, the associated rule set or decision table definition remains with the rule group and is listed in the Available Rule Logic section of the page. The scheduled rule logic entry can be added back either as the default rule logic or with a specific date and time.

Modifying template values of decision tables:

You modify the structure and values of a decision table template by using the **Special actions** menu and by directly entering values into the appropriate input fields.

Before you begin

You need to be in the edit mode for the decision table you want to modify.

About this task

To modify a decision table template, perform the following steps.

Procedure

1. Click the **Special action** icon located beside the decision table field you want to modify to open the list box of options for the field, and select **Change Template**.
2. Type the new value for the template over the existing value in the input field.
3. Click **Change** in the **Action** column.
4. Click **Save**.

Results

The decision table template has been modified and is now ready for publishing. For more information, see "Publishing and reverting business rules" on page 174.

Deleting scheduled rule logic entries

You can delete existing scheduled rule logic entries from the scheduled rule logic table. When a scheduled rule logic entry is deleted, the associated rule set or decision table definition remains with the rule group and is listed in the Available Rule Logic section of the page. The scheduled rule logic entry can be added back either as the default rule logic or with a specific date and time.

Before you begin

You need to be in the edit mode for the rule you want to delete.

About this task

To delete a scheduled rule logic, perform the following steps.

Procedure

1. On the Rule Schedule page select the scheduled rule logic, and click **Delete**.

   The scheduled rule logic is deleted. The associated rule set or decision table definition remains with the rule group and is listed in the Available Rule Logic section of the page.

   Note: Each operation on a business rule group must have at least one active business rule associated with it, either as a scheduled rule logic entry or as a default rule logic. Attempting to delete all scheduled rule logic entries will result in an error.
2. Click **Save**.

Results

The scheduled rule logic entry is temporarily saved and is ready for publishing to the repository.

Publishing and reverting business rules

When you save any part of a business rule group, the changes are saved locally. In order to store the changes to the data source that resides on the application server, you need to *publish* the changes. Alternatively, you can cancel the changes that have been saved locally to a business rule by *reverting* the rule to its original state.

Before you begin

You need to be on any business rules manager page that has a navigation pane.

About this task

The server publishes changes at the business rule group and rule schedule levels. At the publishing stage, the business rules manager does not need to do any validations because the business rules manager validates all changes you enter on each edit page when you save the information.

To publish the changes to a business rule group or rule schedule, perform the following steps.

Procedure

1. Click **Publish and Revert**.
2. On the Publish and Revert page select the business rule groups and rule schedules to send to the repository by clicking their check boxes in the left-hand column of the content area. You can publish all the business rule groups and rule schedules together as a single transaction, or just a subset of them.

   Note: To cancel all changes that have been saved locally to a business rule group or rule schedule and replace the changed resource with the original copy in the server memory, select the check box for the business rule group or rule schedule and click **Revert**. Business rule groups and rule schedules cannot be reverted after publishing since publishing changes the original copy that resides in the server memory.

3. Click **Publish**.

   The selected business rule groups and rule schedules are written to the server memory.

What to do next

The business rule is ready to be exported to the data source.

Troubleshooting the business rules manager

Some of the problems you might encounter using the business rules manager are login errors, login conflicts, and access conflicts.

You can take various steps to troubleshoot these problems.

Resolving login errors:

A log in error occurs upon logging in.

Before you begin

About this task

The login error message is as follows:

```
Unable to process login. Please check User ID and password and try again.
```

Note: Login errors occur only when administrative security is enabled and either the user ID, password, or both, are incorrect.

To resolve login errors, perform the following steps.

Procedure

1. Click **OK** on the error message to return to the Login page.
2. Enter the valid **User ID** and **Password**.
   - If passwords are case sensitive, make sure that Caps Lock key is not on.
   - Make sure the user ID and password are spelled correctly.
   - Check with the system administrator to be sure that the user ID and password are correct.
3. Click **Login**.

What to do next

If you resolve the login error, you will now be able to log in to the business rules manager. If the error is not resolved, contact your system administrator.

Resolving login conflict errors:

A login conflict error occurs when another user with the same user ID is already logged in to the application.

Before you begin

About this task

The login conflict message is as follows:

```
Another user is currently logged in with the same User ID. Select from the
following options:
```

Usually this error occurs when a user closed the browser without logging out. When this condition occurs, the next attempted login before the session timeout expires results in a login conflict.

Note: Login conflict errors occur only when administrative security is enabled.

To resolve login conflict errors, select from the following three options:

- Return to the Login page.

  Use this option if you want to open the application with a different user ID.
- Log out the other user with the same user ID.

  Use this option to log out the other user and start a new session.

  Note: Any unpublished local changes made in the other session will be lost.
- Inherit the context of the other user with the same user ID and log out that user.

  Use this option to continue work already in progress. All unpublished local changes in the previous session that have been saved will not be lost. The business rules manager will open to the last page displayed in the previous session.

Resolving access conflict errors:

An access conflict error occurs when a business rule is updated in the data source by one user at the same time another user is updating the same rule.

Before you begin

This error is reported when you publish your local changes to the repository.

About this task

To correct access conflict errors, perform the following actions:

- Find the source of the business rule that is causing the error and check if your changes on the local machine are still valid. Your change may no longer be required after the changes done by another user.
- If you choose to continue working in the business rules manager, you must reload those business rule groups and rule schedules in error from the data source as your local changes of business rule groups and rule schedules in error are no longer usable. Reload a business rule group or rule schedule page, by clicking **Reload** in the Publish and Revert page of the rule for which the error was reported. You can still use local changes in other business rule groups and rule schedules that are not in error.

Overview of selector components

As businesses change, the business processes that drive them must change, too. Some of those changes may require that certain processes return different results than as originally designed without changing the design of the process. The selector component provides the framework for that flexibility.

Selector components provide a single interface to a service that may change results based on certain criteria. The selector component includes an interface and a selector table. Based on the criteria, the selector table determines which component (named the target component) processes the request. The server returns the processing result provided by a target component to the client.

When building a business process, the solution architect identifies the need for a selector component and defines the interface and selector table that the selector component uses to complete processing. The tasks involved in developing a selector component are described in the WebSphere Integration Developer Information Center.

Administering a selector component consists of tasks related to the selector component or tasks related to the selector table.

Restriction: To access any of the selector component pages, you must supply a user ID when you log in to the administrative console. If you are logged in without a user ID, you will receive a warning to log out and log back in with a valid user ID.

Displaying selector components

Displaying selector components is the first step in administering selector components. From the display you can export any or all of the selector components or display the selector tables which comprise the selector components.

Before you begin

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

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

About this task

To determine which selector components exist in your server, perform the following steps.

Restriction: To access any of the selector component pages, you must supply a user ID when you log in to the administrative console. If you are logged in without a user ID, you will receive a warning to log out and log back in with a valid user ID.

Procedure

1. In the navigation pane, click **Servers** to display the different server types.
2. Click **Application servers** to expand the Application server list.
3. Click the name of your server in the server list.
4. Under **Business Integration** click **Selectors > Selectors** .

   The console displays a list of all the selector components with each component’s description.

Displaying selector tables

Displaying selector tables is the first step in administering the tables. The resulting display is a list of target components from which you can alter the processing criteria, change the target component that runs for a specific criterion, add a new target component or delete a target component from the table, thereby removing a criterion.

Before you begin

You must be at the administrative console for the WebSphere Process Server.

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

About this task

Display a selector table to determine the entries that comprise the table and to do other selector table-related tasks. To display a selector table, perform the following steps.

Procedure

1. Display the selector components by clicking **Servers > Application servers >** *servername* **> Business Integration > Selectors > Selectors**.
2. Click the selector name from the selector components display to view the selector tables in the selected component.
3. Click one of the selector tables in the display to view the target components that comprise the selector table.

Changing target components

Changing target components allows you to alter selector component processing by either changing the selection criteria for a specific target component, changing the target component for a selection criteria, or changing both the selection criteria and the target component.

Before you begin

To perform this task, a selector table must exist.

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

About this task

Change a target component to alter the selection criteria or use a different target component for an entry in the selector table. To change target components, perform the following steps.

Important: Before changing target components for long-running applications, stop the application. Do not change target components while a long-running application is processing.

Procedure

1. Display the selector table as described in "Displaying selector tables." Click **Servers > Application Servers >** *servername* **> Business Integration > Selectors > Selectors >** *selectorname*.

2. Click one of the selector tables in the display to view the target components that comprise the selector table.
3. Click the target ID of the target component that you want to change.
4. Change the entry.

Portion of entry to change	Steps to change
Target destination	1. Click the arrow next to the target component list to display the eligible target components. 2. Select the target component from the list.
Selection criteria	1. Type over either the **Start Date**, **End Date** or both. The date you enter depends on the locale of your system and will display according to the locale format. For the US English locale the format displayed is the following: • Month • Day of month • Year in YYYY format. • Time in HH:MM:SS format • Time zone **Important:** The **Start Date** you specify must be prior to the **End Date** or you will be unable to commit this change.
Target destination and selection criteria	1. Click the arrow next to the target component list to display the eligible target components. 2. Select the target component from the list. 3. Type over either the **Start Date**, **End Date** or both. The date you enter depends on the locale of your system and will display according to the locale format. For the US English locale the format displayed is the following: • Month • Day of month • Year in YYYY format. • Time in HH:MM:SS format • Time zone **Important:** The **Start Date** you specify must be prior to the **End Date** or you will be unable to commit this change.

5. Optional: Click the **Default** check box to make this the default target component.

   If the selection criteria does not fall within the range of any other target components, the selector component uses this target component.
6. Click **Apply** to continue working in this display, or click **OK** to return to the target component display.
7. Click **Save** on the target component display to save the changes to the selector table.

Results

The selector table file now contains the updated selection criteria and target components. The selector component uses the updated selector table to process the next request received.

Adding target components

Add a target component when you need additional processing for a different selection criterion than currently exists in the selector table.

Before you begin

To perform this task, a selector table must exist.

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

About this task

Add a target component when you need additional flexibility in your business process. The new components can be added while the selector component is active.

To add a target component, perform the following steps.

Procedure

1. Display the selector table as described in “Displaying selector tables”. Click **Servers > Application Servers >** *servername* **> Business Integration > Selectors > Selectors >** *selectorname*.
2. Click one of the selector tables in the display to view the target components that comprise the selector table.
3. Click **New** to display a pre-filled target component page.
4. Edit the target destination information to fit the application requirements as described in “Changing target components.”
5. Click **OK** to save the target component and return to the target components display.

Results

The selector table now contains the new target components. The selector component uses the updated selector table to process the next request received.

Deleting target components

Deleting target components alters selector component processing by removing the entry in the selector table for a specific selection criterion.

Before you begin

To perform this task, a selector table must exist.

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

About this task

Delete a target component when the processing is no longer required for the business process. After deleting a target component, if a request comes in and it does not match any other specific selection criteria, the default criteria processes the request.

To delete target components, perform the following steps.

Procedure

1. Display the target components as described in “Displaying selector tables.”
2. Click the check box next to the target components that you want to delete, and click **Delete**.

   The system updates the page by displaying the remaining target components.
3. Click **Save**.

   The system saves the updated selector table with the entries representing the remaining target components.

Results

The selector table file now contains only the remaining target components. The selector component uses the updated selector table to process the next request received.

Exporting selector components using the administrative console

Export selector components when you have made changes to the selector tables. This will create a file that you can import into your development environment, thereby keeping the development artifacts synchronized with the actual production system artifacts.

Before you begin

Before starting this task, you need to display your selector components as described in “Displaying selector components.” Click **Servers > Application servers >** ***servername*** **> Business Integration > Selectors > Selectors**.

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as an administrator or a configurator to perform this task. When security is not enabled, you must log in to the administrative console with a user ID.

About this task

To export selectors using the administrative console, perform the following steps.

Procedure

1. Select the check boxes next to one or more selectors and click **Export**.

   The browser displays a list of HTML links to the selector components you chose. This is the Selectors to Export page. Each selector has a file extension of `.zip`.
2. Download the files to your file system by clicking each file name. When the system prompts you to save the file, click **OK**.

   Note: If you choose to, you can rename the files as you download them.

3. Click **Back** to return to the list of selectors.

Results

The system saves the files where you specified.

Importing selector components using the administrative console

Import selectors in order to update installed selector components without reinstalling an application.

Before you begin

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

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

About this task

Import selectors when you have made changes to selectors in use by installed applications, and you are ready to bring those changes into another cluster or server. You can also use this facility to synchronize your development environment with changes in the production environment.

To import selectors using the administrative console, perform the following steps.

Tip: You can also import selector components using the command line.

Procedure

1. Display the selectors on the server to which you are importing the selector components as described in "Displaying selector components." Click **Servers > Application servers >** *servername* **> Business Integration > Selectors > Selectors**.
2. Click **Import**.
3. Specify the path to the file on the Preparing for importing selectors page.

What to do next

Display the selector tables for the updated selectors to verify the changes.

Chapter 9. Working with bindings

At the core of a service-oriented architecture is the concept of a *service*, a unit of functionality accomplished by an interaction between computing devices. An *export* defines the external interface (or access point) of a module, so that SCA components within the module can provide their services to external clients. An *import* defines an interface to services outside a module, so the services can be called from within the module. You use protocol-specific *bindings* with imports and exports to specify the means of transporting the data into or out of the module.

Exports

External clients can invoke SCA components in an integration module over a variety of protocols (such as HTTP, JMS, MQ, and RMI/IIOP) with data in a variety of formats (such as XML, CSV, COBOL, and JavaBean). Exports are components that receive these requests from external sources and then invoke WebSphere Process Server components using the SCA programming model.

For example, in the following figure, an export receives a request over the HTTP protocol from a client application. The data is transformed into a business object, the format used by the SCA component. The component is then invoked with that data object.

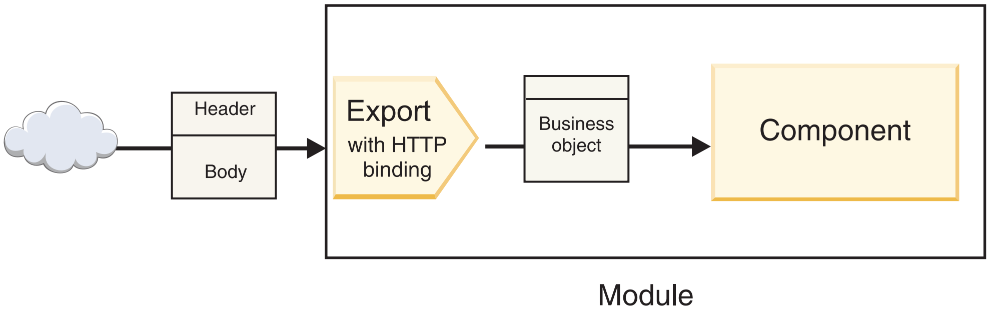


Figure 5. An export with HTTP binding

Imports

An SCA component might want to invoke a non-SCA external service that expects data in a different format. An import is used by the SCA component to invoke the external service using the SCA programming model. The import then invokes the target service in the way that the service expects.

For example, in the following figure, a request from an SCA component is sent, by the import, to an external service. The business object, which is the format used by the SCA component, is transformed to the format expect by the service, and the service is invoked.

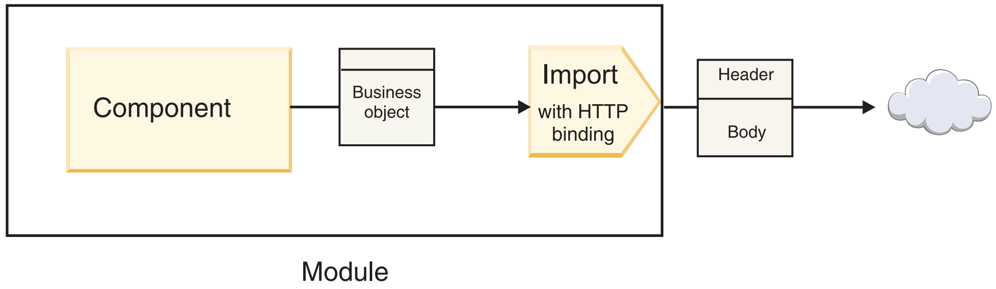


Figure 6. An import with HTTP binding

List of bindings

You use WebSphere Integration Developer to generate a binding for an import or export and to configure the binding. The types of bindings that are available are described in the following list.

- SCA

  The SCA binding, which is the default, lets your service communicate with services in other SCA modules. You use an import with an SCA binding to access a service in another SCA module. You use an export with an SCA binding to offer a service to other SCA modules.

  For more information about SCA bindings, see SCA bindings.
- Web service

  A Web service binding lets you access an external service using interoperable SOAP messages and qualities of service. The Web service binding can use a transport protocol of either SOAP/HTTP (SOAP over HTTP) or SOAP/JMS (SOAP over JMS). You can also use Web service bindings to include an unreferenced attachment as part of the SOAP message.
- HTTP

  The HTTP binding lets you access an external service using the HTTP protocol, where non-SOAP messages are used, or where direct HTTP access is required. This binding is used when you are working with Web services that are based on the HTTP model (that is, services that use well-known HTTP interface operations such as GET, PUT, DELETE, and so on).
- EIS

  The EIS (enterprise information system) binding, when used with a JCA resource adapter, lets you access services on an enterprise information system or make your services available to the EIS.
- JMS bindings

  Java Message Service (JMS), generic JMS, and WebSphere MQ JMS (MQ JMS) bindings are used for interactions with messaging systems, where asynchronous communication through message queues is critical for reliability.

  An export with one of the JMS bindings watches a queue for the arrival of a message and asynchronously sends the response, if any, to the reply queue. An import with one of the JMS bindings builds and sends a message to a JMS queue and watches a queue for the arrival of the response, if any.
  - JMS

    The JMS binding lets you access the WebSphere-embedded JMS provider.
  - Generic JMS

    The generic JMS binding lets you access a non-IBM vendor messaging system.
  - MQ JMS

The MQ JMS binding lets you access the JMS subset of a WebSphere MQ messaging system. You would use this binding when the JMS subset of functions is sufficient for your application.

- MQ

  The WebSphere MQ binding lets you communicate with MQ native applications, bringing them into the service oriented architecture framework and providing access to MQ-specific header information. You would use this binding when you need to use MQ native functions.
- Enterprise Java Bean

  The Enterprise Java Bean (EJB) import binding lets SCA components invoke services provided by J2EE business logic running on a J2EE server.

Export and import binding overview

An export lets you make services in an integration module available to external clients, and an import makes it possible for your SCA components in an integration module to call external services. The binding associated with the export or import specifies the relationship between protocol messages and business objects. It also specifies the way that operations and faults are selected.

Flow of information through an export

An export receives a request, which is intended for the component to which the import is wired, over a specific transport determined by the associated binding (for example, HTTP).

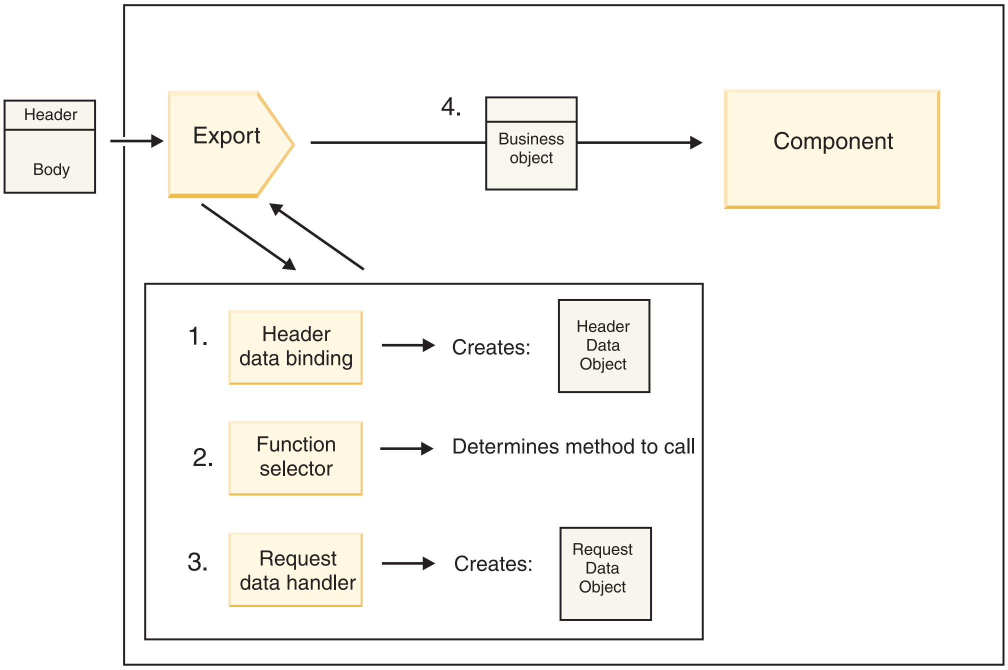


Figure 7. Flow of a request through the export to a component

When the export receives the request, the following sequence of events occurs:

1. For WebSphere MQ bindings only, the header data binding transforms the protocol header into a header data object.

2. The function selector determines the native method name from the protocol message. The native method name is mapped by the export configuration to the name of an operation on the interface of the export.
3. The request data handler or data binding on the method transforms the request to a request business object.
4. The export invokes the component method with the request business object.
   - The HTTP export binding and the Web Services export binding invoke the SCA component synchronously.
   - The JMS, Generic JMS, MQ JMS, and WebSphere MQ export bindings invoke the SCA component asynchronously.

Note that an export can propagate the headers and user properties it receives over the protocol, if context propagation is enabled. Components that are wired to the export can then access these headers and user properties. See the "Propagation" topic in the WebSphere Integration Developer information center for more information.

If this is a two-way operation, the component returns a response.

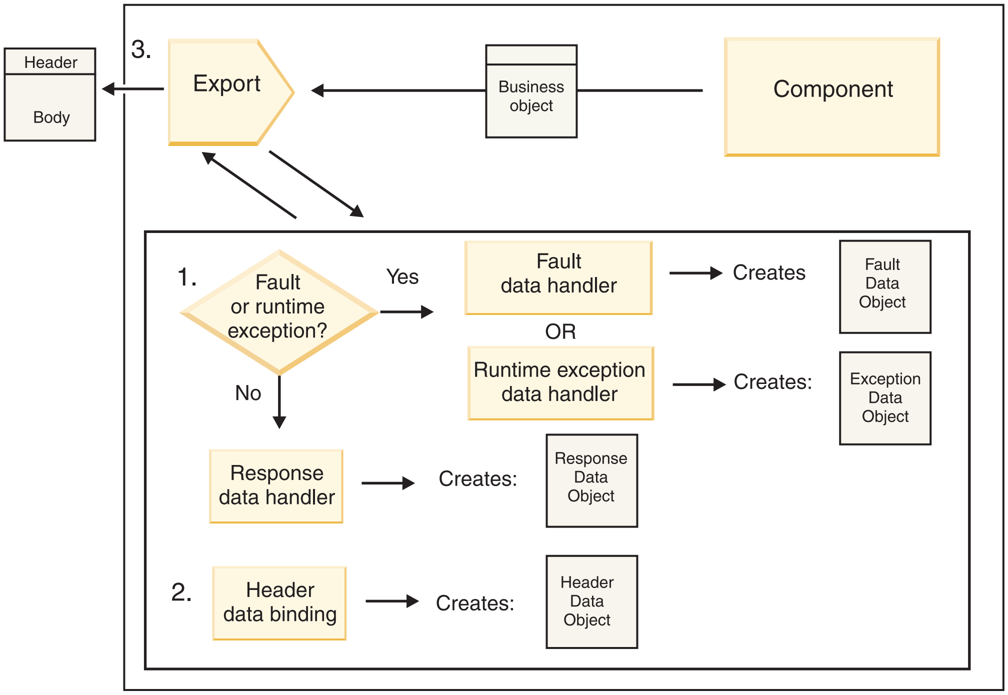


Figure 8. Flow of a response back through the export

The following sequence of steps occurs:

1. If a normal response message is received by the export binding, the response data handler or data binding on the method transforms the business object to a response.

   If the response is a fault, the fault data handler or data binding on the method transforms the fault to a fault response.

   For HTTP export bindings only, if the response is a runtime exception, the runtime exception data handler, if configured, is called.
2. For WebSphere MQ bindings only, the header data binding transforms the header data objects into protocol headers.

3. The export sends the service response over the transport.

Flow of information through an import

Components send requests to services outside the module using an import. The request is sent, over a specific transport determined by the associated binding.

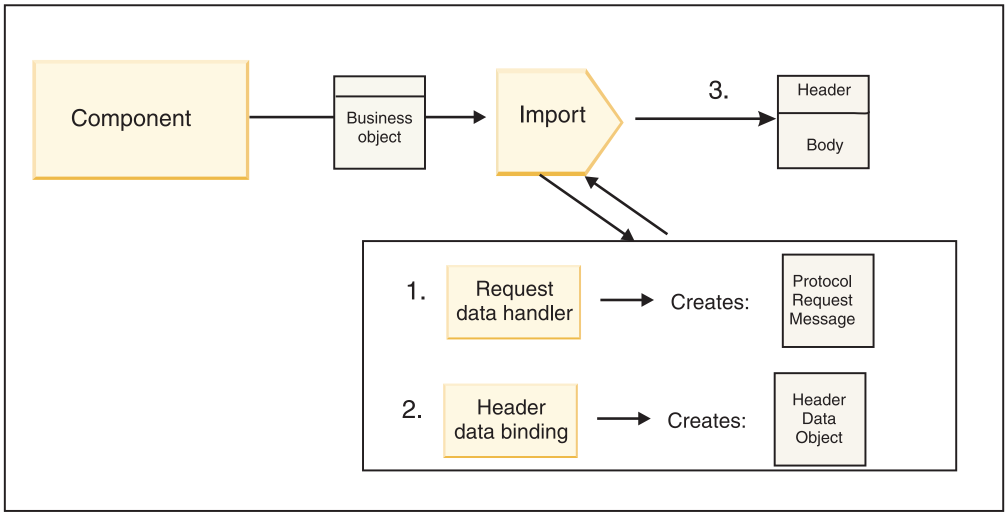


Figure 9. Flow from a component through the import to a service

The component invokes the import with a request business object.

Note:

- The HTTP import binding, the Web Services import binding, and the EJB import binding should be invoked synchronously by the calling component.
- The JMS, Generic JMS, MQ JMS, and WebSphere MQ import binding should be invoked asynchronously.

After the component invokes the import, the following sequence of events occurs:

1. The request data handler or data binding on the method transforms the request business object into a protocol request message.
2. For WebSphere MQ bindings only, the header data binding on the method sets the header business object in the protocol header.
3. The import invokes the service with the service request over the transport.

If this is a two-way operation, the service returns a response, and the following sequence of steps occurs:

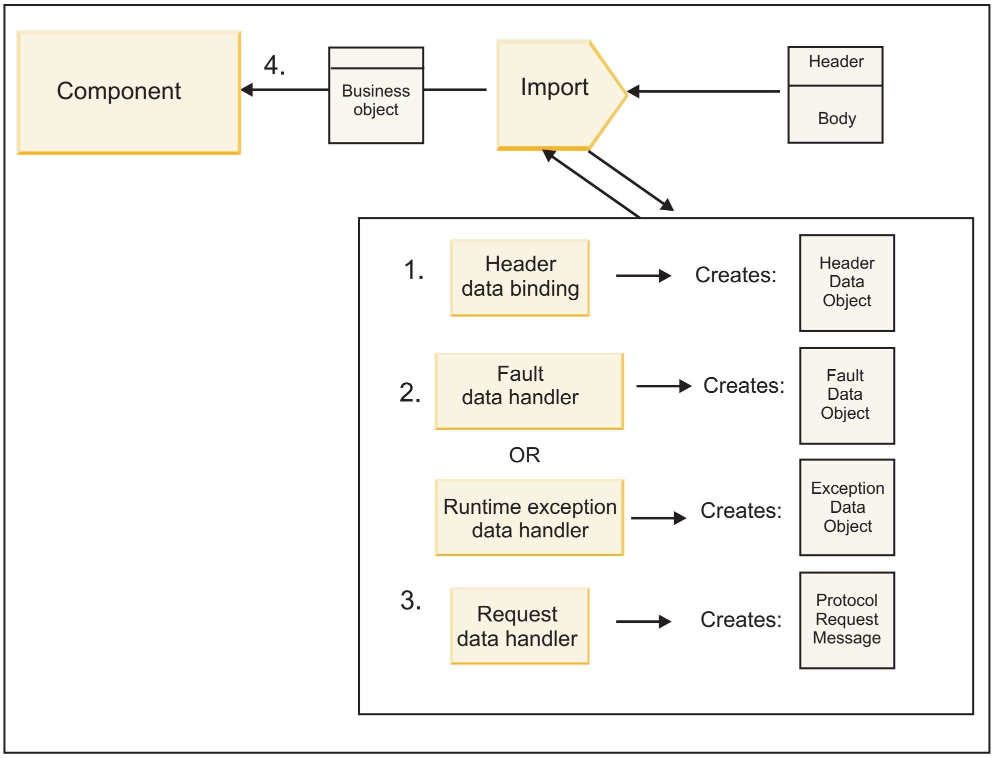


Figure 10. Flow of a response back through the import

1. For WebSphere MQ bindings only, the header data binding transforms the protocol header into a header data object.
2. A determination is made about whether the response is a fault.
   - If the response is a fault, the fault selector inspects the fault to determine which WSDL fault it maps to. The fault data handler on the method then transforms the fault to a fault response.
   - If the response is a runtime exception, the runtime exception data handler, if configured, is called.
3. The response data handler or binding on the method transforms the response to a response business object.
4. The import returns the response business object to the component.

Export and import binding configuration

One of the key aspects of export and import bindings is data format transformation, which indicates how data is mapped (deserialized) from a native wire format to a business object or how it is mapped (serialized) from a business object to a native wire format. For bindings associated with exports, you can also specify a function selector to indicate which operation should be performed on the data. For bindings associated with exports or imports, you can indicate how faults that occur during processing should be handled.

In addition, you specify transport-specific information on bindings. For example, for an HTTP binding, you specify the endpoint URL. You can find more information in the WebSphere Integration Developer information center. For example, for the HTTP binding, the transport-specific information is described in the “Generating an HTTP import binding” and “Generating an HTTP export binding” topics.

Data format transformation in imports and exports

When an export or import binding is configured in WebSphere Integration Developer, one of the configuration properties that you specify is the data format used by the binding.

- For export bindings, in which a client application sends requests to and receives responses from an SCA component, you indicate the format of the native data. Depending on the format, the system selects the appropriate data handler or data binding to transform the native data to a business object (which is used by the SCA component) and conversely to transform the business object to native data (which is the response to the client application).
- For import bindings, in which an SCA component sends requests to and receives responses from a service outside the module, you indicate the data format of the native data. Depending on the format, the system selects the appropriate data handler or data binding to transform the business object to native data and vice versa.

WebSphere Process Server provides a set of predefined data formats and corresponding data handlers or data bindings that support the formats. You can also create your own custom data handlers and register the data format for those data handlers. For more information, see the “Developing data handlers” topic in the WebSphere Integration Developer information center.

- *Data handlers* are protocol-neutral and transform data from one format to another. In WebSphere Process Server, data handlers typically transform native data (such as XML, CSV, and COBOL) to a business object and a business object to native data. Because they are protocol-neutral, you can reuse the same data handler with a variety of export and import bindings. For example, you can use the same XML data handler with an HTTP export or import binding or with a JMS export or import binding.
- *Data bindings* also transform native data to a business object (and vice versa), but they are protocol-specific. For example, an HTTP data binding can be used with an HTTP export or import binding only. Unlike data handlers, an HTTP data binding cannot be reused with an MQ export or import binding.

As noted earlier, you can create custom data handlers, if necessary. You can also create custom data bindings; however, it is recommended that you create custom data handlers because they can be used across multiple bindings.

Data handlers

Data handlers are configured against export and import bindings to transform data from one format to another in a protocol-neutral fashion. Several data handlers are provided as part of the product, but you can also create your own data handler, if necessary. You can associate a data handler with an export or import binding at one of two levels–you can associate it with all operations in the interface of the export or import, or you can associate it with a specific operation for the request or response.

Predefined data handlers

You use WebSphere Integration Developer to specify the data handler that you want to use.

You can specify a data handler for data transformation except in the following cases:

- You are transforming JMS StreamMessage and JMS MapMessage. These types are not supported by data handlers; a custom data binding must be written for them.
- A predefined data binding is provided out of the box but there is no corresponding data handler. For example, a data binding for WrappedBytes is provided but there is no corresponding data handler.

The data handlers that are predefined for your use are listed in the following table, which also describes how each data handler transforms inbound and outbound data.

Note: Except where noted, these data handlers can be used with JMS, Generic JMS, MQ JMS, WebSphere MQ, and HTTP bindings.
See the "Data handlers" topic in the WebSphere Integration Developer information center for more detailed information.

Table 18. Predefined data handlers

Data handler	Native data to business object	Business object to native data
ATOM	Parses ATOM feeds into an ATOM feed business object.	Serializes an ATOM feed business object to ATOM feeds.
Delimited	Parses delimited data into a business object.	Serializes a business object to delimited data, including CSV.
Fixed Width	Parses fixed-width data into a business object.	Serializes a business object to fixed-width data.
Handled by WTX	Delegates data format transformation to the WebSphere Transformation Extender (WTX). The WTX map name is derived by the data handler.	Delegates data format transformation to the WebSphere Transformation Extender (WTX). The WTX map name is derived by the data handler.
Handled by WTX Invoker	Delegates the data format transformation to the WebSphere Transformation Extender (WTX). The WTX map name is supplied by the user.	Delegates the data format transformation to the WebSphere Transformation Extender (WTX). The WTX map name is supplied by the user.
JAXB	Converts Java beans to a business object using the JAXB specification.	Converts a business object to Java beans using the JAXB specification.
JAXWS **Note:** The JAXWS data handler can be used only with the EJB binding.	Transforms the response Java object or exception Java object to a response business object.	Transforms a business object to the outgoing Java method parameters.
JSON	Parses JSON data into a business object.	Serializes a business object to JSON data.
SOAP	Parses the SOAP message (and the header) into a business object.	Serializes a business object to a SOAP message.
XML	Parses XML data into a business object.	Serializes a business object to XML data.

Table 18. Predefined data handlers (continued)

Data handler	Native data to business object	Business object to native data
UTF8XMLDataHandler	Parses UTF-8 encoded XML data into a business object.	Serializes a business object into UTF-8 encoded XML data when sending a message.

Creating a data handler

Detailed information about creating a data handler can be found in the "Developing data handlers" topic in the WebSphere Integration Developer information center.

Data bindings

Data bindings are configured against export and import bindings to transform data from one format to another. Data bindings are specific to a protocol. Several data bindings are provided as part of the product, but you can also create your own data binding, if necessary. You can associate a data binding with an export or import binding at one of two levels–you can associate it with all operations in the interface of the export or import, or you can associate it with a specific operation for the request or response.

You use WebSphere Integration Developer to specify which data binding you want to use or to create your own data binding. A discussion of creating data bindings can be found in the "Overview of JMS, MQ JMS and generic JMS bindings" section of the WebSphere Integration Developer information center.

JMS bindings

The following table lists the data bindings that can be used with:

- JMS bindings
- Generic JMS bindings
- WebSphere MQ JMS bindings

The table also includes a description of the tasks that the data bindings perform.

Table 19. Predefined data bindings for JMS bindings

Data binding	Native data to business object	Business object to native data
Serialized Java object	Transforms the Java serialized object into a business object (which is mapped as the input or output type in the WSDL).	Serializes a business object to the Java serialized object in the JMS object message.
Wrapped bytes	Extracts the bytes from the incoming JMS bytes message and wraps them into the JMSBytesBody business object.	Extracts the bytes from the JMSBytesBody business object and wraps them into the outgoing JMS bytes message

Table 19. Predefined data bindings for JMS bindings (continued)

Data binding	Native data to business object	Business object to native data
Wrapped map entry	Extracts the name, value, and type information for every entry in the incoming JMS map message and creates a list of MapEntry business objects. It then wraps the list into the JMSMapBody business object	Extracts the name, value, and type information from the MapEntry list in the JMSMapBody business object and creates the corresponding entries in the outgoing JMS map message.
Wrapped object	Extracts the object from the incoming JMS object message and wraps it into the JMSObjectBody business object.	Extracts the object from the JMSObjectBody business object and wraps it into the outgoing JMS object message.
Wrapped text	Extracts the text from the incoming JMS text message and wraps it into the JMSTextBody business object.	Extracts the text from the JMSTextBody business object and wraps it into the outgoing JMS text message.

WebSphere MQ bindings

The following table lists the data bindings that can be used with WebSphere MQ and describes the tasks that the data bindings perform.

Table 20. Predefined data bindings for WebSphere MQ bindings

Data binding	Native data to business object	Business object to native data
Serialized Java object	Transforms the Java serialized object from the incoming message into a business object (which is mapped as the input or output type in the WSDL).	Transforms a business object to the Java serialized object in the outgoing message
Wrapped bytes	Extracts the bytes from the unstructured MQ bytes message and wraps them into the JMSTextBody business object.	Extracts the text from a JMSTextBody business object and wraps the text into the outgoing unstructured MQ bytes message.
Wrapped text	Extracts the text from an unstructured MQ text message and wraps it into a JMSTextBody business object.	Extracts text from a JMSTextBody business object and wraps it in an unstructured MQ text message.
Wrapped stream entry	Extracts the name and type information for every entry in the incoming JMS stream message and creates a list of the StreamEntry business objects. It then wraps the list into the JMSStreamBody business object.	Extracts the name and type information from the StreamEntry list in the JMSStreamBody business object and creates corresponding entries in the outgoing JMSStreamMessage.

In addition to the data bindings listed in Table 20 on page 192, WebSphere MQ also uses header data bindings. See the WebSphere Integration Developer information center for details.

HTTP bindings

The following table lists the data binding that can be used with HTTP and describes the tasks that the data binding performs.

Table 21. Predefined data binding for HTTP bindings

Data binding	Native data to business object	Business object to native data
Wrapped bytes	Extracts the bytes from the body of the incoming HTTP message and wraps them into the HTTPBytes business object.	Extracts the bytes from the HTTPBytes business object and adds them to the body of the outgoing HTTP message.

For other data formats, you must employ custom data bindings and data handlers.

Function selectors in export bindings

A function selector is used to indicate which operation should be performed on the data for a request message. Function selectors are configured as part of an export binding.

Consider an SCA export that exposes an interface. The interface contains two operations–Create and Update. The export has a JMS binding that reads from a queue.

When a message arrives on the queue, the export is passed the associated data, but which operation from the export's interface should be invoked on the wired component? The operation is determined by the function selector and the export binding configuration.

The function selector returns the native function name (the function name in the client system that sent the message). The native function name is then mapped to the operation or function name on the interface associated with the export. For example, in the following figure, the function selector returns the native function name (CRT) from the incoming message, the native function name is mapped to the Create operation, and the business object is sent to the SCA component with the Create operation.

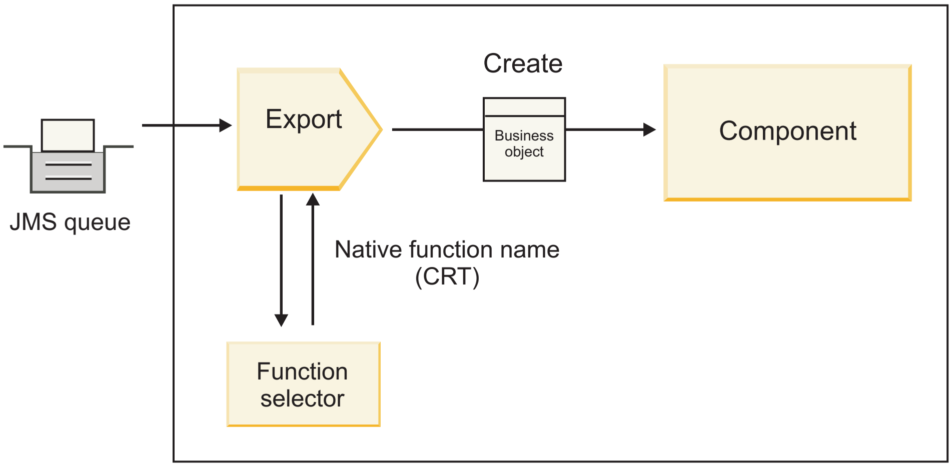


Figure 11. The function selector

If the interface has only one operation, there is no need to specify a function selector.

Several prepackaged function selectors are available and are listed in the sections that follow.

JMS bindings

The following table lists the function selectors that can be used with:

- JMS bindings
- Generic JMS bindings
- WebSphere MQ JMS bindings

Table 22. Predefined function selectors for JMS bindings

Function selector	Description
JMS function selector for simple JMS data bindings	Uses the JMSType property of the message to select the operation name.
JMS header property function selector	Returns the value of the JMS String Property, TargetFunctionName, from the header.
JMS service gateway function selector	Determines if the request is a one-way or two-way operation by examining the JMSReplyTo property set by the client.

WebSphere MQ bindings

The following table lists the function selectors that can be used with WebSphere MQ bindings.

Table 23. Predefined function selectors for WebSphere MQ bindings

Function selector	Description
MQ handleMessage function selector	Returns handleMessage as a value, which is mapped using the export method bindings to the name of an operation on the interface.

Table 23. Predefined function selectors for WebSphere MQ bindings (continued)

Function selector	Description
MQ uses JMS default function selector	Reads the native operation from the TargetFunctionName property of the folder of an MQRFH2 header.
MQ uses message body format as native function	Finds the Format field of the last header and returns that field as a String.
MQ type function selector	Creates a method in your export binding by retrieving a URL containing the Msd, Set, Type and Format properties found in the MQRFH2 header.
MQ service gateway function selector	Uses the MsgType property in the MQMD header to determine the operation name.

HTTP bindings

The following table lists the function selectors that can be used with HTTP bindings.

Table 24. Predefined function selectors for HTTP bindings

Function selector	Description
HTTP function selector based on the TargetFunctionName header	Uses the TargetFunctionName HTTP header property from the client to determine which operation to invoke at runtime from the export.
HTTP function selector based on the URL and HTTP method	Uses the relative path from the URL appended with the HTTP method from the client to determine the native operation defined on the export.
HTTP service gateway function selector based on URL with an operation name	Determines the method to invoke based on the URL if ″operationMode = oneWay″ has been appended to the request URL.

Note: You can also create your own function selector, using WebSphere Integration Developer. Information about creating a function selector is provided in the WebSphere Integration Developer information center. For example, a description of creating a function selector for WebSphere MQ bindings can be found in “Overview of the MQ function selectors”.

Fault handling

You can configure your import and export bindings to handle faults (for example, business exceptions) that occur during processing by specifying fault data handlers. You can set up a fault data handler at three levels–you can associate a fault data handler with a fault, with an operation, or for all operations with a binding.

A fault data handler processes fault data and transforms it into the correct format to be sent by the export or import binding.

- For an export binding, the fault data handler transforms the exception business object sent from the component to a response message that can be used by the client application.

- For an import binding, the fault data handler transforms the fault data or response message sent from a service into an exception business object that can be used by the SCA component.

For import bindings, the binding calls the fault selector, which determines whether the response message is a normal response, a business fault, or a runtime exception.

You can specify a fault data handler for a particular fault, for an operation, and for all operations with a binding.

- If the fault data handler is set at all three levels, the data handler associated with a particular fault is called.
- If fault data handlers are set at the operation and binding levels, the data handler associated with the operation is called.

Two editors are used in WebSphere Integration Developer to specify fault handling. The interface editor is used to indicate whether there will be a fault on an operation. After a binding is generated with this interface, the editor in the properties view lets you configure how the fault will be handled. For more information, see the "Fault selectors" topic in the WebSphere Integration Developer information center.

How faults are handled in export bindings

When a fault occurs during the processing of the request from a client application, the export binding can return the fault information to the client. You configure the export binding to specify how the fault should be processed and returned to the client.

You configure the export binding using WebSphere Integration Developer.

During request processing, a client invokes an export with a request, and the export invokes the SCA component. During the processing of the request, the SCA component can either return a business response or can throw a service business exception or a service runtime exception. When this occurs, the export binding transforms the exception into a fault message and sends it to the client, as shown in the following figure and described in the sections that follow.

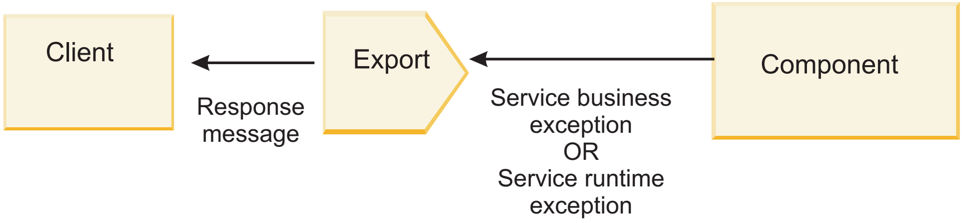


Figure 12. How fault information is sent from the component through the export binding to the client

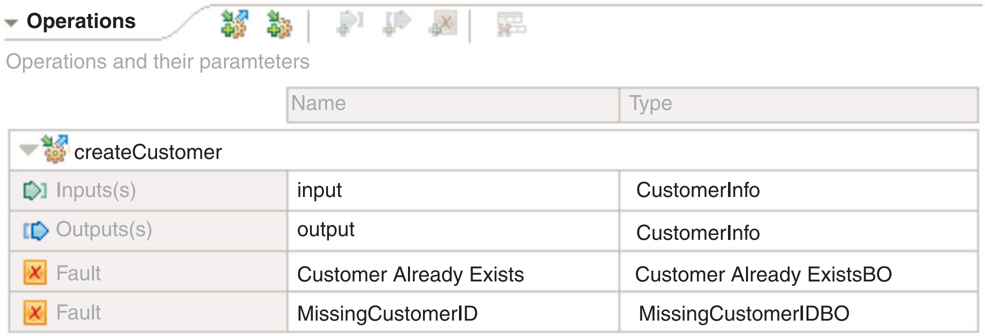
You can create a custom data handler or data binding to handle faults.

Business faults

Business faults are business errors or exceptions that occur during processing.

Consider the following interface, which has a createCustomer operation on it. This operation has two business faults defined: CustomerAlreadyExists and

MissingCustomerId.



Operations

Operations and their paramteters

	Name	Type
createCustomer		
Inputs(s)	input	CustomerInfo
Outputs(s)	output	CustomerInfo
Fault	Customer Already Exists	Customer Already ExistsBO
Fault	MissingCustomerID	MissingCustomerIDBO

Figure 13. Interface with two faults

In this example, if a client sends a request to create a customer (to this SCA component) and that customer already exists, the component throws a CustomerAlreadyExists fault to the export. The export needs to propagate this business fault back to the calling client. To do so, it uses the fault data handler that is set up on the export binding.

When a business fault is received by the export binding, the following processing occurs:

1. The binding determines which fault data handler to invoke for handling the fault. If the service business exception contains the fault name, the data handler that is set up on the fault is called. If the service business exception does not contain the name of the fault, the fault name is derived by matching the fault types.
2. The binding calls the fault data handler with the data object from the service business exception.
3. The fault data handler transforms the fault data object to a response message and returns it to the export binding.
4. The export returns the response message to the client.

If the service business exception contains the fault name, the data handler that is set up on the fault is called. If the service business exception does not contain the name of the fault, the fault name is derived by matching the fault types.

Runtime exceptions

A runtime exception is an exception that occurs in the SCA application during the processing of a request that does not correspond to a business fault. Unlike business faults, runtime exceptions are not defined on the interface.

In certain scenarios, you might want to propagate these runtime exceptions to the client application so that the client application can take the appropriate action.

For example, if a client sends a request (to the SCA component) to create a customer and an authorization error occurs during processing of the request, the component throws a runtime exception. This runtime exception has to be propagated back to the calling client so it can take the appropriate action regarding the authorization. This is achieved by the runtime exception data handler configured on the export binding.

Note: You can configure a runtime exception data handler only on HTTP bindings.

The processing of a runtime exception is similar to the processing of a business fault. If a runtime exception data handler was set up, the following processing occurs:

1. The export binding calls the appropriate data handler with the service runtime exception.
2. The data handler transforms the fault data object to a response message and returns it to the export binding.
3. The export returns the response message to the client.

Fault handling and runtime exception handling are optional. If you do not want to propagate faults or runtime exceptions to the calling client, do not configure the fault data handler or runtime exception data handler.

How faults are handled in import bindings

A component uses an import to send a request to a service outside the module. When a fault occurs during the processing of the request, the service returns the fault to the import binding. You can configure the import binding to specify how the fault should be processed and returned to the component.

You configure the import binding using WebSphere Integration Developer. You can specify a fault data handler (or data binding), and you also specify a fault selector.

Fault data handlers

The service that processes the request sends, to the import binding, fault information in the form of an exception or a response message that contains the fault data.

The import binding transforms the service exception or response message into a service business exception or service runtime exception, as shown in the following figure and described in the sections that follow.

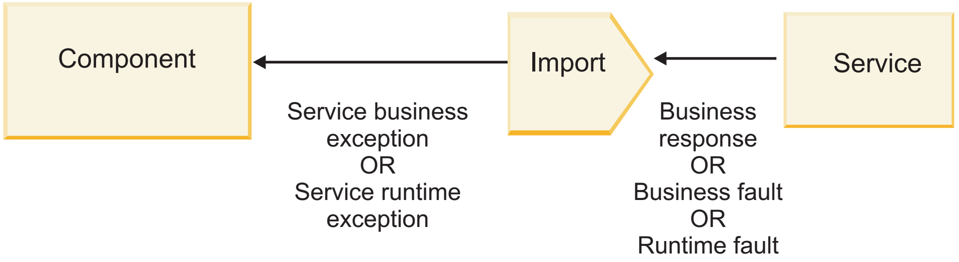

Component
Import
Service
Service business exception OR Service runtime exception
Business response OR Business fault OR Runtime fault

Figure 14. How fault information is sent from the service through the import to the component

You can create a custom data handler or data binding to handle faults.

Fault selectors

When you configure an import binding, you can specify a fault selector. The fault selector determines whether the import response is an actual response, a business exception, or a runtime fault. It also determines, from the response body or header, the native fault name, which is mapped by the binding configuration to the name of a fault in the associated interface.

Two types of prepackaged fault selectors are available for use with JMS, MQ JMS, Generic JMS, WebSphere MQ, and HTTP imports:

Table 25. Prepackaged fault selectors

Fault selector type	Description
Header-based	Determines whether a response message is a business fault, a runtime exception, or a normal message based on the headers in the incoming response message.
SOAP	Determines whether the response SOAP message is a normal response, business fault, or runtime exception.

The following shows examples of header-based fault selectors and the SOAP fault selector.

- Header-based fault selector

  If an application wants to indicate that the incoming message is a business fault, there must be two headers in the incoming message for business faults, which is shown as follows:

  ```
  Header name = FaultType, Header value = Business
  Header name = FaultName, Header value = <user defined native fault name>
  ```

  If an application wants to indicate that the incoming response message is a runtime exception, then there must be one header in the incoming message, which is shown as follows:

  ```
  Header name = FaultType, Header value = Runtime
  ```

- SOAP fault selector

  A business fault can be sent as part of the SOAP message with the following custom SOAP header. ″CustomerAlreadyExists″ is the name of the fault in this case.

  ```
  <ibmSoap:BusinessFaultName
  xmlns:ibmSoap="http://www.ibm.com/soap">CustomerAlreadyExists
  <ibmSoap:BusinessFaultName>
  ```

The fault selector is optional. If you do not specify a fault selector, the import binding cannot determine the type of response. The binding therefore treats it as a business response and calls the response data handler or data binding.

You can create a custom fault selector. The steps for creating a custom fault selector are provided in the “Developing a custom fault selector” topic of the WebSphere Integration Developer information center.

Business faults

A business fault can occur when there is an error in the processing of a request. For example, if you send a request to create a customer and that customer already exists, the service sends a business exception to the import binding.

When a business exception is received by the binding, the processing steps depend on whether a fault selector has been set up for the binding.

- If no fault selector was set up, the binding calls the response data handler or data binding.
- If a fault selector was set up, the following processing occurs:
  1. The import binding calls the fault selector to determine whether the response is business fault, business response, or runtime fault.

2. If the response is a business fault, the import binding calls the fault selector to provide the native fault name.
3. The import binding determines the WSDL fault corresponding to the native fault name returned by the fault selector.
4. The import binding determines the fault data handler that is configured for this WSDL fault.
5. The import binding calls this fault data handler with the fault data.
6. The fault data handler transforms the fault data to a data object and returns it to the import binding.
7. The import binding constructs a service business exception object with the data object and the fault name.
8. The import returns the service business exception object to the component.

Runtime exceptions

A runtime exception can occur when there is a problem in communicating with the service. The processing of a runtime exception is similar to the processing of a business exception. If a fault selector was set up, the following processing occurs:

1. The import binding calls the appropriate runtime exception data handler with the exception data.
2. The runtime exception data handler transforms the exception data to a service runtime exception object and returns it to the import binding.
3. The import returns the service runtime exception object to the component.

EIS bindings

Enterprise information system (EIS) bindings provide connectivity between SCA components and an external EIS. This communication is achieved using EIS exports and EIS imports that support JCA 1.5 resource adapters and Websphere Adapters.

Your SCA components might require that data be transferred to or from an external EIS. When you create an SCA module requiring such connectivity, you will include (in addition to your SCA component) an import or export with an EIS binding for communication with a specific external EIS.

Resource adapters in WebSphere Integration Developer are used within the context of an import or an export. You develop an import or an export with the external service wizard and, in developing it, include the resource adapter. An EIS import, which lets your application invoke a service on an EIS system, or an EIS export, which lets an application on an EIS system invoke a service developed in WebSphere Integration Developer, are created with a particular resource adapter. For example, you would create an import with the JD Edwards adapter to invoke a service on the JD Edwards system.

When you use the external service wizard, the EIS binding information is created for you. You can also use another tool, the assembly editor, to add or modify binding information. See WebSphere Integration Developer Information Center for more information.

After the SCA module containing the EIS binding is deployed to the server, you can use the administrative console to view information about the binding or to configure the binding.

EIS bindings: a general perspective

The EIS (enterprise information system) binding, when used with a JCA resource adapter, lets you access services on an enterprise information system or make your services available to the EIS.

The following example shows how an SCA module named ContactSyncModule synchronizes contact information between a Siebel system and an SAP system.

1. The SCA component named ContactSync listens (by way of an EIS application export named Siebel Contact) for changes to Siebel contacts.
2. The ContactSync SCA component itself makes use of an SAP application (through an EIS application import) in order to update the SAP contact information accordingly.

Because the data structures used for storing contacts are different in Siebel and SAP systems, the ContactSync SCA component must provide mapping.

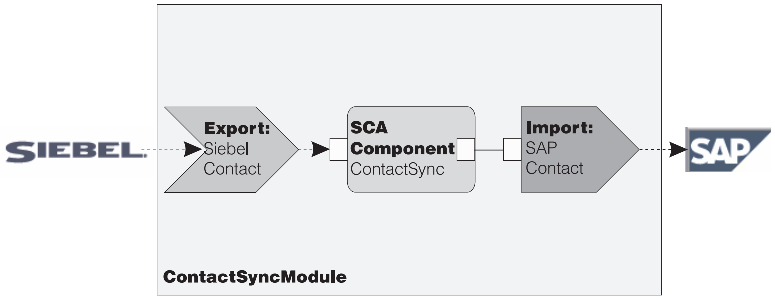


Figure 15. Flow from a Siebel system to an SAP system

The Siebel Contact export and the SAP Contact import have the appropriate resource adapters configured.

Key features of EIS bindings

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

Imports

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

The EIS import provides SCA components with a uniform view of the applications external to the module. This allows components to communicate with an external EIS, such as SAP, Siebel, or PeopleSoft, using a consistent SCA model.

On the client side of the import, there is an interface, exposed by the EIS import application, with one or more methods, each taking data objects as arguments and return values. On the implementation side, there is a Common Client Interface (CCI) implemented by a resource adapter.

The runtime implementation of an EIS import connects the client-side interface and the CCI. The import maps the invocation of the method on the interface to the invocation on the CCI.

Bindings are created at three levels: the interface binding, which then uses the contained method bindings, which in turn use data bindings.

The interface binding relates the interface of the import to the connection to the EIS system providing the application. This reflects the fact that the set of applications, represented by the interface, is provided by the specific instance of the EIS and the connection provides access to this instance. The binding element contains properties with enough information to create the connection (these properties are part of the javax.resource.spi.ManagedConnectionFactory instance).

The method binding associates the method with the specific interaction with the EIS system. For J2C, the interaction is characterized by the set of properties of the javax.resource.cci.InteractionSpec interface implementation. The interaction element of the method binding contains these properties, along with the name of the class, thus providing enough information to perform the interaction. The method binding uses data bindings describing the mapping of the argument and result of the interface method to EIS representation.

The runtime scenario for an EIS import is as follows:

1. The method on the import interface is invoked using the SCA programming model.
2. The request, reaching the EIS import, contains the name of the method and its arguments.
3. The import first creates an interface binding implementation; then, using data from the import binding, it creates a ConnectionFactory and associates the two. That is, the import calls setConnectionFactory on the interface binding.
4. The method binding implementation matching the invoked method is created.
5. 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.
6. The CCI interface is used to perform the interaction.
7. When the call returns, the data binding is used to create the result of the invocation, and the result is returned to the caller.

Exports

The role of the EIS export is to bridge the gap between an SCA component and an external EIS. External applications can be treated as an EIS export. In this case, the external application sends its data in the form of periodic notifications. An EIS export can be thought of as a subscription application listening to an external request from an EIS. The SCA component that utilizes the EIS export views it as a local application.

The EIS export provides SCA components with a uniform view of the applications external to the module. This allows components to communicate with an EIS, such as SAP, Siebel, or PeopleSoft, using a consistent SCA model.

The export features a listener implementation receiving requests from the EIS. The listener implements either the javax.jms.MessageLister interface for the JMS export or a resource adapter-specific listener interface. The export also contains a component implementing interface, exposed to the EIS through the export.

The runtime implementation of an EIS export connects the listener with the component implementing interface. The export maps the EIS request to the invocation of the appropriate operation on the component. Bindings are created at three levels: a listener binding, which then uses a contained native method binding, which in turn uses a 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 by way of the export. There is no relationship 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:

1. The EIS request triggers invocation of the method on the listener implementation.
2. The listener locates and invokes the export, passing to it all the invocation arguments.
3. The export creates the listener binding implementation.
4. The export instantiates the function selector and sets it on the listener binding.
5. The export initializes native method bindings and adds them to the listener binding. For each native method binding, the data bindings are also initialized.
6. The export invokes the listener binding.
7. The listener binding locates exported components and uses the function selector to retrieve the native method name.
8. This name is used to locate the native method binding, which then invokes the target component.

The adapter interaction style allows for the EIS export binding to invoke the target component either asynchronously (the default) or synchronously.

Resource adapters

You develop an import or an export with the external service wizard and, in developing it, include a resource adapter. The adapters that come with WebSphere Integration Developer used to access CICS, IMS, JD Edwards, PeopleSoft, SAP and Siebel systems are intended for development and test purposes only. This means you use them to develop and test your applications.

Once you deploy your application, you will need licensed runtime adapters to run your application. However, when you build your service you can embed the adapter with your service. Your adapter licensing might allow you to use the embedded adapter as the licensed runtime adapter. These adapters are compliant to the Java Connector Architecture (J2C 1.5). J2C, an open standard, is the J2EE standard for EIS connectivity. J2C provides a managed framework; that is Quality of Service (QoS) is provided by the application server, which offers life-cycle management and security to transactions. They are also compliant to the Enterprise Metadata Discovery specification with the exception of IBM CICS ECI Resource Adapter and IBM IMS Connector for Java.

The following IBM WebSphere Adapters are supported in WebSphere Integration Developer:

- IBM CICS ECI Resource Adapter version 7.1.0.2
- IBM IMS TM Resource Adapter version 9.1.0.2
- IBM WebSphere Adapter for Email version 6.2.0
- IBM WebSphere Adapter for FTP version 6.2.0
- IBM WebSphere Adapter for Flat Files version 6.2.0
- IBM WebSphere Adapter for IBM i version 6.2.0
- IBM WebSphere Adapter for JDBC™ version 6.2.0
- IBM WebSphere Adapter for JD Edwards EnterpriseOne® version 6.2.0
- IBM WebSphere Adapter for Oracle® E-Business Suite version 6.2.0
- IBM WebSphere Adapter for PeopleSoft Enterprise version 6.2.0
- IBM WebSphere Adapter for SAP® Software version 6.2.0
- IBM WebSphere Adapter for Siebel® Business Applications Version 6.2.0

The WebSphere Business Integration Adapters, an older set of adapters, is also supported by the wizard.

J2EE resources

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

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

JCA Interaction Spec and Connection Spec dynamic properties

The 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 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:

- 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`.

- The name/value pair represents the name and the value of the simple type of the Interaction Spec property.

In this example, an interface specifies that the input of an operation is an Account data object. This interface invokes an EIS import binding application with the intention to send and receive a dynamic InteractionSpec property called workingSet with the value xyz.

The business graph or business objects in the server contain an underlying properties business object that permits the sending of protocol-specific data with the payload. This properties business object is built-in and does not need to be specified in the XML schema when constructing a business object. It only needs to be created and used. If you have your own data types defined based on an XML schema, then you 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 style 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, set the resAuth specified on the import binding to Application. Also, make sure the resource adapter supports component authorization. See chapter 8 of the J2EE Connector Architecture Specification for more details.

Administering EIS bindings

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

Before you begin

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

About this task

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

To change configuration properties after you deploy the adapter as part of a module, you use the administrative console of the runtime environment. You can update resource adapter properties (used for general adapter operation), managed connection factory properties (used for outbound processing), and activation specification properties (used for inbound processing).

Note: You can also set configuration properties after you install a stand-alone adapter. To do so, from the administrative console, expand **Resources** → **Resource adapters**, and select the adapter whose properties you want to configure.

Procedure

1. View the import that you want to administer. Expand **Applications**, click **SCA Modules**, and click the name of the SCA module that you want to administer.
2. Under the heading **Module Components**, expand the module component for Import or Export, and then expand the import or export that you want to administer. You can choose to administer the interfaces or the bindings.
3. 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.
4. Optional: Administer the binding. Expand **Bindings** and click the import or export binding that you want to administer. You can change the port or the name of the imported or exported service.
5. Save changes to the configuration in order for your changes to take effect.

External clients with EIS bindings

The server can send messages to, or receive messages from, external clients using EIS bindings.

An external client, for example a Web portal or an EIS, needs to send a message to an SCA module in the server or needs to be invoked by a component from within the server.

The client invokes the EIS import as with any other application, using either the Dynamic Invocation Interface (DII) or Java interface.

1. The external client creates an instance of the ServiceManager and looks up the EIS import using its reference name. The result of the lookup is a service interface implementation.

2. The client creates an input argument, a generic data object, created dynamically using the data object schema. This step is done using the Service Data Object DataFactory interface implementation.
3. The external client invokes the EIS and obtains the required results.

Alternatively, the client can invoke the EIS import using the Java interface.

1. The client creates an 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.
2. The client creates an input argument and a typed data object.
3. The client invokes EIS and obtains the required results.

The EIS export interface defines the interface of the exported SCA component that is available to the external EIS applications. This interface can be thought of as the interface that an external application (such as SAP or PeopleSoft) will invoke through the implementation of the EIS export application runtime.

The export uses EISExportBinding to bind exported services to the external EIS application. It allows you to subscribe an application contained in your SCA module to listen for EIS service requests. The EIS export binding specifies the mapping between the definition of inbound events as it is understood by the resource adapter (using J2EE Connector Architecture interfaces) and the invocation of SCA operations.

The EISExportBinding requires external EIS services to be based on J2EE Connector Architecture 1.5 inbound contracts. The EISExportBinding requires that a data handler or data binding be specified either at the binding level or the method level.

Web service bindings

A Web service binding is the means of transmitting messages from a Service Component Architecture (SCA) component to a Web service (and vice versa).

Web service bindings: a general perspective

A Web service import binding allows you to call an external Web service from your Service Component Architecture (SCA) components. A Web service export binding allows you to expose your SCA components to clients as Web services.

With a Web service binding, you access external services using interoperable SOAP messages and qualities of service.

You use WebSphere Integration Developer to generate and configure Web service bindings on imports and exports in SCA modules. A Web service binding can use a transport protocol of either SOAP/HTTP (SOAP over HTTP) or SOAP/JMS (SOAP over JMS).

After the SCA module that contains the Web service binding is deployed to the server, you can use the administrative console to view information about the binding or to change selected properties of the binding.

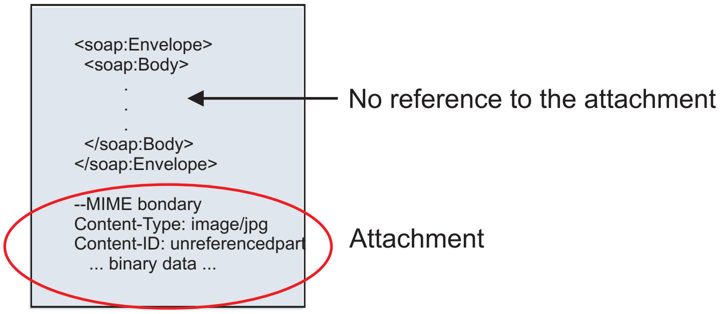
Note: Use of Web Services allows applications to interoperate by using standard descriptions of services and standard formats for the messages they exchange. For example, the Web service import and export bindings can be used to interoperate with services implemented using the Web Services Enhancements (WSE) Version 3.5 and Windows Communication Foundation (WCF) Version 3.5 for Microsoft® .NET. When interoperating with such services, you must ensure that:

- The WSDL file used to access a Web service export includes a non-empty SOAP action value for each operation in the interface.
- The Web service client sets either the SOAPAction header or the wsa:Action header when sending messages to a Web service export.

Unreferenced attachments in SOAP messages

Using one of the Java API for XML Web Services (JAX-WS) bindings, you can send and receive SOAP messages with attachments, such as PDF files or JPG images. You can also use a mediation flow component in an SCA module to process the attachment.

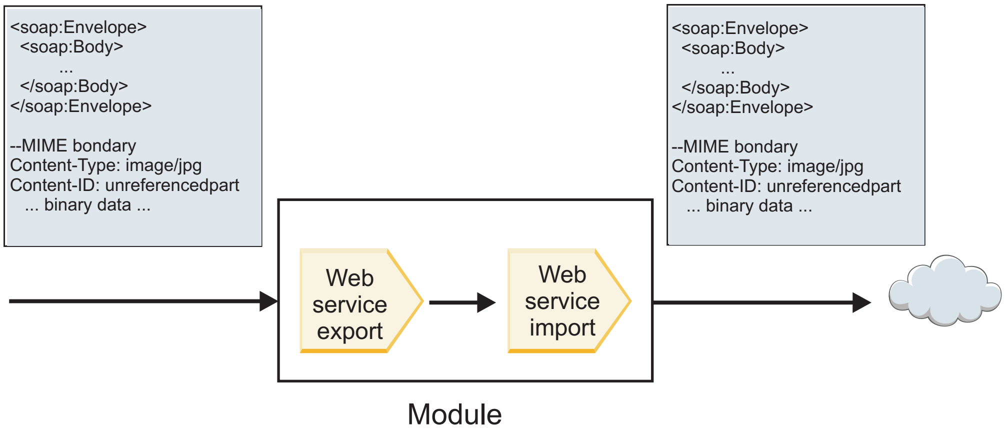
You can send and receive *unreferenced* attachments that are part of a MIME multipart SOAP message. In a MIME multipart SOAP message, the SOAP body is the first part of the message, and the attachments are in subsequent parts. No reference to the attachment is included in the SOAP body.



```
<soap:Envelope>
  <soap:Body>
        .
        .
        .
  </soap:Body>
</soap:Envelope>

--MIME bondary
Content-Type: image/jpg
Content-ID: unreferencedpart
   ... binary data ...
```

Figure 16. A SOAP message with an unreferenced attachment

You can send a SOAP message with an unreferenced attachment through a Web service export to a Web service import. The output message, which is sent to the target Web service, contains the attachment.



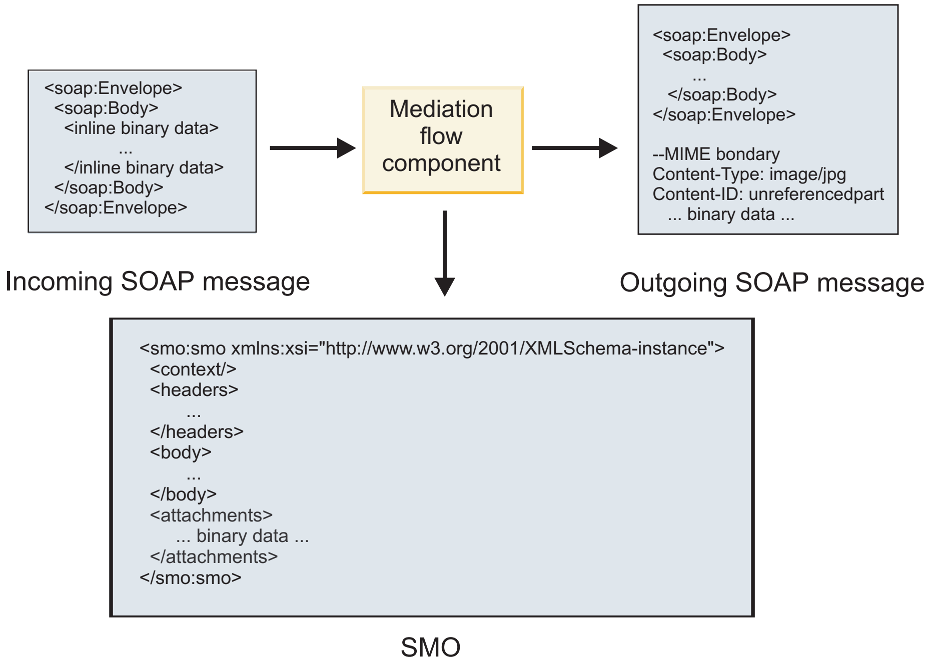
```
<soap:Envelope>
  <soap:Body>
        ...
  </soap:Body>
</soap:Envelope>

--MIME bondary
Content-Type: image/jpg
Content-ID: unreferencedpart
   ... binary data ...
```

```
<soap:Envelope>
  <soap:Body>
        ...
  </soap:Body>
</soap:Envelope>

--MIME bondary
Content-Type: image/jpg
Content-ID: unreferencedpart
   ... binary data ...
```

Figure 17. An attachment passing through an SCA module

In Figure 17, the SOAP message, with the attachment, passes through without modification.

You can also modify the SOAP message by using a mediation flow component. For example, you can use the mediation flow component to extract data from the SOAP message (binary data in the body of the message, in this case) and create a SOAP with attachments message. The data is processed as part of the attachments element of a service message object (SMO).



```
<soap:Envelope>
  <soap:Body>
    <inline binary data>
        ...
    </inline binary data>
  </soap:Body>
</soap:Envelope>
```

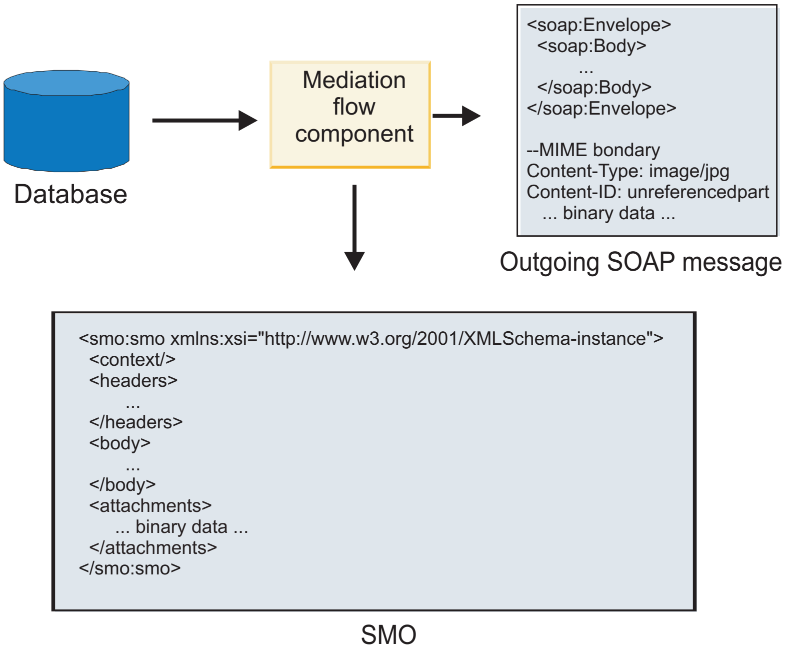
```
<soap:Envelope>
  <soap:Body>
     ...
   </soap:Body>
</soap:Envelope>

--MIME bondary
Content-Type: image/jpg
Content-ID: unreferencedpart
   ... binary data ...
```

```
<smo:smo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <context/>
  <headers>
       ...
  </headers>
  <body>
       ...
  </body>
  <attachments>
      ... binary data ...
  </attachments>
</smo:smo>
```

Figure 18. A message processed by a mediation flow component

Conversely, the mediation flow component can transform the incoming message by extracting and encoding the attachment and then transmitting the message with no attachments.

Instead of extracting data from an incoming SOAP message to form a SOAP with attachments message, you can obtain the attachment data from an external source, such as a database.



```
<soap:Envelope>
  <soap:Body>
    ...
  </soap:Body>
</soap:Envelope>

--MIME bondary
Content-Type: image/jpg
Content-ID: unreferencedpart
   ... binary data ...
```

```
<smo:smo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <context/>
  <headers>
       ...
  </headers>
  <body>
       ...
  </body>
  <attachments>
      ... binary data ...
  </attachments>
</smo:smo>
```

Figure 19. An attachment obtained from a database and added to the SOAP message

Conversely, the mediation flow component can extract the attachment from an incoming SOAP message and process the message (for example, store the attachment in a database).

Unreferenced attachments can be propagated only across mediation flow components. If an attachment must be accessed by or propagated to another component type, use a mediation flow component to move the attachment to a location that is accessible by that component.

Administering Web service bindings

You can view information about the Web service import and export bindings of a module after the module has been deployed to the server. You can also reconfigure selected properties of those bindings and configure policy sets for the bindings.

Administering Web service import bindings

Using the administrative console, you can view information about a Web service import binding, change the endpoint URL, and, for the Java API for XML Web Services (JAX-WS) bindings, configure a policy set for the binding.

Before you begin

To perform this task, you must have permission to change the master configuration.

About this task

A *policy set* is a collection of policy types, each of which provides a quality of service. These types have been configured and can be associated with a Web service provider or consumer.

Policy sets work in pairs. You must have the same policy set on the service requester as on the service provider. Therefore, you should have the same policy set on the import binding as on the service provider it is calling.

Note that you can configure policy sets only for JAX-WS Web service bindings. The **Policy set attachments** section of the page does not appear for JAX-RPC service bindings.

To view information about a Web service import binding or to change properties of the associated module, use the administrative console to complete the following steps.

Procedure

1. Select the module that contains the Web service import binding by navigating to **Applications** → **SCA modules** and clicking the module name.
2. Select the binding by performing the following steps:
   a. In the **Module components** section, expand **Imports**.
   b. Expand the import, and then expand **Binding**.
   c. Click the Web service binding to view information about the binding.
3. Change the value of **Target endpoint address**, which is the location of the Web service, and then click **Apply** or **OK**.
4. To configure policy sets for import bindings and associated operations:
   a. Optional: Expand **Preferences**, indicate the maximum number of rows and whether you want to retain the filter criteria, and click **Apply**.
   b. Optional: Select the filter icon if you want to use a filter to search the table.
   c. Select the import binding or operation, and click **Attach** to attach a policy set to the binding or operation, or click **Detach** to remove the policy set.
   d. To assign a new policy set binding, select the import binding or operation, click **Assign Binding**, and provide a name for the policy set binding.
   e. Repeat steps 4c and 4d for each binding or operation you want to configure.
   f. Save the changes to the master configuration.

Results

The Web service import binding is changed for the selected Service Component Architecture (SCA) module. The changes take effect after you update the master configuration and restart the SCA module.

Administering Web service export bindings

Using the administrative console, you can view information about a Web service export binding (including the WSDL file), configure properties of the associated module, and, for the Java API for XML Web Services (JAX-WS) bindings, configure a policy set for the binding.

Before you begin

To perform this task, you must have permission to change the master configuration.

About this task

A *policy set* is a collection of policy types, each of which provides a quality of service. These types have been configured and can be associated with a Web service provider or consumer.

Policy sets work in pairs. You must have the same policy set on the service requester as on the service provider. Therefore, you should have the same policy set on the export binding as on the client.

Note that you can configure policy sets only for JAX-WS Web service bindings. The **Policy set attachments** section of the page does not appear for JAX-RPC Web service bindings.

To view information about a Web service export binding or to change properties of the associated module, use the administrative console to complete the following steps.

Procedure

1. Select the module that contains the Web service export binding by navigating to **Applications** → **SCA modules** and clicking the module name.
2. Select the binding by performing the following steps:
   a. In the **Module components** section, expand **Exports**.
   b. Expand the export, and then expand **Binding**.
   c. Click the Web service binding to view information about the binding.
      - In the **General Properties** section, view the name, port, and location (endpoint address) of the Web service.
      - From the **Related Properties** list, click the interface to view the Web Services Description Language (WSDL) file that is associated with the Web service.
3. To change properties that are associated with the module, click one of the following properties in the **Web Module Properties** list:
   - Click **Manage Export Binding Web Module** to view or edit deployment-specific information for a Web module. For example, you can edit the **Starting weight**, which specifies the priority of this module during server startup.
   - Click **Context Root** to view the Web module name and Uniform Resource Identifier (URI) and edit the context root.
   - Click **Virtual Hosts** to specify the virtual host for the Web module. Virtual hosts let you associate a unique port with a module or application.
   - Click **JSP reload options for Web modules** to specify information about the reloading of JavaServer Pages (JSP) files (such as the number of seconds to scan a file system for updated JSP files).
   - Click **Session management** to specify information about HTTP session support. For example, you can set the number of minutes before a session times out.
4. To configure policy sets for export bindings and associated operations:
   a. Optional: Expand **Preferences**, indicate the maximum number of rows and whether you want to retain the filter criteria, and click **Apply**.
   b. Optional: Select the filter icon if you want to use a filter to search the table.
   c. Select the export binding or operation, and click **Attach** to attach a policy set to the binding or operation, or click **Detach** to remove the policy set.

d. To assign a new policy set binding, select the export binding or operation, click **Assign Binding**, and provide a name for the policy set binding.

e. Repeat steps 4c on page 212 and 4d for each binding or operation you want to configure.

f. Save the changes to the master configuration.

Results

If you made any changes to the Web module properties, those changes take effect after you update the master configuration and restart the SCA module.

JMS bindings

A Java Message Service (JMS) provider enables messaging based on the Java Messaging Service API and programming model. It provides J2EE connection factories to create connections for JMS destinations and to send and receive messages.

Three JMS bindings are provided:

- Service integration bus (SIB) provider binding compliant with JMS JCA 1.5 (*JMS binding*)
- Generic, non-JCA JMS binding, compliant with JMS 1.1 (*Generic JMS binding*)
- WebSphere MQ JMS binding, providing JMS provider support for WebSphere MQ and allowing J2EE application interoperability (*WebSphere MQ JMS binding*)

The JMS export and import bindings allow a Service Component Architecture (SCA) module to make calls to, and receive messages from, external JMS systems.

Also supported are WebSphere MQ bindings (*WebSphere MQ binding*) which allow native MQ users to handle arbitrary incoming and outgoing message formats (WebSphere MQ required).

The JMS import and export bindings provide integration with JMS applications using the JCA 1.5-based SIB JMS provider that is part of WebSphere Application Server. Other JCA 1.5–based JMS resource adapters are not supported

JMS bindings: a general perspective

JMS bindings provide connectivity between the Service Component Architecture (SCA) environment and JMS systems.

JMS bindings

The major components of both JMS import and JMS export bindings are:

- Resource adapter: enables managed, bidirectional connectivity between an SCA module and external JMS systems
- Connections: encapsulate a virtual connection between a client and a provider application
- Destinations: used by a client to specify the target of messages it produces or the source of messages it consumes
- Authentication data: used to secure access to the binding

JMS import bindings

JMS import bindings allow you to import an external JMS application to be used inside your SCA module. JMS import bindings allow components within your SCA module to communicate with services provided by external JMS applications.

Connections to the associated JMS provider of JMS destinations are created by using a JMS connection factory. Use connection factory administrative objects to manage JMS connection factories for the default messaging provider.

Interaction with external JMS systems includes the use of destinations for sending requests and receiving replies.

Two types of usage scenarios for the JMS import binding are supported, depending on the type of operation being invoked:

- One-way: The JMS import puts a message on the send destination configured in the import binding. Nothing is set in the replyTo field of the JMS header.
- Two-way (request-response): The JMS import puts a message on the send destination and then persists the reply it receives from the SCA component.

  The receive destination is set in the replyTo header property of the outbound message. A message driven bean (MDB) is deployed to listen on the receive destination, and when a reply is received, the MDB passes the reply back to the component.

  The import binding can be configured (using the **Response correlation scheme** field in WebSphere Integration Developer) to expect the response message correlation ID to have been copied from the request message ID (the default) or from the request message correlation ID.

For both one-way and two-way usage scenarios, dynamic and static header properties can be specified. Static properties can be set from the JMS import method binding. Some of these properties have special meanings to the SCA JMS runtime.

It is important to note that JMS is an asynchronous binding. If a calling component invokes a JMS import synchronously (for a two-way operation), the calling component is blocked until the response is returned by the JMS service.

Figure 20 on page 215 illustrates how the import is linked to the external service.

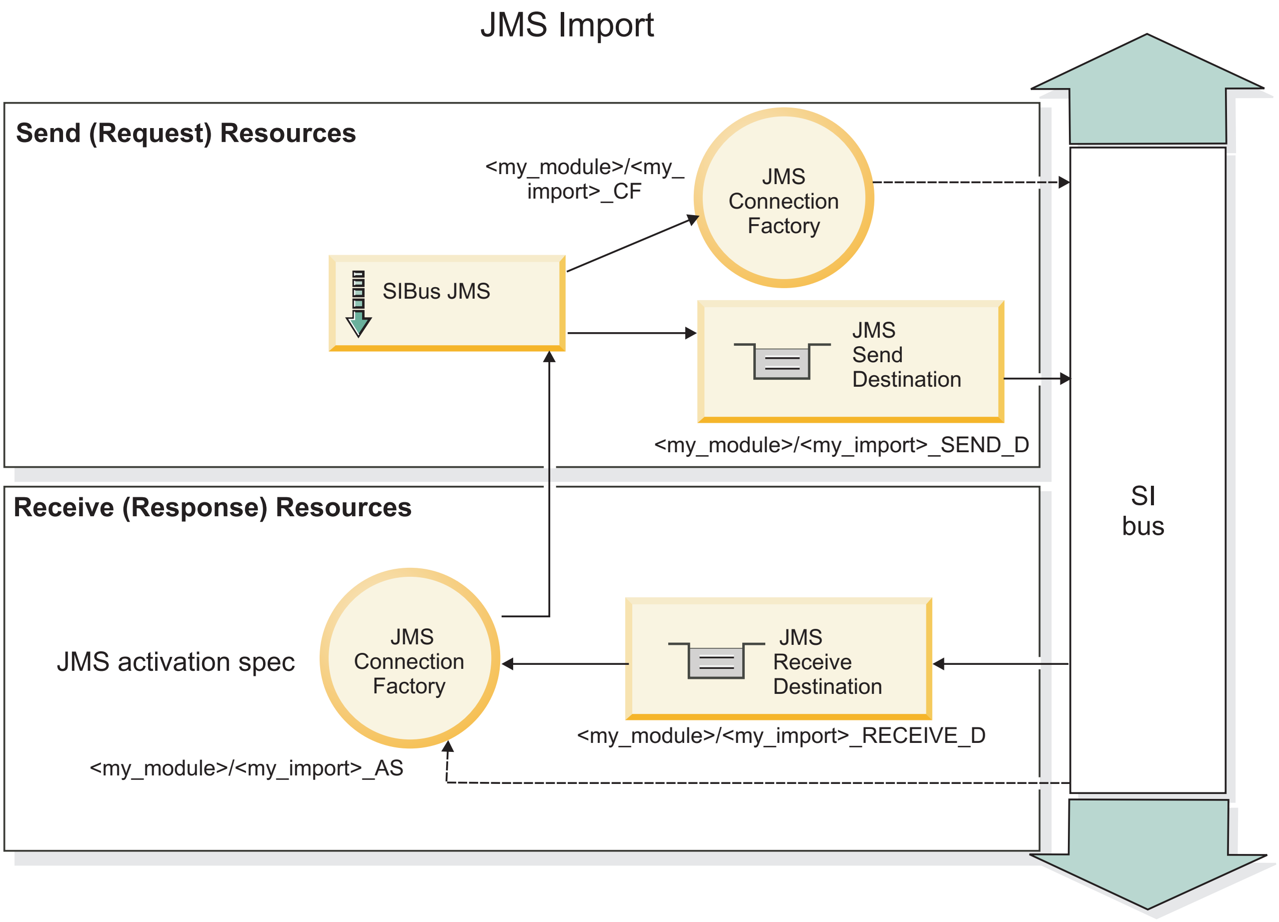


Figure 20. JMS import binding resources

JMS export bindings

JMS export bindings provide the means for SCA modules to provide services to external JMS applications.

The connection that is part of a JMS export is a configurable activation specification.

A JMS export has send and receive destinations.

- The receive destination is where the incoming message for the target component should be placed.
- The send destination is where the reply will be sent, unless the incoming message has overridden this using the replyTo header property.

An MDB is deployed to listen to requests incoming to the receive destination specified in the export binding. The destination specified in the send field is used to send the reply to the inbound request if the invoked component provides a reply. The destination specified in the replyTo field of the incoming message overrides the destination specified in the send.

Figure 21 on page 216 illustrates how the external requestor is linked to the export.

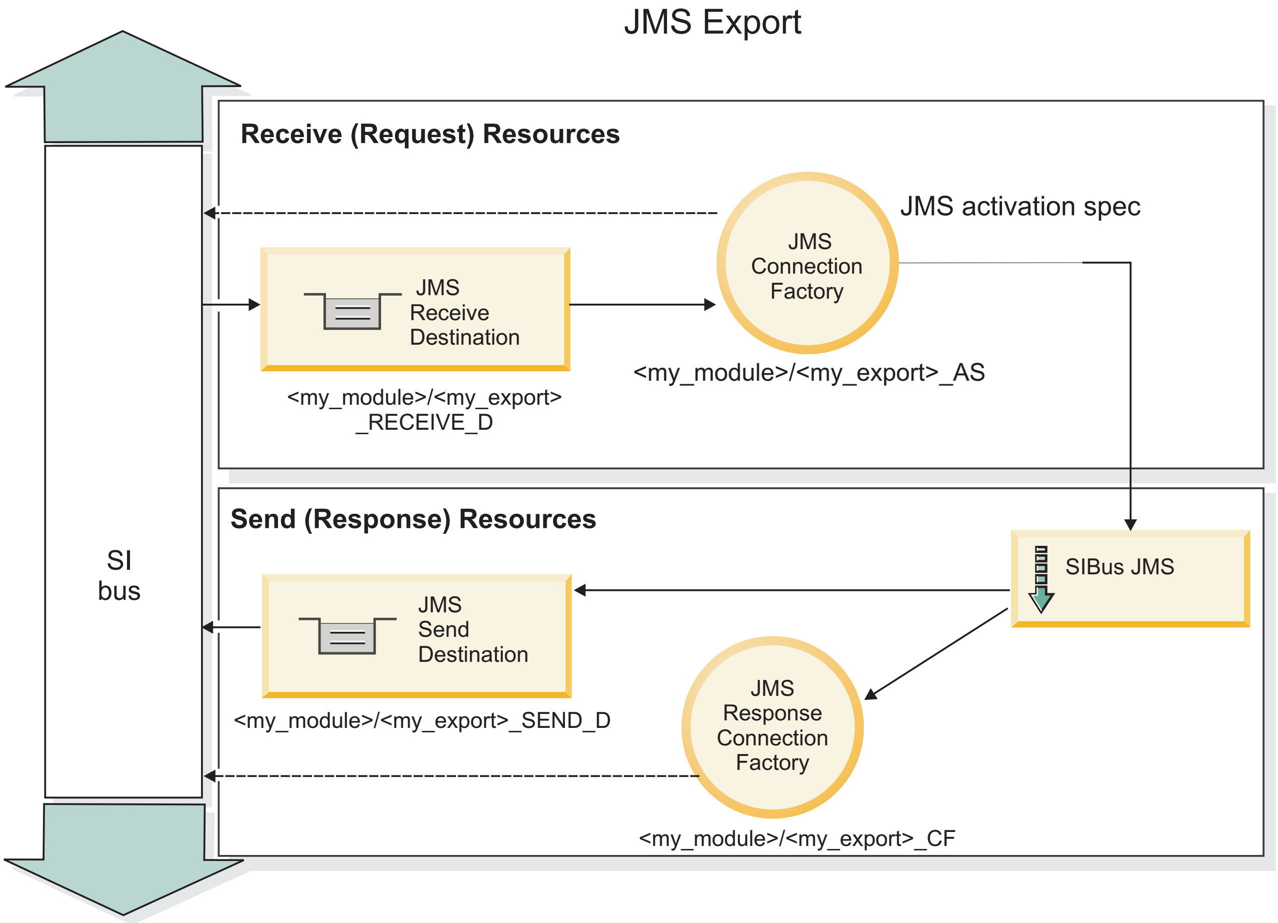

JMS Export
Receive (Request) Resources
JMS activation spec
JMS Connection Factory
JMS Receive Destination
<my_module>/<my_export>_AS
<my_module>/<my_export> _RECEIVE_D
SI bus
Send (Response) Resources
SIBus JMS
JMS Send Destination
<my_module>/<my_export>_SEND_D
JMS Response Connection Factory
<my_module>/<my_export>_CF

Figure 21. JMS export binding resources

JMS integration and resource adapters

The Java Message Service (JMS) provides integration through an available JMS JCA 1.5-based resource adapter. Complete support for JMS integration is provided for the Service Integration Bus (SIB) JMS resource adapter.

Use a JMS provider for JCA 1.5 resource adapter when you want to integrate with an external JCA 1.5-compliant JMS system. External services compliant with JCA 1.5 can receive messages and send messages to integrate with your service component architecture (SCA) components using the SIB JMS resource adapter.

The use of other provider-specific JCA 1.5 resource adapters is not supported.

JMS modules cannot be deployed to a J2SE environment. Such modules are only deployable to a J2EE environment.

Key features of JMS bindings

Key features of JMS import and export bindings include headers and created J2EE resources.

Special headers

Special header properties are used in JMS imports and exports to tell the target how to handle the message.

For example, TargetFunctionName maps from the native method to the operation method.

J2EE resources

A number of J2EE resources are created when JMS imports and exports are deployed to a J2EE environment.

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

ActivationSpec
Imports use this for receiving the response to a request; exports use it when configuring the message endpoints that represent MDBs in their interactions with the messaging system.

Destinations

- Send destination: on an import, this is where the request or outgoing message is sent; on an export, this is the destination where the response message will be sent, if not superseded by the JMSReplyTo header field in the incoming message.
- Receive destination: where the incoming message should be placed; with imports, this is a response; with exports, this is a request.
- Callback destination: SCA JMS system destination used to store correlation information. Do not read or write to this destination.

The installation task creates the ConnectionFactory and three destinations. It also creates the ActivationSpec to enable the runtime MDB to listen for replies on the receive destination. The properties of these resources are specified in the import or export file.

Administering JMS bindings

JMS import and export bindings can be administered from within the server.

Use the administrative console to configure and administer JMS import and export bindings.

Detailed instructions on creating JMS import and export bindings can be found in the WebSphere Integration Developer information center at **WebSphere Integration Developer** > **Developing integration applications** > **Accessing external services with messaging systems > Java Message Service (JMS)** > **Working with JMS bindings**.

Properties of JMS bindings

The JMS import and export bindings can be installed with all the necessary connection factories having been created during deployment, or they can be configured to point to a set of existing resources.

Typically, JMS import and export bindings are created in WebSphere Integration Developer. When you configure the binding, you can either create the connections and destinations required for the JMS binding (by selecting **Configure new**

messaging provider resources, which is the default), or you can select **Use pre-configured messaging provider resources**. If you choose pre-configured, you add the JNDI names for the connection factory and the send destination (for a one-way operation) or the send and receive destinations (for a request-response operation).

Configuring the JMS binding depends upon which option was selected.

The following list shows examples of the resources you specify when you select **Use pre-configured messaging provider resources**. In these examples, the module name is `Inventory`.

Table 26. Example values for import bindings

Property	Example
JNDI name for connection factory	`Inventory/Inventory_CF`
JNDI name for send destination	`Inventory/Inventory_SEND_D`
JNDI name for receive destination	`Inventory/Inventory_RECEIVE_D`

Table 27. Example values for export bindings

Property	Example
JNDI name for activation specification	`Inventory/Inventory_AS`
JNDI name for receive destination	`Inventory/Inventory_RECEIVE_D`
JNDI name for send destination	`Inventory/Inventory_SEND_D`

Note: The resources are created at the server scope. The scope in the administrative console is initially set to **All Scopes**. You must set the scope to **cell** or **node** to create a new resource. You can select an existing resource from the default list.

Configuring JMS bindings

You can configure JMS import and export bindings to apply special features of the resource. The administrative tasks are performed using the WebSphere administrative console.

Before you begin

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

About this task

The JMS binding must be installed as part of a Service Component Architecture (SCA) application in your server profile.

Procedure

1. Select the SCA module. From the administrative console, click **Applications** → **SCA Modules** and then click the *modulename*. The configuration page opens.
2. Under **Module components**, expand **Imports** or **Exports**. This reveals a list of the installed imports or exports.
3. Expand the import or export and then expand **Binding**. This reveals a list of available bindings to administer.

4. Click on the binding to be administered. The general properties of the binding are displayed:
   - The **Send Resources** category contains the Connection Factory and the Send Destination.
   - The **Receive Resources** category contains the Response Connection Factory and the Activation Specification.
   - The **Advanced Resources** category contains Callback resources and other available resources.

   Note: You can also access a resource by typing the JNDI name in the text box. Doing so, however, allows you to enter the name of a resource that is not yet configured.
5. Administer the desired resource:
   a. Click **Browse** to open a window with a list of JNDI names; then, select the desired JNDI name.
   b. Click **Configure** to open the corresponding panel referred to by the JNDI name.

      When **Configure** has been selected, the corresponding server panel opens.
6. When you are ready to administer your settings, click **Apply** or **OK**.

What to do next

The application will be automatically restarted for the JNDI bindings to take effect.

JMS headers

A JMS message contains two types of headers–the JMS system header and multiple JMS properties. Both types of headers can be accessed either in a mediation module in the Service Message Object (SMO) or by using the ContextService API.

JMS system header

The JMS system header is represented in the SMO by the JMSHeader element, which contains all the fields typically found in a JMS header. Although these can be modified in the mediation (or ContextService), some JMS system header fields set in the SMO will not be propagated in the outbound JMS message as they are overridden by system or static values.

The key fields in the JMS system header that can be updated in a mediation (or ContextService) are:

- **JMSType** and **JMSCorrelationID** – values of the specific predefined message header properties
- **JMSDeliveryMode** – values for delivery mode (persistent or nonpersistent; default is persistent)
- **JMSPriority** – priority value (0 to 9; default is JMS_Default_Priority)

JMS properties

JMS properties are represented in the SMO as entries in the Properties list. The properties can be added, updated, or deleted in a mediation or by using the ContextService API.

Properties can also be set statically in the JMS binding. Properties that are set statically override settings (with the same name) that are set dynamically.

User properties propagated from other bindings (for example, an HTTP binding) will be output in the JMS binding as JMS properties.

Header propagation settings

The propagation of the JMS system header and properties either from the inbound JMS message to downstream components or from upstream components to the outbound JMS message can be controlled by the Propagate Protocol Header flag on the binding.

When Propagate Protocol Header is set, header information is allowed to flow to the message or to the target component, as described in the following list:

- JMS export request

  The JMS header received in the message will be propagated to target components by way of the context service. JMS properties received in the message will be propagated to target components by way of the context service.
- JMS export response

  Any JMS header fields set in the context service will be used in the outbound message, if not overridden by static properties set on the JMS export binding. Any properties set in the context service will be used in the outbound message if not overridden by static properties set on the JMS export binding.
- JMS import request

  Any JMS header fields set in the context service will be used in the outbound message, if not overridden by static properties set on the JMS import binding. Any properties set in the context service will be used in the outbound message if not overridden by static properties set on the JMS import binding.
- JMS import response

  The JMS header received in the message will be propagated to target components by way of the context service. JMS properties received in the message will be propagated to target components by way of the context service.

External clients

The server can send messages to, or receive messages from, external clients using JMS bindings.

An external client (such as a Web portal or an enterprise information system) can send a message to an SCA module in the server, or it can be invoked by a component from within the server.

The JMS export components deploy message driven beans (MDBs) to listen to requests incoming to the receive destination specified in the export binding. The destination specified in the send field is used to send the reply to the inbound request if the invoked application provides a reply. Thus, an external client is able to invoke applications with the export binding.

JMS imports bind to, and can deliver messages to, external clients. This message might or might not demand a response from the external client.

Working with external clients

An external client (that is, outside the server) might need to interact with an application installed in the server.

About this task

Consider a very simple scenario in which an external client wants to interact with an application on the server. The figure depicts a typical simple scenario.

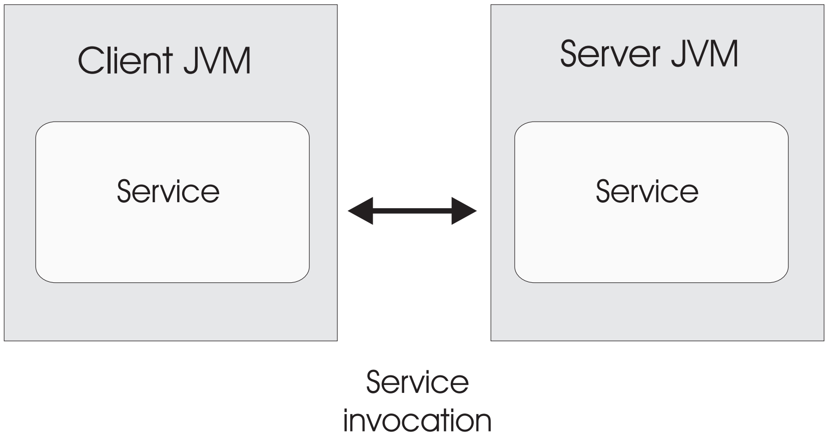


Figure 22. Simple use-case scenario: external client interacts with server application

The SCA application includes an export with a JMS binding; this makes the application available to external clients.

When you have an external client in a Java virtual machine (JVM) separate from your server, there are several steps you must take in order to make a connection and interact with a JMS export. The client obtains an InitialContext with the correct values and then looks up the resources through JNDI. The client then uses the JMS 1.1 specification client to access the destinations and the send and receive messages on the destinations.

The default JNDI names of the resources created automatically by the runtime are listed in the configuration topic of this section. However, if you have pre-created resources, use those JNDI names.

Procedure

1. Configure JMS destinations and the connection factory to send the message.
2. Make sure that the JNDI context, the port for the SIB resource adapter, and the messaging bootstrapping port are correct.

   The server uses some default ports, but if there are more servers installed on that system, alternate ports are created at installation time to avoid conflicts with other server instances. You can use the administrative console to determine which ports your server is employing . Go to **Servers** → **Application Servers** → *your_server_ name* → **Configuration** and click **Ports** under **Communication**. You can then edit the port being used.
3. The client obtains an initial context with the correct values and then looks up the resources through JNDI.
4. Using JMS 1.1 specifications, the client accesses the destinations and the send and receive messages on the destinations.

Troubleshooting JMS bindings

You can diagnose and fix problems with JMS bindings.

Implementation exceptions

In response to various error conditions, the JMS import and export implementation can return one of two types of exceptions:

- Service Business Exception: this exception is returned if the fault specified on the service business interface (WSDL port type) occurred.
- Service Runtime Exception: raised in all other cases. In most cases, the cause exception will contain the original exception (JMSException).

  For example, an import expects only one response message for each request message. If more than one response arrives, or if a late response (one for which the SCA response expiration has expired) arrives, a Service Runtime Exception is thrown. The transaction is rolled back, and the response message is backed out of the queue or handled by the failed event manager.

Primary failure conditions

The primary failure conditions of JMS bindings are determined by transactional semantics, by JMS provider configuration, or by reference to existing behavior in other components. The primary failure conditions include:

- Failure to connect to the JMS provider or destination.

  A failure to connect to the JMS provider to receive messages will result in the MDB Listener Port failing to start. This condition will be logged in the WebSphere Application Server log. Persistent messages will remain on the destination until they are successfully retrieved (or expired).

  A failure to connect to the JMS provider to send outbound messages will cause rollback of the transaction controlling the send.
- Failure to parse an inbound message or to construct an outbound message.

  A failure in the data binding or data handler causes rollback of the transaction controlling the work.
- Failure to send the outbound message.

  A failure to send a message causes rollback of the relevant transaction.
- Multiple or unexpected late response messages.

  The import expects only one response message for each request message. Also the valid time period in which a response can be received in determined by the SCA Response Expiration qualifier on the request. When a response arrives or the expiration time is exceeded, the correlation record is deleted. If response messages arrive unexpectedly or arrive late, a Service Runtime Exception is thrown.

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.

Handling exceptions

The way in which the binding is configured determines how exceptions that are raised by data handlers or data bindings are handled. Additionally, the nature of the mediation flow dictates the behavior of the system when such an exception is thrown.

A variety of problems can occur when a data handler or data binding is called by your binding. For example, a data handler might receive a message that has a corrupt payload, or it might try to read a message that has an incorrect format.

The way your binding handles such an exception is determined by how you implement the data handler or data binding. The recommended behavior is that you design your data binding to throw a `DataBindingException`.

When any runtime exception, including a `DataBindingException`, is thrown:

- If the mediation flow is configured to be transactional, the JMS message , by default, is stored in the Failed Event Manager for manual replay or deletion.

  Note: You can change the recovery mode on the binding so that the message is rolled back instead of being stored in the Failed Event Manager.
- If the mediation flow is not transactional, the exception is logged and the message is lost.

The situation is similar for a data handler. Since the data handler is invoked by the data binding, any data handler exception is wrapped into a data binding exception. Therefore a `DataHandlerException` is reported to you as a `DataBindingException`.

Generic JMS bindings

The Generic JMS binding provides connectivity to third-party JMS 1.1 compliant providers. The operation of the Generic JMS bindings is similar to that of JMS bindings.

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

The Generic JMS binding provides integration with non-JCA 1.5-compliant JMS providers that support JMS 1.1 and implement the optional JMS Application Server Facility. The Generic JMS binding supports those JMS providers (including Oracle AQ, TIBCO, SonicMQ, WebMethods, BEA WebLogic, and WebSphere MQ) that do not support JCA 1.5 but do support the Application Server Facility of the JMS 1.1 specification. SIB, which is a JCA 1.5 JMS provider, is not supported.

Use this Generic binding when integrating with a non-JCA 1.5-compliant JMS-based system within an SCA environment. The target external applications can then receive messages and send messages to integrate with an SCA component.

Generic JMS bindings: a general perspective

Generic JMS bindings are non-JCA JMS bindings that provide connectivity between the Service Component Architecture (SCA) environment and JMS systems that are compliant with JMS 1.1 and that implement the optional JMS Application Server Facility.

Generic JMS bindings

The major aspects of Generic JMS import and export bindings are:

- Resource adapter: enables managed, bidirectional connectivity between enterprise information systems (EISs) and J2EE components
- Connections: encapsulate a virtual connection between a client and a provider application
- Destinations: used by a client to specify the target of messages it produces or the source of messages it consumes
- Authentication data: used to secure access to the binding

Generic JMS import bindings

Generic JMS import bindings allow components within your SCA module to communicate with services provided by external non-JCA 1.5-compliant JMS providers.

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

Interaction with external JMS systems includes the use of destinations for sending requests and receiving replies.

Two types of usage scenarios for the Generic JMS import binding are supported, depending on the type of operation being invoked:

- One-way: The Generic JMS import puts a message on the send destination configured in the import binding Nothing is sent to the replyTo field of the JMS header.
- Two-way (request-response): The Generic JMS import puts a message on the send destination and then persists the reply it receives from the SCA component.

  The receive destination is set in the replyTo header property of the outbound message. A message driven bean (MDB) is deployed to listen on the receive destination, and when a reply is received, the MDB passes the reply back to the component.

  The import binding can be configured (using the **Response correlation scheme** field in WebSphere Integration Developer) to expect the response message correlation ID to have been copied from the request message ID (the default) or from the request message correlation ID.

For both one-way and two-way usage scenarios, dynamic and static header properties can be specified. Static properties can be set from the Generic JMS import method binding Some of these properties have special meanings to the SCA JMS runtime.

It is important to note that Generic JMS is an asynchronous binding. If a calling component invokes a Generic JMS import synchronously (for a two-way operation), the calling component is blocked until the response is returned by the JMS service.

Figure 23 on page 225 illustrates how the import is linked to the external service.

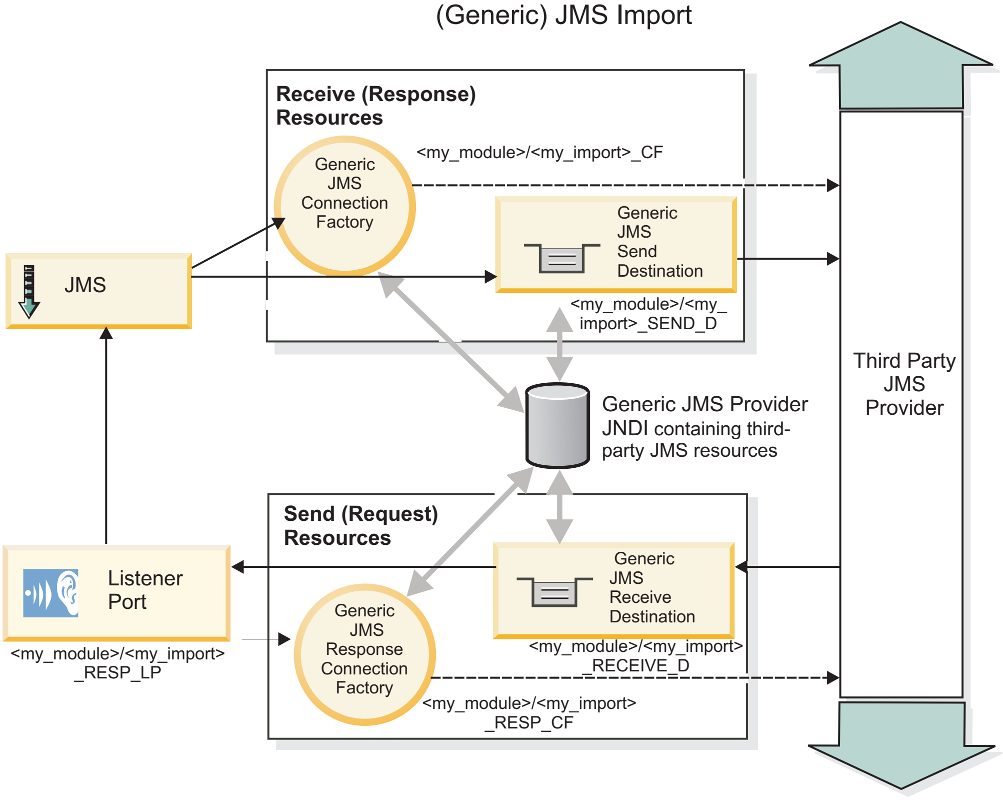

(Generic) JMS Import
Receive (Response) Resources
Generic JMS Connection Factory
<my_module>/<my_import>_CF
Generic JMS Send Destination
<my_module>/<my_import>_SEND_D
JMS
Generic JMS Provider JNDI containing third-party JMS resources
Third Party JMS Provider
Send (Request) Resources
Listener Port
<my_module>/<my_import>_RESP_LP
Generic JMS Receive Destination
<my_module>/<my_import>_RECEIVE_D
Generic JMS Response Connection Factory
<my_module>/<my_import>_RESP_CF

Figure 23. Generic JMS import binding resources

Generic JMS export bindings

Generic JMS export bindings provide the means for SCA modules to provide services to external JMS applications.

The connection part of a JMS export is composed of a ConnectionFactory and a ListenerPort.

A Generic JMS export has send and receive destinations.

- The receive destination is where the incoming message for the target component should be placed.
- The send destination is where the reply will be sent, unless the incoming message has overridden this using the replyTo header property.

An MDB is deployed to listen to requests incoming to the receive destination specified in the export binding.

- The destination specified in the send field is used to send the reply to the inbound request if the invoked component provides a reply.
- The destination specified in the replyTo field of the incoming message overrides the destination specified in the send field.
- For request/response scenarios, the import binding can be configured (using the **Response correlation scheme** field in WebSphere Integration Developer) to expect the response to copy the request message ID to the correlation ID field of the response message (default), or the response can copy the request correlation ID to the correlation ID field of the response message.

Figure 24 illustrates how the external requestor is linked to the export.

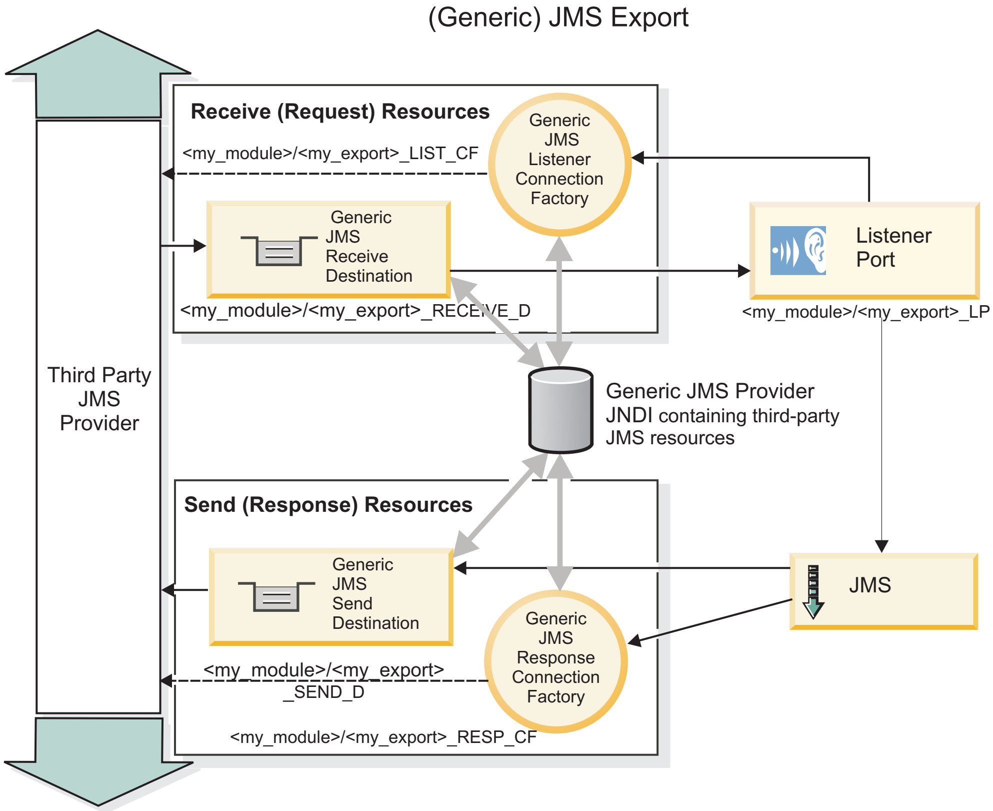

(Generic) JMS Export
Third Party JMS Provider
Receive (Request) Resources
<my_module>/<my_export>_LIST_CF
Generic JMS Listener Connection Factory
Generic JMS Receive Destination
<my_module>/<my_export>_RECEIVE_D
Listener Port
<my_module>/<my_export>_LP
Generic JMS Provider JNDI containing third-party JMS resources
Send (Response) Resources
Generic JMS Send Destination
<my_module>/<my_export> _SEND_D
Generic JMS Response Connection Factory
<my_module>/<my_export>_RESP_CF
JMS

Figure 24. Generic JMS export binding resources

Key features of Generic JMS bindings

The features of the Generic JMS import and export binding are consistent with those of the WebSphere embedded JMS and MQ JMS import bindings. Key features include header definitions and access to existing J2EE resources. However, because of its generic nature, there are no JMS provider-specific connectivity options, and this binding has limited capability to generate resources at deployment and installation.

Generic imports

Like the MQ JMS import application, the Generic JMS implementation is asynchronous and supports three invocations: one-way, two-way (also known as request-response), and callback.

When the JMS import is deployed, a message driven bean (MDB) provided by the runtime environment is deployed. The MDB listens for replies to the request message. The MDB is associated with (listens on) the destination sent with the request in the replyTo header field of the JMS message.

Generic exports

Generic JMS export bindings differ from EIS export bindings in their handling of the return of the result. A Generic JMS export explicitly sends the response to the replyTo destination specified on the incoming message. If none is specified, the send destination is used.

When the Generic JMS export is deployed, a message driven bean (a different MDB than the one used for Generic JMS imports) is deployed. It listens for the incoming requests on the receive destination and then dispatches the requests to be processed by the SCA runtime.

Special headers

Special header properties are used in Generic JMS imports and exports to tell the target binding how to handle the message.

For example, the TargetFunctionName property is used by the default function selector to identify the name of the operation in the export interface that is being invoked.

Note: The import binding can be configured to set the TargetFunctionName header to the operation name for each operation.

J2EE resources

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

- Listener port for listening on the receive (response) destination (two-way only) for imports and on the receive (request) destination for exports
- Generic JMS connection factory for the outboundConnection (import) and inboundConnection (export)
- Generic JMS destination for the send (import) and receive (export; two-way only) destinations
- Generic JMS connection factory for the responseConnection (two-way only and optional; otherwise, outboundConnection is used for imports, and inboundConnection is used for exports)
- Generic JMS destination for the receive (import) and send (export) destination (two-way only)
- Default messaging provider callback JMS destination used to access the SIB callback queue destination (two-way only)
- Default messaging provider callback JMS connection factory used to access the callback JMS destination (two-way only)
- SIB callback queue destination used to store information about the request message for use during response processing (two-way only)

The installation task creates the ConnectionFactory, the three destinations, and the ActivationSpec from the information in the import and export files.

Administering Generic JMS bindings

Generic JMS bindings can be administered from within the server.

Use the administrative console to configure and administer Generic JMS import and export bindings.

Detailed instructions on creating Generic JMS import and export bindings can be found in the WebSphere Integration Developer information center at **WebSphere Integration Developer** > **Developing integration applications** > **Accessing external services with messaging systems** > **Generic JMS**.

Properties of Generic JMS bindings

The Generic JMS import and export bindings can be installed with all the necessary connection factories having been created during deployment, or they can be configured to point to a set of existing resources.

Typically, the Generic JMS bindings are created in WebSphere Integration Developer. You can either create the connections and destinations required for the JMS binding at the time the component is installed on your server, or you can specify the JNDI name of the resources on the server that you intend your JMS import or export to utilize.

Configuring the Generic JMS binding depends upon which option was selected.

In the case where new message provider resources are created (that is, the resources are created on the server during installation), the resources will exist and can be located and administered using the administrative console. The JNDI names of the generated artifacts are described in the following tables.

Table 28. Generic JMS imports: Names and JNDI names of resources created at installation on the server

Resource	Generated resource JNDI name
Outbound Connection	[moduleName]/[importName]_CF
Response Connection	[moduleName]/[importName]_RESP_CF
Send destination	[moduleName]/[importName]_SEND_D
Receive destination	[moduleName]/[importName]_RECEIVE_D
Callback destination	[moduleName]/[importName]_CALLBACK_D

Table 29. Generic JMS exports: Names and JNDI names of resources created at installation on the server

Resource	Generated resource JNDI name
Inbound Connection	[moduleName]/[exportName]_LIS_CF
Response Connection	[moduleName]/[exportName]_RESP_CF
Receive destination	[moduleName]/[exportName]_RECEIVE_D
Send destination	[moduleName]/[exportName]_SEND_D
Callback destination	[moduleName]/[exportName]_CALLBACK_D

Note: The resources are created at the server scope. The scope in the administrative console is initially set to **All scopes**. You must set the scope to **cell** or **node** to create a new resource. You can select an existing resource from the default list.

If you select the other option and the JMS import is expecting to find required resources on the server, you must have these resources installed and the import and export files must contain the JNDI names. The association between the JMS binding and the resources will then be made.

Setting up connectivity for the Generic JMS binding

You must set up connectivity to and from a third-party JMS provider to use the Generic JMS binding.

Before you begin

You must have permission to make and save changes to the profile on the administrative console. You must have the appropriate permissions to make and save changes in WebSphere Integration Developer and in WebSphere Application Server.

About this task

This task provides a procedural outline only; providing specific instructions for individual third-party JMS providers is beyond the scope of this topic.

The application in this scenario contains a mediation component connection to other applications at both ends by means of the Generic JMS binding; the application contains an interface with a single two-way operation.

Procedure

1. Configure your third-party JMS provider to create a queue manager, queues, and JMS connection factories and destinations using the provider-specific tooling.
2. In WebSphere Application Server, you must define a generic messaging provider.
3. In WebSphere Integration Developer, you must perform the following tasks:
   a. Add an import and export to the application and connect them to a previously-implemented mediation component.
   b. Add a Generic JMS binding to both the export and the import: **Generate binding** → **Messaging binding** → **Generate JMS binding** .
   c. Set the genericMessagingProviderName property on both the import and export to match the properties previously defined to WebSphere Application Server.
   d. Set the ExternalJNDIName for the connections and send/receive destinations to match those defined in your third-party JMS provider tooling.
4. Deploy the application to a single server.

   Make sure that the third-party JMS provider queue manager is running and available for connection and that the context to which the generic messaging provider definition points in WebSphere Application Server is available.

   You can build and deploy your application using WebSphere Integration Developer. Another way to deploy applications is to export the modules as zip files and then use the serviceDeploy command of WebSphere Process Server or WebSphere Enterprise Service Bus (mediation modules only) to build and deploy them as EAR files.
5. Start the application.
6. Run the application.

Results

The application can be run by placing messages on the third-party JMS provider queue defined in the Generic JMS export receive destination. Responses will be returned to the Generic JMS export send destination.

Similarly, the application will issue requests to the Generic JMS import send destination and expect responses on the Generic JMS import receive destination.

Configuring Generic JMS bindings

You can administer Generic JMS import and export bindings to configure special features of the resource. The administrative tasks are performed using the administrative console.

Before you begin

You must have permission to make and save changes to the profile on the administrative console, and you must have completed the connectivity setup procedure.

About this task

The Generic JMS import or export must be installed as part of a Service Component Architecture (SCA) application in your server profile.

Procedure

1. Select the SCA module. From the administrative console, select **Applications** → **SCA Modules** and then select the *modulename*. The configuration page opens.
2. Under **Module components**, expand **Imports** or **Exports**. This reveals a list of the installed imports or exports.
3. Expand the import or export and then expand **Binding**. This reveals a list of available bindings to administer.
4. Click on the binding to be administered. The general properties of the binding are displayed:
   - The **Send Resources** category contains the Connection Factory and the Send Destination.
   - The **Receive Resources** category contains the Response Connection Factory, the Listener Port, and the Activation Specification.
   - The **Advanced Resources** category contains Callback resources and other available resources.

   Note: You can also access a resource by typing the JNDI name in the text box. Doing so, however, allows you to enter the name of a resource that is not yet configured.
5. Administer the desired resource:
   a. Click **Browse** to open a window with a list of JNDI names; then, select the desired JNDI name. The selected name will populate the appropriate text field.
   b. Click **Configure** to open the corresponding panel referred to by the JNDI name. While most resources can be configured at cluster scope, selecting the **Configure** option at Listener Port displays a panel showing all listener ports with the cluster Listener Port names for all the members of the given cluster; you can then select one listener port.

      When **Configure** has been selected, the corresponding WebSphere Application Server panel will open.
6. When you are ready to administer your settings, click **Apply** or **OK**.

What to do next

The application will be automatically restarted for the JNDI bindings to take effect.

Generic JMS headers

Generic JMS headers are Service Data Objects (SDO) that contain all the properties of the Generic JMS message properties. These properties can be from the inbound message or they can be the properties that will be applied to the outbound message.

The headers are provided using the system programming interface; consequently, you set or get the headers by accessing the service component architecture message. You cannot set or get the header from a client program. You can, however, use an Enterprise Service Bus mediation to retrieve the header values or set them on an outgoing request.

The following properties are set statically on the methodBinding:

- JMSType
- JMSCorrelationID
- JMSDeliveryMode
- JMSPriority

The Generic JMS binding also supports dynamic modification of JMS headers and properties in the same manner as the JMS and MQ JMS bindings.

Some Generic JMS providers place restrictions on which properties can be set by the application and in what combinations. You must consult your third-party product documentation for more information. However, an additional property has been added to the methodBinding, ignoreInvalidOutboundJMSProperties, which allows any exceptions to be propagated.

The Generic JMS headers and message properties are used only 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`.

Troubleshooting Generic JMS bindings

You can diagnose and fix problems with Generic JMS bindings.

Implementation exceptions

In response to various error conditions, the Generic JMS import and export implementation can return one of two types of exceptions:

- Service Business Exception: this exception is returned if the fault specified on the service business interface (WSDL port type) occurred.
- Service Runtime Exception: raised in all other cases. In most cases, the cause exception will contain the original exception (JMSException).

Troubleshooting Generic JMS message expiry

A request message by the JMS provider is subject to expiration.

Request expiry refers to the expiration of a request message by the JMS provider when the JMSExpiration time on the request message is reached. As with other JMS bindings, the Generic JMS binding handles the request expiry by setting expiration on the callback message placed by the import to be the same as for the outgoing request. Notification of the expiration of the callback message will indicate that the request message has expired and the client should be notified by means of a business exception.

If the callback destination is moved to the third-party provider, however, this type of request expiry is not supported.

Response expiry refers to the expiration of a response message by the JMS provider when the JMSExpiration time on the response message is reached.

Response expiry for the generic JMS binding is not supported, because the exact expiry behavior of a third-party JMS provider is not defined. You can, however, check that the response is not expired if and when it is received.

For outbound request messages, the JMSExpiration value will be calculated from the time waited and from the requestExpiration values carried in the asyncHeader, if set.

Troubleshooting Generic JMS connection factory errors

When you define certain types of connection factories in your Generic JMS provider, you might receive an error message when you try to start an application. You can modify the external connection factory to avoid this problem.

When launching an application, you might receive the following error message:

```
MDB Listener Port JMSConnectionFactory type does not match
JMSDestination type
```

This problem can arise when you are defining external connection factories. Specifically, the exception can be thrown when you create a JMS 1.0.2 Topic Connection Factory, instead of a JMS 1.1 (unified) Connection Factory (that is, one that is able to support both point-to-point and publish/subscribe communication).

To resolve this issue, take the following steps:

1. Access the Generic JMS provider that you are using.
2. Replace the JMS 1.0.2 Topic Connection Factory that you defined with a JMS 1.1 (unified) Connection Factory.

When you launch the application with the newly defined JMS 1.1 Connection Factory, you should no longer receive an error message.

JMS-based events

If SCA messages originated through a JMS interaction fail, use the administrative facilities of the JMS provider to manage the events.

Handling exceptions

The way in which the binding is configured determines how exceptions that are raised by data handlers or data bindings are handled. Additionally, the nature of the mediation flow dictates the behavior of the system when such an exception is thrown.

A variety of problems can occur when a data handler or data binding is called by your binding. For example, a data handler might receive a message that has a corrupt payload, or it might try to read a message that has an incorrect format.

The way your binding handles such an exception is determined by how you implement the data handler or data binding. The recommended behavior is that you design your data binding to throw a `DataBindingException`.

The situation is similar for a data handler. Since the data handler is invoked by the data binding, any data handler exception is wrapped into a data binding exception. Therefore a `DataHandlerException` is reported to you as a `DataBindingException`.

WebSphere MQ JMS bindings

The WebSphere MQ JMS binding provides integration with external applications that use a WebSphere MQ JMS-based provider.

Use the WebSphere MQ JMS export and import bindings to integrate directly with external JMS or MQ JMS systems from your server environment. This eliminates the need to use MQ Link or Client Link features of the Service Integration Bus.

When a component interacts with a WebSphere MQ JMS-based service by way of an import, the WebSphere MQ JMS import binding utilizes a destination to which data will be sent and a destination where the reply can be received. Conversion of the data to and from a JMS message is accomplished through the JMS data handler or data binding edge component.

When an SCA module provides a service to WebSphere MQ JMS clients, the WebSphere MQ JMS export binding utilizes a destination where the request can be received and the response can be sent. The conversion of the data to and from a JMS message is done through the JMS data handler or data binding.

The function selector provides a mapping to the operation on the target component to be invoked.

WebSphere MQ JMS bindings: a general perspective

The WebSphere MQ JMS binding provides integration with external applications that use the WebSphere MQ JMS provider.

WebSphere MQ administrative tasks

The WebSphere MQ system administrator is expected to create the underlying WebSphere MQ Queue Manager, which the WebSphere MQ JMS bindings will utilize, before running an application containing these bindings.

WebSphere MQ JMS import bindings

The WebSphere MQ JMS import allows components within your SCA module to communicate with services provided by WebSphere MQ JMS-based providers. You must be using a supported version of WebSphere MQ. Detailed hardware and software requirements can be found on the IBM support pages.

Two types of usage scenarios for WebSphere MQ JMS import bindings are supported, depending on the type of operation being invoked:

- One-way: The WebSphere MQ JMS import puts a message on the send destination configured in the import binding

  Nothing is sent to the replyTo field of the JMS header.
- Two-way (request-response): The WebSphere MQ JMS import puts a message on the send destination.

  The receive destination is set in the replyTo header field. A message-driven bean (MDB) is deployed to listen on the receive destination, and when a reply is received, the MDB passes the reply back to the component.

  The import binding can be configured (using the **Response correlation scheme** field in WebSphere Integration Developer) to expect the response message correlation ID to have been copied from the request message ID (the default) or from the request message correlation ID.

For both one-way and two-way usage scenarios, dynamic and static header properties can be specified.Static properties can be set from the JMS import method binding. Some of these properties have special meanings to the SCA JMS runtime.

It is important to note that WebSphere MQ JMS is an asynchronous binding. If a calling component invokes a WebSphere MQ JMS import synchronously (for a two-way operation), the calling component is blocked until the response is returned by the JMS service.

Figure 25 on page 235 illustrates how the import is linked to the external service.

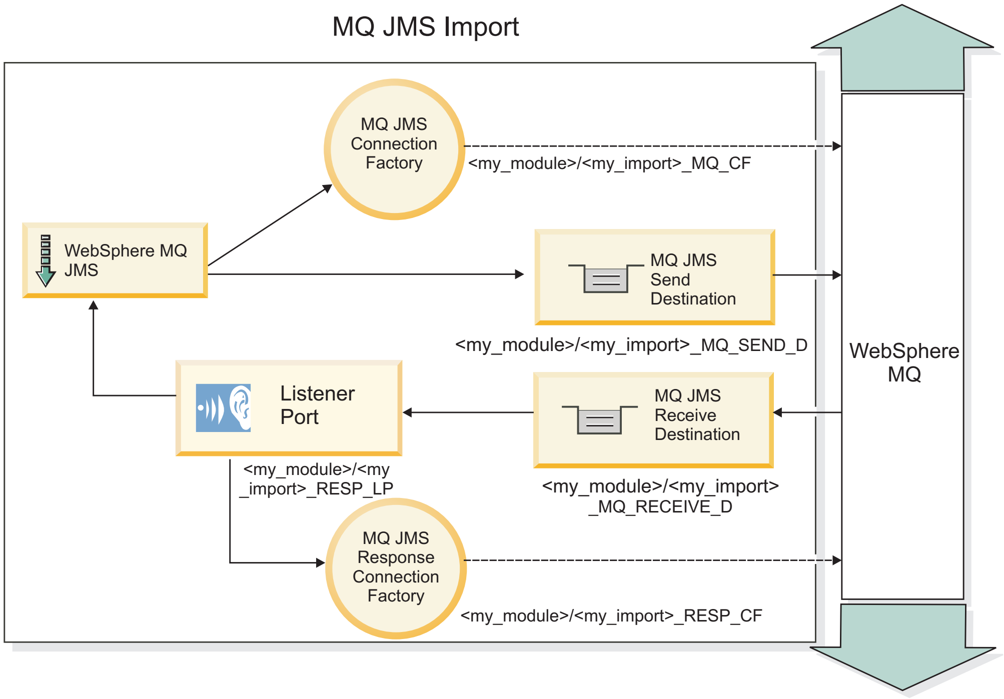

MQ JMS Import
MQ JMS Connection Factory
<my_module>/<my_import>_MQ_CF
WebSphere MQ JMS
MQ JMS Send Destination
<my_module>/<my_import>_MQ_SEND_D
WebSphere MQ
Listener Port
MQ JMS Receive Destination
<my_module>/<my_import>_RESP_LP
<my_module>/<my_import> _MQ_RECEIVE_D
MQ JMS Response Connection Factory
<my_module>/<my_import>_RESP_CF

Figure 25. WebSphere MQ JMS import binding resources

WebSphere MQ JMS export bindings

The WebSphere MQ JMS export binding provides the means for SCA modules to provide services to external JMS applications on the WebSphere MQ-based JMS provider.

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 component provides a reply. The destination specified in the replyTo field of the response message overrides the destination specified in the send field.

Figure 26 on page 236 illustrates how the external requestor is linked to the export.

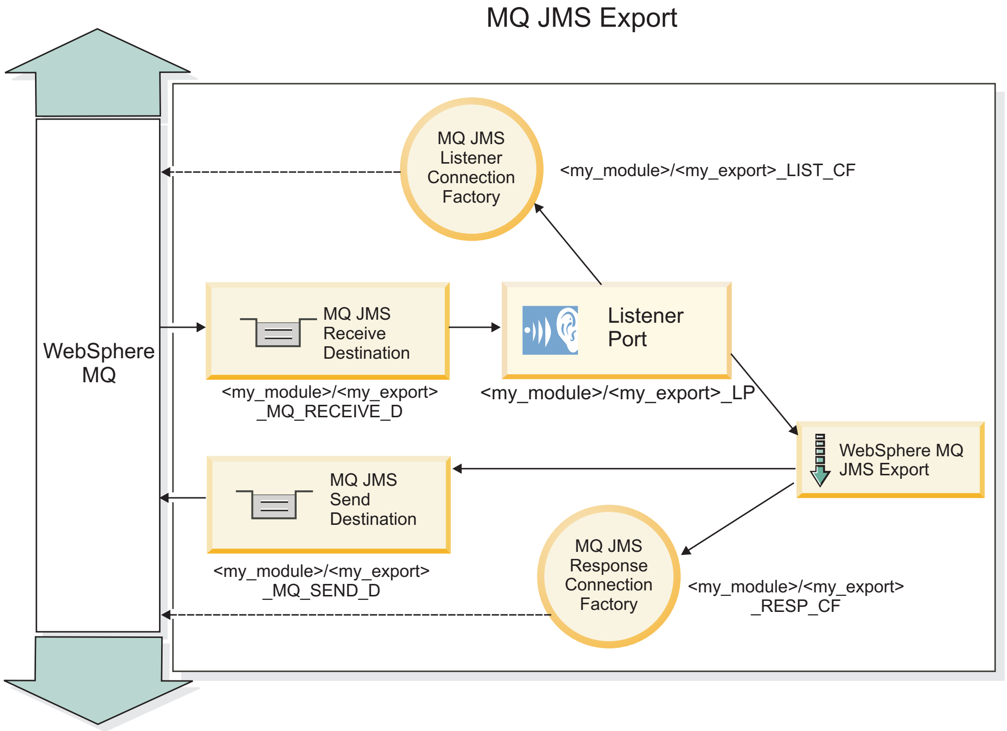

MQ JMS Export
WebSphere MQ
MQ JMS Listener Connection Factory
<my_module>/<my_export>_LIST_CF
MQ JMS Receive Destination
<my_module>/<my_export> _MQ_RECEIVE_D
Listener Port
<my_module>/<my_export>_LP
WebSphere MQ JMS Export
MQ JMS Send Destination
<my_module>/<my_export> _MQ_SEND_D
MQ JMS Response Connection Factory
<my_module>/<my_export> _RESP_CF

Figure 26. WebSphere MQ JMS export binding resources

Key features of WebSphere MQ JMS bindings

Key features of WebSphere MQ JMS bindings include headers, J2EE artifacts, and created J2EE resources.

Headers

A JMS message header contains a number of predefined fields containing values used by both clients and providers to identify and to route messages. You can use binding properties to configure these headers with fixed values, or the headers can be specified dynamically at runtime.

JMSCorrelationID
Links to a related message. Typically, this field is set to the message identifier string of the message that is being replied to.

TargetFunctionName
This header is used by one of the supplied function selectors to identify the operation being invoked. Setting the TargetFunctionName JMS header property in messages sent to a JMS export allows this function selector to be used. The property can be set directly in JMS client applications or when connecting an import with a JMS binding to such an export. In this case, the JMS import binding should be configured to set the TargetFunctionName header for each operation in the interface to the name of the operation.

Correlation schemes

The WebSphere MQ JMS bindings provide various correlation schemes that are used to determine how to correlate request messages with response messages.

RequestMsgIDToCorrelID
The JMSMessageID is copied to the JMSCorrelationID field. This is the default setting.

RequestCorrelIDToCorrelID
The JMSCorrelationID is copied to the JMSCorrelationID field.

J2EE resources

A number of J2EE resources are created when an MQ JMS import is deployed to a J2EE environment.

Parameters

MQ Connection Factory
Used by clients to create a connection to the MQ JMS provider.

Response Connection Factory
Used by the SCA MQ JMS runtime when the send destination is on a different Queue Manager than the receive destination.

Listener Port
Specifies an association between a connection factory, a destination, and a message-driven bean. This enables deployed message-driven beans associated with the port to retrieve messages from the destination.

Destinations

- Send destination:
  - Imports: Where the request or outgoing message is sent.
  - Exports: Where the response message will be sent if it is not superseded by the JMSReplyTo header field of the incoming message.
- Receive destination:
  - Imports: Where the response or incoming message should be placed.
  - Exports: Where the incoming or request message should be placed.

Administering WebSphere MQ JMS bindings

WebSphere MQ JMS bindings can be administered from within the server.

Use the administrative console to access WebSphere MQ JMS bindings.

For detailed instructions on generating WebSphere MQ JMS imports and exports, see ″Working with MQ JMS bindings″ in the WebSphere Integration Developer Information Center .

Properties of MQ JMS bindings

MQ JMS bindings can be installed with all the necessary connection factories having been created during deployment, or they can be configured to point to a set of existing resources.

Typically, MQ JMS bindings are created in WebSphere Integration Developer. You can either create the connections and destinations required for the JMS binding at the time the component is installed on your server, or you can specify the JNDI name of the resources on the server that you intend your MQ JMS import or export to utilize.

Configuring the MQ JMS binding depends upon which option was selected.

In the case where new message provider resources are created (that is, the resources are created on the server during installation), the resources will exist and can be located and administered using the administrative console.

Examples of the JNDI names of the generated artifacts are described in the following tables.

Table 30. MQ JMS imports: Names and JNDI names of resources created at installation on the server

Resource	Module name	Import name	Resource global JNDI name
Outbound Connection Factory	mqjms.module	my/import	mqjms.module/my/import_MQ_CF
Response Listener Port	mqjms.module	my/import	mqjms.module.my.import_RESP_LP (Note: This is only a name, not JNDI)
Response Connection Factory	mqjms.module	my/import	mqjms.module/my/import_RESP_CF
Send	mqjms.module	my/import	mqjms.module/my/import_MQ_SEND_D
Receive	mqjms.module	my/import	mqjms.module/my/export_MQ_RECEIVE_D
SIB Callback Destination	mqjms.module	my/import	mqjms.module/my/import_MQ_CALLBACK_D
SIB Callback Connection Factory	All modules	my/import	SCA.MQJMS/Callback_CF

Table 31. MQ JMS exports: Names and JNDI names of resources created at installation on the server

Resource	Module name	Export name	Resource global JNDI name
Listener Port	mqjms.module	my/export	mqjms.module.my.export_LP (Note: This is just a name, not JNDI)
Inbound Connection Factory	mqjms.module	my/export	mqjms.module/my/export_LIS_CF
Response Connection Factory	mqjms.module	my/export	mqjms.module/my/export_RESP_CF
Receive	mqjms.module	my/export	mqjms.module/my/export_MQ_RECEIVE_D
Send	mqjms.module	my/export	mqjms.module/my/export_MQ_SEND_D
SIB Callback Destination	mqjms.module	my/export	mqjms.module/my/export_MQ_CALLBACK_D
SIB Callback Connection Factory	All modules	my/export	SCA.MQJMS/Callback_CF

Note:

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

- The SIB callback destination and SIB callback connection factory are SIB JMS resources. The other entries in the table are MQ JMS resources. The two types of resources are administered separately in the administrative console.

If you select the other option and the MQ JMS import or export binding is expecting to find on the server resources that it will use, you must have these resources installed and the import file must contain their JNDI names. The association between the MQ JMS import and the resources will then be made.

Configuring MQ JMS bindings

You can administer MQ JMS bindings to configure special features of the resource. The administrative tasks are performed using the administrative console.

Before you begin

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

The queue and queue manager are not automatically generated; they must be created in WebSphere MQ by your WebSphere MQ administrator.

About this task

The MQ JMS import or export must be installed as part of a Service Component Architecture (SCA) application in your server profile.

Procedure

1. Select the SCA module. From the administrative console, click **Applications** → **SCA Modules** and then click the *modulename*. The configuration page opens.
2. Under **Module components**, expand **Imports** or **Exports**. This reveals a list of the installed imports or exports.
3. Expand the import or export and then expand **Binding**. This reveals a list of available bindings to administer.
4. Click on the binding to be administered. The general properties of the binding are displayed:
   - The **Send Resources** category contains the Connection Factory and the Send Destination.
   - The **Receive Resources** category contains the Response Connection Factory, the Listener Port, and the Activation Specification.
   - The **Advanced Resources** category contains Callback resources and other available resources.

   Note: You can also access a resource by typing the JNDI name in the text box. Doing so, however, allows you to enter the name of a resource that is not yet configured.
5. Administer the desired resource:
   a. Click **Browse** to open a window with a list of JNDI names; then, select the desired JNDI name. The selected name will populate the appropriate text field.
   b. Click **Configure** to open the corresponding panel referred to by the JNDI name. While most resources can be configured at cluster scope, selecting the **Configure** option at Listener Port displays a panel showing all listener ports with the cluster Listener Port names for all the members of the given cluster; you can then select one listener port.

When **Configure** has been selected, the corresponding WebSphere Application Server panel will open.

6. When you are ready to administer your settings, click **Apply** or **OK**.

What to do next

The application will be automatically restarted for the JNDI bindings to take effect.

JMS headers

A JMS message contains two types of headers–the JMS system header and multiple JMS properties. Both types of headers can be accessed either in a mediation module in the Service Message Object (SMO) or by using the ContextService API.

JMS system header

The JMS system header is represented in the SMO by the JMSHeader element, which contains all the fields typically found in a JMS header. Although these can be modified in the mediation (or ContextService), some JMS system header fields set in the SMO will not be propagated in the outbound JMS message as they are overridden by system or static values.

The key fields in the JMS system header that can be updated in a mediation (or ContextService) are:

- **JMSType** and **JMSCorrelationID** – values of the specific predefined message header properties
- **JMSDeliveryMode** – values for delivery mode (persistent or nonpersistent; default is persistent)
- **JMSPriority** – priority value (0 to 9; default is JMS_Default_Priority)

JMS properties

JMS properties are represented in the SMO as entries in the Properties list. The properties can be added, updated, or deleted in a mediation or by using the ContextService API.

Properties can also be set statically in the JMS binding. Properties that are set statically override settings (with the same name) that are set dynamically.

User properties propagated from other bindings (for example, an HTTP binding) will be output in the JMS binding as JMS properties.

Header propagation settings

The propagation of the JMS system header and properties either from the inbound JMS message to downstream components or from upstream components to the outbound JMS message can be controlled by the Propagate Protocol Header flag on the binding.

When Propagate Protocol Header is set, header information is allowed to flow to the message or to the target component, as described in the following list:

- JMS export request

  The JMS header received in the message will be propagated to target components by way of the context service. JMS properties received in the message will be propagated to target components by way of the context service.

- JMS export response

  Any JMS header fields set in the context service will be used in the outbound message, if not overridden by static properties set on the JMS export binding. Any properties set in the context service will be used in the outbound message if not overridden by static properties set on the JMS export binding.
- JMS import request

  Any JMS header fields set in the context service will be used in the outbound message, if not overridden by static properties set on the JMS import binding. Any properties set in the context service will be used in the outbound message if not overridden by static properties set on the JMS import binding.
- JMS import response

  The JMS header received in the message will be propagated to target components by way of the context service. JMS properties received in the message will be propagated to target components by way of the context service.

External clients

The server can send messages to, or receive messages from, external clients using WebSphere MQ JMS bindings.

An external client (such as a Web portal or an enterprise information system) can send a message to an SCA component in the application by way of an export or it can be invoked by an SCA component in the application by way of an import.

The WebSphere MQ JMS export binding deploys message driven beans (MDBs) to listen to requests incoming to the receive destination specified in the export binding. The destination specified in the send field is used to send the reply to the inbound request if the invoked application provides a reply. Thus, an external client is able to invoke applications via the export binding.

WebSphere MQ JMS imports bind to, and can deliver message to, external clients. This message might or might not demand a response from the external client.

More information on how to interact with external clients using WebSphere MQ can be found at the WebSphere MQ information center.

Troubleshooting WebSphere MQ JMS bindings

You can diagnose and fix problems with WebSphere MQ JMS bindings.

Implementation exceptions

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

- Service Business Exception: this exception is returned if the fault specified on the service business interface (WSDL port type) occurred.
- Service Runtime Exception: raised in all other cases. In most cases, the cause exception will contain the original exception (JMSException).

  For example, an import expects only one response message for each request message. If more than one response arrives, or if a late response (one for which the SCA response expiration has expired) arrives, a Service Runtime Exception is thrown. The transaction is rolled back, and the response message is backed out of the queue or handled by the failed event manager.

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.

Misusage scenarios: comparison with WebSphere MQ bindings

The WebSphere MQ JMS binding is designed to interoperate with JMS applications deployed against WebSphere MQ, which exposes messages according to the JMS message model. The WebSphere MQ import and export, however, are principally designed to interoperate with native WebSphere MQ applications and expose the full content of the WebSphere MQ message body to mediations.

The following scenarios should be built using the WebSphere MQ JMS binding, not the WebSphere MQ binding:

- Invoking a JMS message-driven bean (MDB) from an SCA module, where the MDB is deployed against the WebSphere MQ JMS provider. Use a WebSphere MQ JMS import.
- Allowing the SCA module to be called from a J2EE component servlet or EJB by way of JMS. Use a WebSphere MQ JMS export.
- Mediating the contents of a JMS MapMessage, in transit across WebSphere MQ. Use a WebSphere MQ JMS export and import in conjunction with the appropriate data handler or data binding.

There are situations in which the WebSphere MQ binding and WebSphere MQ JMS binding might be expected to interoperate. In particular, when you are bridging between J2EE and non-J2EE WebSphere MQ applications, use a WebSphere MQ export and WebSphere MQ JMS import (or vice versa) in conjunction with appropriate data bindings or mediation modules (or both).

Handling exceptions

The way in which the binding is configured determines how exceptions that are raised by data handlers or data bindings are handled. Additionally, the nature of the mediation flow dictates the behavior of the system when such an exception is thrown.

A variety of problems can occur when a data handler or data binding is called by your binding. For example, a data handler might receive a message that has a corrupt payload, or it might try to read a message that has an incorrect format.

The way your binding handles such an exception is determined by how you implement the data handler or data binding. The recommended behavior is that you design your data binding to throw a `DataBindingException`.

The situation is similar for a data handler. Since the data handler is invoked by the data binding, any data handler exception is wrapped into a data binding exception. Therefore a `DataHandlerException` is reported to you as a `DataBindingException`.

WebSphere MQ bindings

The WebSphere MQ binding provides Service Component Architecture (SCA) connectivity with WebSphere MQ applications.

Use the WebSphere MQ export and import bindings to integrate directly with a WebSphere MQ-based system from your server environment. This eliminates the need to use MQ Link or Client Link features of the Service Integration Bus.

Note: MQ bindings only support CLIENT transport mode.

When a component interacts with a WebSphere MQ service by way of an import, the WebSphere MQ import binding utilizes a queue to which data will be sent and a queue where the reply can be received.

When an SCA module provides a service to WebSphere MQ clients, the WebSphere MQ export binding utilizes a queue where the request can be received and the response can be sent. The function selector provides a mapping to the operation on the target component to be invoked.

Conversion of the payload data to and from an MQ message is done through the MQ body data handler or data binding. Conversion of the header data to and from an MQ message is done through the MQ header data binding.

For information about the WebSphere MQ versions supported, see the WebSphere Process Server system requirements Web site.

WebSphere MQ bindings: a general perspective

The WebSphere MQ binding provides integration with native MQ-based applications.

WebSphere MQ administrative tasks

The WebSphere MQ system administrator is expected to create the underlying WebSphere MQ Queue Manager, which the WebSphere MQ bindings will use, before running an application containing these bindings.

WebSphere administrative tasks

You must change the MQ_INSTALL_ROOT environment variable in WebSphere to the WebSphere MQ version supported by the server, and restart the server. This ensures that the libraries of a supported version of WebSphere MQ are being used. Detailed hardware and software requirements can be found on the IBM support pages

WebSphere MQ import bindings

The WebSphere MQ import binding allows components within your SCA module to communicate with services provided by external WebSphere MQ-based applications. You must be using a supported version of WebSphere MQ. Detailed hardware and software requirements can be found on the IBM support pages.

Interaction with external WebSphere MQ systems includes the use of queues for sending requests and receiving replies.

Two types of usage scenarios for the WebSphere MQ import binding are supported, depending on the type of operation being invoked:

- One-way: The WebSphere MQ import puts a message on the queue configured in the **Send destination queue** field of the import binding. Nothing is sent to the replyTo field of the MQMD header.
- Two-way (request-response): The WebSphere MQ import puts a message on the queue configured in the **Send destination queue** field

  The receive queue is set in the replyTo MQMD header field. A message driven bean (MDB) is deployed to listen on the receive queue, and when a reply is received, the MDB passes the reply back to the component.

  The import binding can be configured (using the **Response correlation scheme** field) to expect the response message correlation ID to have been copied from the request message ID (the default) or from the request message correlation ID.

It is important to note that JMS is an asynchronous binding. If a calling component invokes a JMS import synchronously (for a two-way operation), the calling component is blocked until the response is returned by the JMS service.

Figure 27 illustrates how the import is linked to the external service.

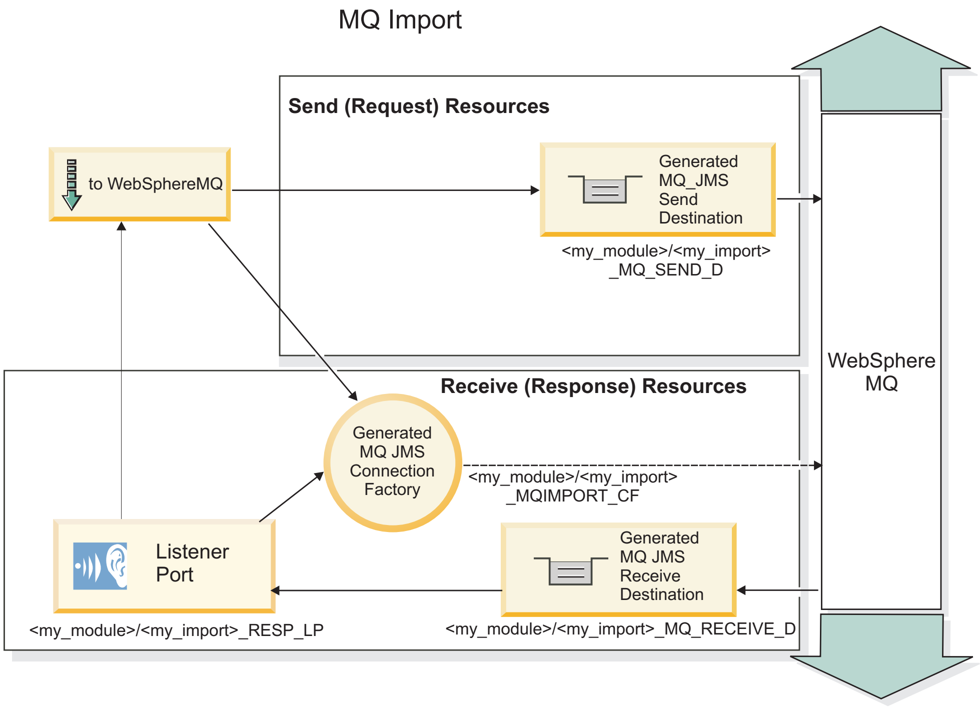

MQ Import
Send (Request) Resources
to WebSphereMQ
Generated MQ_JMS Send Destination
<my_module>/<my_import> _MQ_SEND_D
WebSphere MQ
Receive (Response) Resources
Generated MQ JMS Connection Factory
<my_module>/<my_import> _MQIMPORT_CF
Listener Port
Generated MQ JMS Receive Destination
<my_module>/<my_import>_RESP_LP
<my_module>/<my_import>_MQ_RECEIVE_D

Figure 27. WebSphere MQ import binding resources

WebSphere MQ export bindings

The WebSphere MQ export binding provides the means for SCA modules to provide services to external WebSphere MQ-based applications.

An MDB is deployed to listen to requests incoming to the **Receive destination queue** specified in the export binding. The queue specified in the **Send destination queue** field is used to send the reply to the inbound request if the invoked component provides a reply. The queue specified in the replyTo field of the response message overrides the queue specified in the **Send destination queue** field.

Figure 28 illustrates how the external requestor is linked to the export.

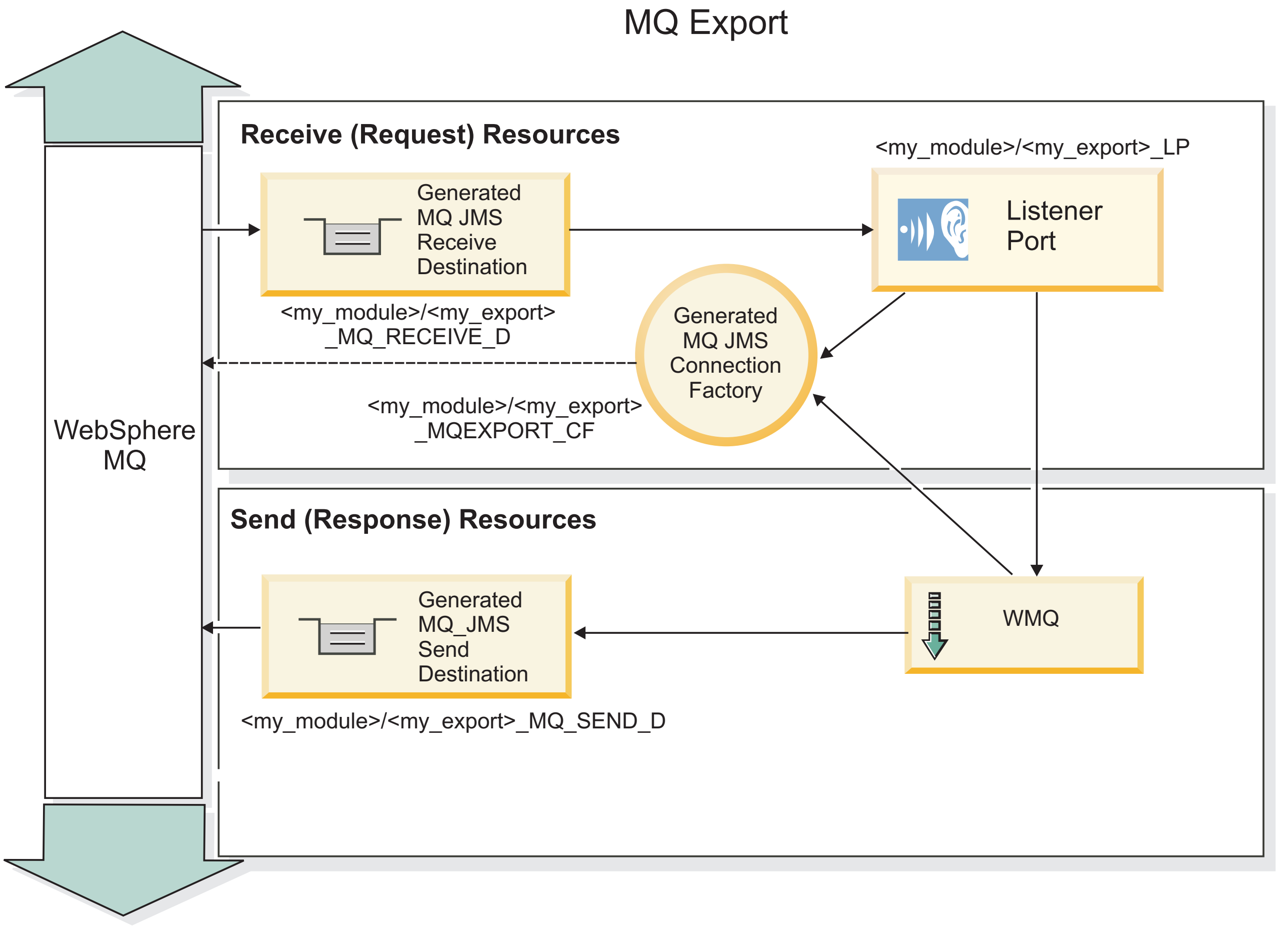

MQ Export
Receive (Request) Resources
<my_module>/<my_export>_LP
Generated MQ JMS Receive Destination
Listener Port
<my_module>/<my_export> _MQ_RECEIVE_D
Generated MQ JMS Connection Factory
<my_module>/<my_export> _MQEXPORT_CF
WebSphere MQ
Send (Response) Resources
Generated MQ_JMS Send Destination
WMQ
<my_module>/<my_export>_MQ_SEND_D

Figure 28. WebSphere MQ export binding resources

Key features of a WebSphere MQ binding

Key features of a WebSphere MQ binding include headers, J2EE artifacts, and created J2EE resources.

Correlation schemes

A WebSphere MQ request/reply application can use one of a number of techniques to correlate response messages with requests, built around the MQMD MessageID and CorrelID fields. In the vast majority of cases, the requestor lets the queue manager select a MessageID and expects the responding application to copy this into the CorrelID of the response. In most cases, the requestor and responding application implicitly know which correlation technique is in use. Occasionally the responding application will honor various flags in the Report field of the request that describe how to handle these fields.

Export bindings for WebSphere MQ messages can be configured with the following options:

Response MsgId options:

New MsgID
Allows the queue manager to select a unique MsgId for the response (default).

Copy from Request MsgID
Copies the MsgId field from the MsgId field in the request.

Copy from SCA message
Sets the MsgId to be that carried in WebSphere MQ headers in the SCA response message, or lets the queue manager define a new Id if the value does not exist.

As Report Options
Inspects the Report field of the MQMD in the request for a hint as to how to handle the MsgId. The MQRO_NEW_MSG_ID and MQRO_PASS_MSG_ID options are supported and behave like New MsgId and Copy from Request MsgID, respectively

Response CorrelId options:

Copy from Request MsgID
Copies the CorrelId field from the MsgId field in the request (default).

Copy from Request CorrelID
Copies the CorrelId field from the CorrelId field in the request.

Copy from SCA message
Sets the CorrelId to be carried in WebSphere MQ headers in the SCA response message or leaves it blank if the value does not exist.

As Report Options
Inspects the Report field of the MQMD in the request for a hint as to how to handle the CorrelId. The MQRO_COPY_MSG_ID_TO_CORREL_ID and MQRO_PASS_CORREL_ID options are supported and behave like Copy from Request MsgID and Copy from Request CorrelID, respectively

Import bindings for WebSphere MQ messages can be configured with the following options:

Request MsgId options:

New MsgID
Allows the queue manager to select a unique MsgId for the request (default)

Copy from SCA message
Sets the MsgId to be carried in WebSphere MQ headers in the SCA request message or lets the queue manager define a new Id if the value does not exist.

Response correlation options:

Response has CorrelID copied from MsgId
Expects the response message to have a CorrelId field set, per the MsgId of the request (default).

Response has MsgID copied from MsgId
Expects the response message to have a MsgId field set, per the MsgId of the request.

Response has CorrelID copied from CorrelId
Expects the response message to have a CorrelId field set, per the CorrelId of the request.

J2EE resources

A number of J2EE resources are created when a WebSphere MQ binding is deployed to a J2EE environment.

Parameters

MQ Connection Factory
Used by clients to create a connection to the WebSphere MQ provider.

Response Connection Factory
Used by the SCA MQ runtime when the send destination is on a different Queue Manager than the receive destination.

Listener Port
Specifies an association between a connection factory, a destination, and a message-driven bean. This enables deployed message-driven beans associated with the port to retrieve messages from the destination.

Destinations

- Send destination: where the request or outgoing message is sent (import); where the response message will be sent (export), if not superseded by the MQMD ReplyTo header field in the incoming message.
- Receive destination: where the response/request or incoming message should be placed.

Administering WebSphere MQ bindings

WebSphere MQ bindings can be administered from within the server.

Use the administrative console to access WebSphere MQ bindings.

Detailed instructions on creating WebSphere MQ imports and exports can be found in the WebSphere Integration Developer information center at **WebSphere Integration Developer** > **Developing integration applications** > **Accessing external services with messaging systems** > **WebSphere MQ > Working with MQ bindings**.

Properties of WebSphere MQ bindings

The WebSphere MQ binding can be installed with all the necessary connection factories having been created during deployment, or they can be configured to point to a set of existing resources.

Typically, WebSphere MQ import and export bindings are created in WebSphere Integration Developer. You can either create the connections and destinations required for the WebSphere MQ binding at the time the component is installed on your server, or you can specify the JNDI name of the resources on the server that you intend your WebSphere MQ binding to utilize.

Configuring the WebSphere MQ binding depends upon which option was selected.

In the case where new message provider resources are created (that is, the resources are created on the server during installation), the resources will exist and can be located and administered using the administrative console.

Examples of the JNDI names of the generated artifacts are described in the following tables.

Table 32. WebSphere MQ import: Names and JNDI names of resources created at installation on the server

Resource	Module name	Import name	Resource global JNDI name
Outbound Connection Factory	mq.module	my/import	mq.module/my/import_MQ_CF
Response Listener Port	mq.module	my/import	mq.module.my.import_RESP_LP (Note: This is only a name, not JNDI)
Response Connection Factory	mq.module	my/import	mq.module/my/import_RESP_CF
Send	mq.module	my/import	mq.module/my/import_MQ_SEND_D
Receive	mq.module	my/import	mq.module/my/export_MQ_RECEIVE_D
SIB Callback Destination	mq.module	my/import	mq.module/my/ import_MQ_CALLBACK_D
SIB Callback Connection Factory	All modules	my/import	SCA.MQ/Callback_CF

Table 33. WebSphere MQ export: Names and JNDI names of resources created at installation on the server

Resource	Module name	Export name	Resource global JNDI name
Listener Port	mq.module	my/export	mq.module.my.export_LP (Note: This is just a name, not JNDI)
Inbound Connection Factory	mq.module	my/export	mq.module/my/export_LIS_CF
Response Connection Factory	mq.module	my/export	mq.module/my/export_RESP_CF
Receive	mq.module	my/export	mq.module/my/export_MQ_RECEIVE_D
Send	mq.module	my/export	mq.module/my/export_MQ_SEND_D
SIB Callback Destination	mq.module	my/export	mq.module/my/ export_MQ_CALLBACK_D
SIB Callback Connection Factory	All modules	my/export	SCA.MQ/Callback_CF

Note:

- The resources are created at the server scope. The default scope in the administrative console is cell. You must change the scope in order to locate and administer the resources.
- The SIB Callback Destination and SIB Callback Connection Factory are SIB JMS resources. The other entries in the table are WebSphere MQ resources. The two types of resources are administered separately from the administrative console.

If you select the other option and the WebSphere MQ binding is expecting to find resources on the server that it will use, you must have these resources installed and the import or export file must contain their JNDI names. The association between the WebSphere MQ binding and the resources will then be made.

Configuring WebSphere MQ bindings

You can administer WebSphere MQ import and export bindings to tune, or set, special features of the resource. The administrative tasks are performed using the administrative console.

Before you begin

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

The queue and queue manager are not automatically generated and must be created in WebSphere MQ by your WebSphere MQ administrator.

About this task

The WebSphere MQ import must be installed as part of a Service Component Architecture (SCA) application in your server profile.

Procedure

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

   Expand **JMS Providers** and click **WebSphere MQ**.
2. Optional: Administer WebSphere MQ connection factories.

   Click **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 the MQ connection factory that you want to administer, or click **New** to create a new connection factory.

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

   You set these properties in the bindings for the resource reference of the application. If you do not want to modify the bindings for an existing application, locate this connection factory in the J2C panels where you can find these properties.
3. Optional: Administer WebSphere MQ queue connection factories.

   Click **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 the WebSphere MQ queue connection factory that you want to administer, or click **New** to create a new queue connection factory.

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

   A WebSphere MQ queue connection factory is used to create JMS connections to queues provided by WebSphere MQ for point-to-point messaging. Use WebSphere MQ queue connection factory administrative objects to manage queue connection factories for the WebSphere MQ JMS provider.
4. Optional: Administer WebSphere MQ queue destinations.

   Click **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 the queue destination that you want to administer, or click **New** to create a new WebSphere MQ queue destination.

Use the subsequent panel to browse or change the configuration properties of the selected queue destination for point-to-point messaging with WebSphere MQ as a messaging provider.

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

5. Save your changes to the master profile and, if necessary, restart the server.

WebSphere MQ headers

WebSphere MQ headers incorporate certain conventions for conversion to the service component architecture (SCA) messages.

WebSphere MQ messages consist of a system header (the MQMD), zero or more other MQ headers (system or custom), and a message body. If multiple message headers exist in the message, the order of the headers is significant.

Each header contains information describing the structure of the following header. The MQMD describes the first header.

How MQ headers are parsed

An MQ Header data binding is used to parse MQ headers. The following headers are supported automatically:

- MQRFH
- MQRFH2
- MQCIH
- MQIIH

Headers that start with `MQH` are handled differently. Specific fields of the header are not parsed; they remain as unparsed bytes.

For other MQ headers, you can write custom MQ header data bindings to parse those headers.

How MQ headers are accessed

MQ headers can be accessed in the product in one of two ways:

- Through the service message object (SMO) in a mediation
- Through the ContextService API

MQ headers are represented internally with the SMO MQHeader element. MQHeader is a container of header data that extends MQControl but contains a value element of anyType. It contains the MQMD, MQControl (MQ message body control information), and a list of other MQ headers.

- MQMD represents the contents of the WebSphere MQ message description, except for information determining the structure and encoding of the body.
- MQControl contains information determining the structure and encoding of a message body.
- MQHeaders contain a list of MQHeader objects.

The MQ header chain is unwound so that, inside the SMO, each MQ header carries its own control information (CCSID, Encoding, and Format). Headers can be added or deleted easily, without altering other header data.

Service Component Architecture modules and WebSphere MQ

SCA modules and WebSphere MQ queues can be connected to provide services to one another.

Service Component Architecture (SCA) modules can communicate with WebSphere MQ applications much in the same way as they do other SCA modules. A module that wants to send requests to a WebSphere MQ application uses an import configured with the correct response and request queues associated with that application. Similarly, an SCA module can provide services to a WebSphere MQ application using an export configured with the appropriate application request and response queues. You define the connections between the SCA modules and the WebSphere MQ queues when you build your module.

From the WebSphere MQ queue manager perspective, the SCA module looks as if it were a normal MQ client. From the SCA module end, the WebSphere MQ queue looks like any other service. You can even further shield the SCA module from the WebSphere MQ queues by using a mediation module between the SCA module and the WebSphere MQ queue and let the mediation transform the original SCA request to the correct format for the target queue and handle the response when it becomes available.

Restrictions: When configuring WebSphere MQ for imports and exports, keep in mind the following:

- You must configure queue destinations with the target client set to **MQ**.
- You must use a TCP/IP client connection to connect to WebSphere MQ. See "WebSphere MQ Intercommunication" for details.
- You cannot use channel compression.
- You must set any necessary channel exits for correct data handling: click **WebSphere MQ messaging provider** → **WebSphere MQ connection factory objects** → **Custom Properties** and set the following custom properties:
  - **SENDEXIT** to the value com.ibm.ws.sca.internal.mq.exit.MQInternalSendExitImpl.
  - **RECEXIT** to the value com.ibm.ws.sca.internal.mq.exit.MQInternalReceiveExitImpl.
  - Optional: **SENDEXITINIT** to a text string of your choice. Setting this property helps you identify the exit in a trace.
  - Optional: **RECEXITINIT** to a text string of your choice. Setting this property helps you identify the exit in a trace.
- You cannot use the same connection factories for WebSphere MQ and Java Message Service (JMS) MQ bindings as the exits will inhibit the JMS MQ bindings.

External clients

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

An external client (for example, a Web portal or an enterprise information system) can send a message to an SCA component in the application by way of an export or it can be invoked by an SCA component in the application by way of an import.

The WebSphere MQ export binding deploys message driven beans (MDBs) to listen to requests incoming to the receive destination specified in the export binding. The destination specified in the send field is used to send the reply to the inbound request if the invoked application provides a reply. Thus, an external client is able to invoke applications by way of the export binding.

WebSphere MQ imports bind to, and can deliver message to, external clients. This message might or might not demand a response from the external client.

More information on how to interact with external clients using WebSphere MQ can be found at the WebSphere MQ information center.

Troubleshooting WebSphere MQ bindings

You can diagnose and fix faults and failure conditions that occur with WebSphere MQ bindings.

Primary failure conditions

The primary failure conditions of WebSphere MQ bindings are determined by transactional semantics, by WebSphere MQ configuration, or by reference to existing behavior in other components.

Note: MQ bindings only support CLIENT transport mode.
The primary failure conditions include:

- Failure to connect to the WebSphere MQ queue manager or queue.

  A failure to connect to WebSphere MQ to receive messages will result in the MDB Listener Port failing to start. This condition will be logged in the WebSphere Application Server log. Persistent messages will remain on the WebSphere MQ queue until they are successfully retrieved (or expired by WebSphere MQ).

  A failure to connect to WebSphere MQ to send outbound messages will cause rollback of the transaction controlling the send.
- Failure to parse an inbound message or to construct an outbound message.

  A failure in the data binding causes rollback of the transaction controlling the work.
- Failure to send the outbound message.

  A failure to send a message causes rollback of the relevant transaction.
- Multiple or unexpected response messages.

  The import expects only one response message for each request message. If more than one response arrives, or if a late response (one for which the SCA response expiration has expired) arrives, a Service Runtime Exception is thrown. The transaction is rolled back, and the response message is backed out of the queue or handled by the failed event manager.

Misusage scenarios: comparison with WebSphere MQ JMS bindings

The WebSphere MQ import and export are principally designed to interoperate with native WebSphere MQ applications and expose the full content of the WebSphere MQ message body to mediations. The WebSphere MQ JMS binding, however, is designed to interoperate with JMS applications deployed against WebSphere MQ, which exposes messages according to the JMS message model.

The following scenarios should be built using the WebSphere MQ JMS binding, not the WebSphere MQ binding:

- Invoking a JMS message-driven bean (MDB) from an SCA module, where the MDB is deployed against the WebSphere MQ JMS provider. Use a WebSphere MQ JMS import.
- Allowing the SCA module to be called from a J2EE component servlet or EJB by way of JMS. Use a WebSphere MQ JMS export.
- Mediating the contents of a JMS MapMessage, in transit across WebSphere MQ. Use a WebSphere MQ JMS export and import in conjunction with the appropriate data binding.

There are situations in which the WebSphere MQ binding and WebSphere MQ JMS binding might be expected to interoperate. In particular, when you are bridging between J2EE and non-J2EE WebSphere MQ applications, use a WebSphere MQ export and WebSphere MQ JMS import (or vice versa) in conjunction with appropriate data bindings or mediation modules (or both).

Undelivered messages

If WebSphere MQ cannot deliver a message to its intended destination (because of configuration errors, for example), it sends the messages instead to a nominated dead-letter queue.

In doing so, it adds a dead-letter header to the start of the message body. This header contains the failure reasons, the original destination, and other information.

Handling exceptions

The way in which the binding is configured determines how exceptions that are raised by data handlers or data bindings are handled. Additionally, the nature of the mediation flow dictates the behavior of the system when such an exception is thrown.

A variety of problems can occur when a data handler or data binding is called by your binding. For example, a data handler might receive a message that has a corrupt payload, or it might try to read a message that has an incorrect format.

The way your binding handles such an exception is determined by how you implement the data handler or data binding. The recommended behavior is that you design your data binding to throw a `DataBindingException`.

The situation is similar for a data handler. Since the data handler is invoked by the data binding, any data handler exception is wrapped into a data binding exception. Therefore a `DataHandlerException` is reported to you as a `DataBindingException`.

EJB bindings

EJB import bindings enable SCA components to invoke services provided by J2EE business logic running on a J2EE server.

The EJB import binding invokes J2EE business logic using the EJB 2.1 programming model if your business logic is running on an EJB 2.1 server.

EJB bindings: a general perspective

The EJB import binding allows a Service Component Architecture (SCA) module to call a EJB implementation outside of the module.

EJB import bindings

EJB import bindings allow an SCA module to call EJB implementations by specifying the way that the consuming module is bound to the import component. For example:

- SCA component -> EJB 2.1 import == RMI/IIOP ==> EJB 2.1

Importing services from an external EJB implementation (an EJB implementation outside of the module) allows users to plug their business logic into the WebSphere Process Server environment and participate in a business process. Two usage scenarios for the EJB import binding are supported:

- One-way: a message is sent to the EJB specified in the import. No response is expected.
- Two-way (request-response): a message is sent to the EJB specified in the import. The response data from the method invocation is returned as a message to the calling component.

The EJB import contains the following major components. JAXWSDataHandler, EJBFaultSelector, and EJBImportFunctionSelector are provided by WebSphere Process Server.

Table 34. Major components of EJB import bindings

Component	**Purpose**
EJB implementation	The EJB which implements the business interface
JNDI Name	Locates the EJB reference
"JAX-WS data handler" on page 255	Performs the data transformation required for the server to turn Java objects into business objects (one-way) or response data into business objects (two-way). This data handler maps data from SCA to Java using JAX-WS and JAXB specifications. **Note:** To customize the mapping from the XSD type to Java types defined by the JAX-WS specification, change the JAXB annotations to fit your business need. The JAXWSDataHandler supports xs:any, xs:anyType, and xs:anySimpleType. Current support is restricted to the JAX-WS 1.0 and JAXB 2.0 specifications.
"EJB fault selector" on page 257	Determines if an EJB invocation has resulted in a fault, runtime exception, or a successful response. If a fault is detected, the fault selector returns the native fault name to the server so the JAXWSDataHandler can convert the exception object into a fault business object.

Table 34. Major components of EJB import bindings (continued)

Component	Purpose
"EJB import function selector" on page 257	Derives the EJB method type based on the WSDL method name invoked by an SCA component on the EJB import. If present, it uses the @WebMethod annotation to determine the correct Java method mapping for the WSDL method. If the @WebMethod annotation is missing, the function selector will return a Java method of the same name as the WSDL method name.

If your user scenario does not support JAX-WS and needs some custom mapping, then a custom data handler, function selector, and fault selector might be needed to perform the tasks otherwise completed by the JAXWSDataHandler, EJBFaultSelector, and EJBImportFunctionSelector. This includes the mapping normally completed by the custom mapping algorithm.

Data handler contract

In circumstances requiring data transformation, the contract between an EJB binding and the data handler is performed using a Java Object array (Object[]).

During outbound communications, the following steps take place:

1. The EJB binding sets the expected type, expected element, and targeted method name in the BindingContext to match those specified in the WSDL.
2. The EJB binding invokes the transform method for the data object requiring data transformation.
3. The data handler returns an Object[] representing the parameters of the method (in the order of their definition within the method).
4. The EJB binding uses the Object[] to invoke the method on the target EJB interface.

The binding prepares an Object[] to process the response from EJB invocation. The first element in the Object[] is the return value from the Java method invocation. The following values represent the input parameters for the method. This is required to support the In/Out and Out type of parameters. For parameters of the type Out, the value needs to be returned in the response data object. The data handler processes and transforms values found in the Object[] and then returns a response to the data object.

The data handler specified at the EJB binding level is used to perform request, response, fault and runtime exception processing. For faults, a specific data handler can be specified for each fault by specifying the faultBindingType configuration property. This overrides the value specified at the EJB binding level.

JAX-WS data handler

The JAX-WS data handler is used by default when the EJB binding has a WSDL interface. This data handler complies with the JAX-WS (Java API for XML-based WebService) specification and is used to convert the WSDL method invocation to the Java method invocation for import (outbound processing). It also handles the conversion of Java exceptions to fault data objects for the purposes of fault handling and runtime exceptions when the fault selector returns exceptions of

ResponseType.RUNTIME. This data handler cannot be used to transform a SOAP message representing a JAX-WS invocation to a data object.

The data handler supports xs:AnyType, xs:AnySimpleType, and xs:Any along with other XSD data types. To enable support for xs:Any, use the @XmlAnyElement (lax=true) for the bean property in the Java code. For example,

```
public class TestType {
 private Object[] object;

 @XmlAnyElement (lax=true)
 public Object[] getObject() {
  return object;
 )

 public void setObject (Object[] object) {
  this.object=object;
 )
)
```

This makes the property object in TestType an xs:any field. The Java class value used in the xs:any field should have the @XmlAnyElement annotation. For example, if Address is the Java class being used to populate the object array, the Address class should have the annotation @XmlRootElement.

The following restrictions are applicable for the JAX-WS data handler:

- No support for the header attribute @WebParam annotation.
- The namespace for business object schema files (XSD files) does not include default mapping from the Java package name. The annotation **@XMLSchema** in package-info.java also does not work. The only way to create an XSD with a namespace is to use the @XmlType and XmlRootElement annotations. @XmlRootElement defines the target namespace for the global element in Java bean types.
- The EJB import wizard does not create XSD files for unrelated classes. Version 2.0 does not support the @XmlSeeAlso annotation, so if the child class is not referenced directly from the parent class an XSD is not created. The solution to this problem is to run SchemaGen for such child classes. SchemaGen is a command line utility provided to create XSD files for a given Java bean. This is located in *WPS_Install_Home*/bin directory. These XSDs must be manually copied to the business integration module for the solution to work.

Fault handler contract

The contract between an EJB binding and the fault handler is performed using a Java object array (Object[]).

On successful import, the binding assembles an Object[] The first element in the Object[] is the return value from the Java method invocation. The following values represent the input parameters for the method. This is required to support the In/Out and Out type of parameters.

For exception scenarios, the binding assembles an Object[] and the first element represents the exception thrown by the method. The fault selector can return any of the following values:

Table 35. Fault handler return values

Type	Return value
Fault	ResponseType.FAULT
Runtime exceptions	ResponseType.RUNTIME
Normal response	ResponseType.RESPONSE

If the fault selector returns a value of ResponseType.FAULT, the getFaultName() should return the native fault name. This native fault name is used by the binding to determine the corresponding WSDL fault name from the model and invoke the correct fault data handler.

EJB fault selector

This fault selector follows the fault selector contract with the EJB import binding. It returns the ResponseType.BUSINESSFAULT when the passed Object[] contains an exception object. If the exception object does not match any of the declared exception types on the method it returns ResponseType.RUNTIME_EXCEPTION. Otherwise, it returns ResponseType.RESPONSE.

Import function selector contract

The contract between an EJB binding and the import function selector is performed using a JAVA Object array (Object[]).

The first element in the Object[] is the name of the WSDL. The second element in the Object[] is the input business object. The function selector returns a java.lang.reflect.Method object that represents the method of the EJB interface.

EJB import function selector

This function selector follows the function selector contract with the EJB import binding. The first element in the Object[] is a Java method with the WSDL name. It follows JAX-WS annotations to derive the Java method for a given WSDL method and uses the annotations in the input data object to derive the name of the Java class that defines the EJB interface.

Working with EJB bindings

EJB imports can be managed in the server. EJB imports are configurable in the administrative console.

For more information on creating EJB imports, see the section devoted to Working with EJB bindings in the WebSphere Integration Developer information center.

Administering EJB bindings

You can configure EJB import bindings using the WebSphere administrative console.

Before you begin

To see or edit an EJB binding, it must be installed as part of a Service Component Architecture (SCA) application in your server profile.

About this task

Configuring EJB import bindings

EJBs invoked by an EJB import may be running in any of the following combinations. For each one of these scenarios it's very important to specify the correct JNDI names as follows:

Table 36. EJB import JNDI name configurations

SLSB scenario	JNDI configuration information
WebSphere Process Server in a different J2EE module	Set the JNDI name in the EJB import binding to match the global namespace. Also, confirm that the JNDI name specified in the EJB import binding matches what is specified in the J2EE module bindings file.
Remote WebSphere Process Server or WebSphere Application Server	Create a namespace binding (of EJB binding type) using the WebSphere Process Server administrative console. This can be found in **Environment → Naming → Namespace**. The name specified in the **namespace** field for the namespace binding should match the JNDI name specified in the EJB import binding configuration.
Remote J2EE server (other than WebSphere Process Server or WebSphere Application Server)	Create a namespace binding of type CORBA if the J2EE server provides a COSNaming interface or of the indirect type if it does not using the WebSphere Process Server administrative console. This can be found in **Environment → Naming → Namespace**. The name specified in the **namespace** field for the namespace binding should match the JNDI name specified in the EJB import binding configuration.

If your implementation involves WebSphere Application Server additional configuration using the WebSphere Application Server administrative console may be required.

Note: All of the properties for EJB binding import except JNDI name are read only. DataHandler, FunctionSelector, and FaultSelector properties are viewable, but they cannot be configured.

To view or configure EJB import properties using the WebSphere Process Server administrative console, complete the following steps:

Procedure

1. Select the SCA module. From the administrative console, click **Applications → SCA Modules** and then click the *modulename*.
2. Under **Module components**, expand **Imports**. This reveals a list of the installed imports.
3. Expand the import and then expand **Binding**. This reveals a list of available bindings to administer.
4. Click on the binding you want to examine or edit.

External clients

WebSphere Process Server uses EJB import bindings to send messages to an external client such as a web portal or enterprise information system (EIS).

When a message is sent from the server to an external client, it's the job of the EJB import to specify the way that the consuming module is bound to the import component. When invoked, the EJB import calls the enterprise Java bean (EJB) and for each method invocation on the EJB import a corresponding method in the WSDL (defined by JAX-WS mapping) is invoked on the EJB.

HTTP bindings

The HTTP binding is designed to provide Service Component Architecture (SCA) connectivity to HTTP. This allows existing or newly-developed HTTP applications to participate in Service Oriented Architecture (SOA) environments.

Hypertext Transfer Protocol (HTTP) is a widely-used protocol for transferring information on the Web. When you are working with an external application that uses the HTTP protocol, an HTTP binding is necessary. The HTTP binding handles the transformation of data passed in as a message in a native format to a business object in an SCA application. The HTTP binding also can transform data passed out as a business object to the native format expected by the external application. for an incoming messaging.

Some common scenarios for using the HTTP binding are described in the following list:

- SCA-hosted services can invoke HTTP applications using an HTTP import.
- SCA-hosted services can expose themselves as HTTP-enabled applications, allowing them to be used by HTTP clients, using an HTTP export.
- WebSphere Process Server and WebSphere Enterprise Service Bus can communicate between themselves across an HTTP infrastructure, allowing users to manage their communications according to corporate standards.
- WebSphere Process Server and WebSphere Enterprise Service Bus can act as mediators of HTTP communications, transforming and routing messages, which improves the integration of applications using a HTTP network.
- WebSphere Process Server and WebSphere Enterprise Service Bus can be used to bridge between HTTP and other protocols, such as SOAP/HTTP Web services, Java Connector Architecture (JCA)-based resource adapters, JMS, and so on.

Detailed information about creating HTTP import and export bindings can be found in the WebSphere Integration Developer information center. See the **Developing integration applications** → **Accessing external services with HTTP**> topics.

HTTP bindings: a general perspective

The HTTP binding provides connectivity to HTTP-hosted applications. It mediates communication between HTTP applications and allows existing HTTP-based applications to be called from a module.

HTTP import bindings

The HTTP import binding provides outbound connectivity from Service Component Architecture (SCA) applications to an HTTP server or applications.

The import invokes an HTTP endpoint URL. The URL can be specified in one of two ways:

- The URL can be specified as a configuration property on the import.
- The URL can be set dynamically in the HTTP headers.

This invocation is always synchronous in nature.

Although HTTP invocations are always request-reply, the HTTP import supports both one-way and two-way operations and ignores the response in the case of a one-way operation.

HTTP export bindings

The HTTP export binding provides inbound connectivity from HTTP applications to an SCA application.

A URL is defined on the HTTP export. HTTP applications that want to send request messages to the export use this URL to invoke the export.

The HTTP export also supports pings.

HTTP bindings at runtime

An import with an HTTP binding at runtime sends a request with or without data in the body of the message from the SCA application to the external Web service. The request is made from the SCA application to the external Web service, as shown in Figure 29.

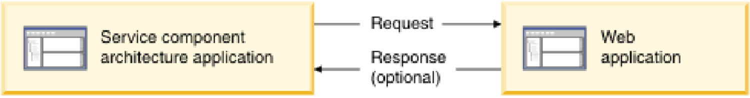


Figure 29. Flow of a request from the SCA application to the Web application

Optionally, the import with the HTTP binding can receive data back from the Web application in a response to the request.

With an export, the request is made by a client application to a Web service, as shown in Figure 30.

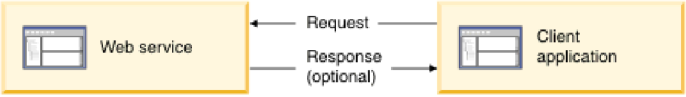


Figure 30. Flow of a request from the Web service to the client application.

The Web service is a Web application running on the server. The export is implemented in that Web application as a servlet so the client sends its request to a URL address. The servlet passes the request to the SCA application in the runtime.

Optionally, the export may send data to the client application in response to the request.

Administering HTTP bindings

You can administer HTTP import and export bindings to tune, or set, special features of the resource. The administrative tasks are performed using the administrative console.

You use WebSphere Integration Developer to create HTTP imports and exports.

Displaying HTTP bindings

After deploying an application, you may want to examine the HTTP bindings to make sure they are correct.

Before you begin

You must be at the administrative console.

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

About this task

Display HTTP bindings as the first step in changing any existing HTTP bindings.

Procedure

1. Display the Service Component Architecture (SCA) applications. Expand **Applications** and click **SCA Modules**.

   The system displays the installed applications.
2. Click the ***module_name*** for which you want to display the bindings.

   The system displays the module configuration.
3. Expand **Imports** or **Exports** depending on which binding you are displaying.
4. Expand the *path* and then expand **Bindings** to display the bindings for the imports or exports contained by the module.

   The system displays all of the contained import or export bindings.
5. Click the binding name to display the binding configuration.

Results

The system displays the configuration of the binding for the import or export you selected.

What to do next

Change the import or export, if needed.

Changing HTTP export bindings

The administrative console allows you to change the configuration of HTTP export bindings without changing the original source and then redeploying the application.

Before you begin

Display the HTTP bindings for the module as described in "Displaying HTTP bindings."

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

About this task

Change HTTP export bindings when you need to change whether a method on a binding is pingable or to change the encodings a method or binding supports.

Procedure

1. Display the module bindings as described in "Displaying HTTP bindings."
2. Select the binding you are changing. Click on the binding name.
3. Change the binding configuration.

Option	Description
To change the configuration on the binding scope	Click the **Binding Scope** tab.
To change the configuration at the method scope	Click the **Method Scope** tab.

   When both configurations exist, the method scope configuration takes precedence over the binding scope configuration.
4. Change the configuration and click **Apply**, to remain on the same page, or **OK**, to return to the previous page.

Results

The configuration for the binding is changed.

Restriction: If the module is redeployed, the configuration is replaced with the new configuration. To ensure this change remains with the module across deployments, you must make the change in the source code for the module using WebSphere Integration Developer.

Changing HTTP import bindings

The administrative console allows you to change the configuration of HTTP import bindings without changing the original source and then redeploying the application.

Before you begin

Display the HTTP bindings for the module as described in "Displaying HTTP bindings."

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

About this task

Change HTTP import bindings when the binding properties of an HTTP application used by a Service Component Architecture (SCA) module change.

Procedure

1. Display the module bindings as described in "Displaying HTTP bindings."

2. Select the binding you are changing. Click on the binding name.
3. Change the binding configuration.

Option	Description
To change the configuration on the binding scope	Click the **Binding Scope** tab.
To change the configuration at the method scope	Click the **Method Scope** tab.

   When both configurations exist, the method scope configuration takes precedence over the binding scope configuration.
4. Change the configuration and click **Apply**, to remain on the same page, or **OK**, to return to the previous page.

Results

The configuration for the binding is changed.

Restriction: If the module is redeployed, the configuration is replaced with the new configuration. To ensure this change remains with the module across deployments, you must make the change in the source code for the module using WebSphere Integration Developer.

HTTP headers

HTTP import and export bindings allow configuration of HTTP headers and their values to be used for outbound messages. The HTTP import uses these headers for requests, and the HTTP export uses them for responses.

Statically configured headers and control information take precedence over values dynamically set at runtime. However, the dynamic override URL, Version, and Method control values override the static values, which are otherwise considered defaults.

The binding supports the dynamic nature of the HTTP import URL by determining the value of HTTP target URL, Version, and Method at runtime. These values are determined by extracting the value of Endpoint Reference, Version, and Method.

- For Endpoint Reference, use com.ibm.websphere.sca.addressing.EndpointReference APIs or set the /headers/SMOHeader/Target/address field in the SMO header.
- For Version and Method, use the HTTP control parameters section of the Service Component Architecture (SCA) message.

The control and header information for outbound messages under HTTP export and import bindings is processed in the following order:

1. Header and control information excluding HTTP dynamic override URL, Version, and Method from the SCA Message. (lowest priority)
2. Changes from the administrative console on the export/import level
3. Changes from the administrative console on the method level of the export/import
4. HTTP dynamic override URL from Endpoint Reference or the SMO header
5. Version and Method from the Service Component Architecture (SCA) message
6. Headers and control information from the data handler or data binding (highest priority)

The HTTP export and import will only populate inbound direction headers and control parameters with data from the incoming message (HTTPExportRequest and HTTPImportResponse) if contextPropagationEnabled is set to `True`. Inversely, the HTTP export and import will only read and process outbound headers and control parameters (HTTPExportResponse and HTTPImportRequest) if contextPropagationEnabled is set to `True`.

Note: Data handler or data binding changes to headers or control parameters in the import response or export request will not alter the processing instructions of the message inside the import or export binding and should be used only to propagate modified values to downstream SCA components.

The context service is responsible for propagating the context (including the protocol headers, such as the HTTP header, and the user context, such as account ID) along a Service Component Architecture (SCA) invocation path. During development in WebSphere Integration Developer, you can set the context service on import and export properties. For more details, see the import and export bindings information in the WebSphere Integration Developer information center.

Supplied HTTP header structures and support

Table 37 itemizes the request/response parameters for HTTPImport and HTTPExport requests and responses.

Table 37. Supplied HTTP header information

Control name	HTTPImport request	HTTPImport response	HTTPExport request	HTTPExport response
URL	Ignored	Not set	Read from the request message. **Note:** Query string is also part of the URL control parameter.	Ignored
Version (possible values: 1.0, 1.1; default is 1.1)	Ignored	Not set	Read from the request message	Ignored
Method	Ignored	Not set	Read from the request message	Ignored
Dynamic Override URL	If set in the data handler or data binding, overrides the HTTP Import URL. Written to the message in the request line. **Note:** Query string is also part of the URL control parameter.	Not set	Not set	Ignored

Table 37. Supplied HTTP header information (continued)

Control name	HTTPImport request	HTTPImport response	HTTPExport request	HTTPExport response
Dynamic Override Version	If set, overrides the HTTP Import Version. Written to the message in the request line.	Not set	Not set	Ignored
Dynamic Override Method	If set, overrides the HTTP Import Method. Written to the message in the request line.	Not set	Not set	Ignored
Media Type (This control parameter carries part of the value of the Content-Type HTTP header.)	If present, written to the message as part of the Content-Type header. **Note:** This control element value should be provided by the data handler or data binding.	Read from the response message, Content-Type header	Read from the request message, Content-Type header	If present, written to the message as part of Content-Type header. **Note:** This control element value should be provided by the data handler or data binding.
Character set (default: UTF-8)	If present, written to the message as part of the Content-Type header. **Note:** This control element value should be provided by the data binding.	Read from the response message, Content-Type header	Read from the request message, Content-Type header	Supported; written to the message as part of the Content-Type header. **Note:** This control element value should be provided by the data binding.
Transfer Encoding (Possible values: chunked, identity; default is identity)	If present, written to the message as a header and controls how the message transformation is encoded.	Read from the response message	Read from the request message	If present, written to the message as a header and controls how the message transformation is encoded.
Content Encoding (Possible values: gzip, x-gzip, deflate, identity; default is identity)	If present, written to the message as a header and controls how the payload is encoded.	Read from the response message	Read from the request message	If present, written to the message as a header and controls how the payload is encoded.
Content-Length	Ignored	Read from the response message	Read from the request message	Ignored

Table 37. Supplied HTTP header information (continued)

Control name	HTTPImport request	HTTPImport response	HTTPExport request	HTTPExport response
StatusCode (default: 200)	Not supported	Read from the response message	Not supported	If present, written to the message in the response line
ReasonPhrase (default: OK)	Not supported	Read from the response message	Not supported	Control value ignored. The message response line value is generated from the StatusCode.
Authentication (contains multiple properties)	If present, used to construct the Basic Authentication header. **Note:** The value for this header will be encoded only on the HTTP protocol. In the SCA, it will be decoded and passed as clear text.	Not applicable	Read from the request message Basic Authentication header. The presence of this header does not indicate the user has been authenticated. Authentication should be controlled in the servlet configuration. **Note:** The value for this header will be encoded only on the HTTP protocol. In the SCA, it will be decoded and passed as clear text.	Not applicable
Proxy (contains multiple properties: Host, Port, Authentication)	If present, used to establish connection through proxy.	Not applicable	Not applicable	Not applicable
SSL (contains multiple properties: Keystore, Keystore Password, Trustore, Trustore Password, ClientAuth)	If populated and the destination url is HTTPS, it is used to establish a connection through SSL.	Not applicable	Not applicable	Not applicable

The WebSphere Transformation Extender data handler

Data that is brought into an SCA module by way of an export can be in a wide range of formats, such as COBOL or EDI. The data must be converted into a data object before processing with any server component. Similarly, data passing from your server through an import must be converted to a format understood by the external system. WebSphere Transformation Extender can be used to perform these transformations.

The exports and imports at the edges of your modules are responsible for converting native data to data objects and vice versa. Exports and imports contain data handlers and data bindings for this purpose.

To transform native data in your imports and exports to a variety of data formats, you can use a prepackaged data handler or data binding, you can write your own data binding, or you can use the WebSphere Transformation Extender data handler to convert native data to data object, and vice versa, at the edges of your modules.

WebSphere Transformation Extender is a transaction-oriented data integration solution that automates the transformation of high-volume, complex transactions without the need for hand-coding. It supports real-time integration of data from multiple applications, databases, messaging middleware, and communications technologies across the enterprise. WebSphere Transformation Extender provides a metadata-driven, tool-oriented approach to data transformation.

WebSphere Transformation Extender is a separately licensed product and is not included as part of WebSphere Process Server.

For WebSphere Process Server to work with WebSphere Transformation Extender, you must use the WebSphere Transformation Extender installer for WebSphere Process Server.

The schema below represents configuring your export to use WebSphere Transformation Extender with the WebSphere Transformation Extender data handler to convert the incoming data to a data object.

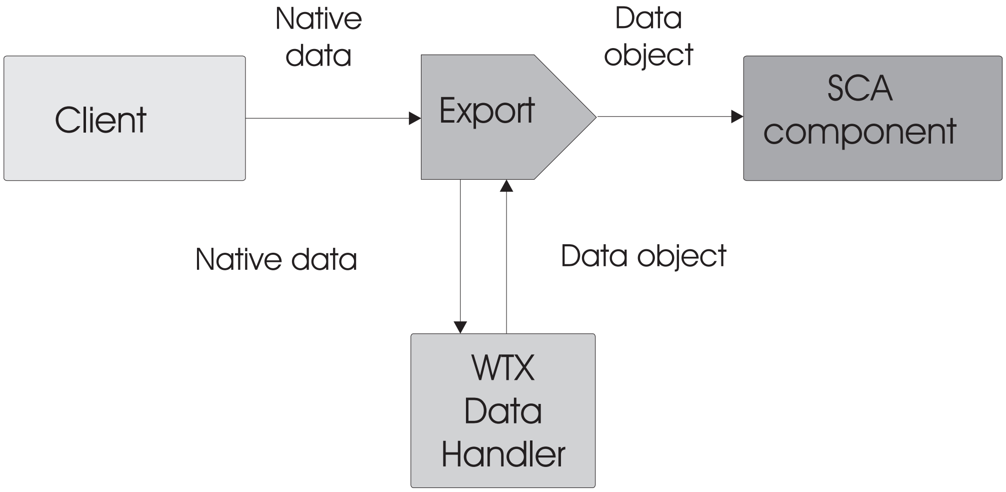


Figure 31. An export configured to use the WTX data handler

The client delivers the data to the export in some native format. The export then passes this data in the same native format to WebSphere Transformation Extender by way of the WebSphere Transformation Extender Data Handler. WebSphere Transformation Extender converts the data to data object format and returns it to the export. The export passes the data object on to the relevant SCA component.

The schema below represents configuring your import to use WebSphere Transformation Extender with the WebSphere Transformation Extender Data Handler to convert the outgoing data from data object format into the native format required by the client.

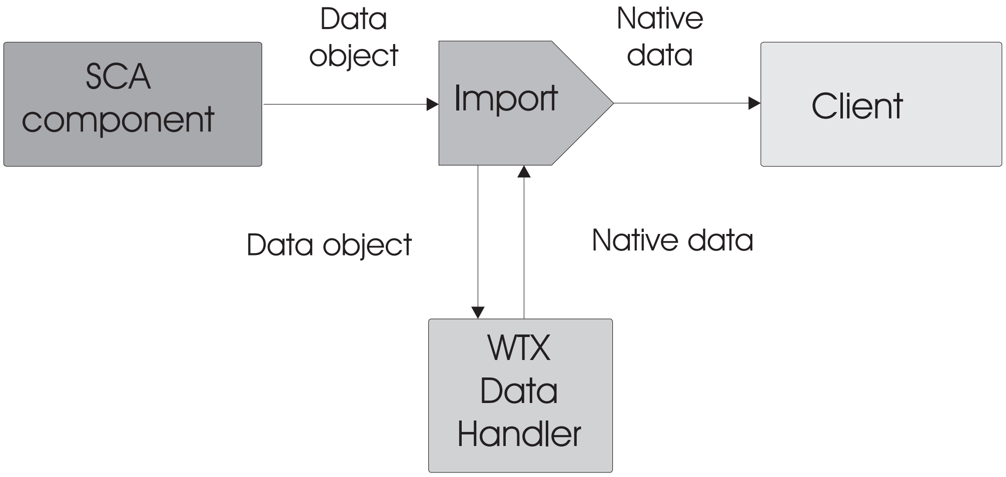


Figure 32. An import configured to use the WTX data handler

The SCA component sends a data object to the import. The import passes this data object to WebSphere Transformation Extender by way of the WebSphere Transformation Extender Data Handler. WebSphere Transformation Extender converts the information to the native format of the client, and this information is returned to the import. The import sends the native data to the client.

Note: For more information about using WebSphere Transformation Extender, see the WebSphere Transformation Extender product library.

Support for the WebSphere Transformation Extender data handler

WebSphere Transformation Extender can be used to convert data between a variety of formats and the data object required by the server. The details of when such a data handler is supported are provided.

Supported platforms

The WebSphere Transformation Extender data handler is supported only on platforms that are supported by both WebSphere Transformation Extender and WebSphere Process Server. See the supported platform lists for both products here: WebSphere Transformation Extender system requirements and WebSphere Process Server.

Note: WebSphere Transformation Extender is a separately licensed product and is not included as part of WebSphere Process Server.

i5/OS WebSphere Transformation Extender is not supported on i5/OS and so this solution does not apply to the i5/OS platform.

WebSphere Transformation Extender must be at version 8.2 or above.

Imports and exports

The following imports and exports support the use of WebSphere Transformation Extender for data transform:

- JMS
- Generic JMS
- WebSphere MQ JMS
- Native MQ (body data binding only)
- EIS Flat File
- EIS FTP
- EIS Email
- HTTP

The WebSphere Transformation Extender data handler is applicable to all imports and exports listed above. These data handlers can be customized. See the WebSphere Integration Developer documentation for more information.

WebSphere Transformation Extender maps and the data handler

The WebSphere Transformation Extender data handler invokes WebSphere Transformation Extender. WebSphere Transformation Extender uses a map to convert data from one format to another. The map provides the system with the details of the conversion. You can utilize existing WebSphere Transformation Extender maps if they are applicable to the solution.

The WebSphere Transformation Extender data handler invokes one of your WebSphere Transformation Extender maps in the context of a WebSphere Transformation Extender Engine. The data handler sends the WebSphere Transformation Extender map name and input data to the WebSphere Transformation Extender engine, which performs the conversion from one format to another. The result of the transformation is sent back to the data handler, which publishes it to the export or import.

On the export side, the WebSphere Transformation Extender data handler invokes a WebSphere Transformation Extender map to transform native data to XML. The data handler then deserializes the XML to a data object. An import also uses the data handler in this way on the response message in a request/response scenario.

On the import end, the data handler serializes a data object to XML and feeds it to a WebSphere Transformation Extender map. The map converts it to native data, which is then published. An export also uses the data handler in this way on the response message.

For more information on creating WebSphere Transformation Extender maps, refer to the WebSphere Transformation Extender documentation.

For more information on creating map naming conventions, usage of maps in SCA modules and configuring the WebSphere Transformation Extender data handler, refer to the WebSphere Integration Developer documentation.

Considerations before using the WebSphere Transformation Extender data handler

The WebSphere Transformation Extender data handler is an ideal choice when you have non-XML data coming into, or leaving, your WebSphere Process Server. Note that you must have WebSphere Transformation Extender available.

You must install WebSphere Process Server before you install the WebSphere Transformation Extender for WebSphere Process Server. You must have a valid license for WebSphere Transformation Extender. WebSphere Transformation Extender is an independent product and has to be installed separately.

For your server to work with WebSphere Transformation Extender, you must use the WebSphere Transformation Extender installer for WebSphere Process Server. In addition to installing WebSphere Transformation Extender, this process also installs the WebSphere Transformation Extender Java client libraries as an OSGi bundle in the WebSphere Process Server product so that it is accessible to WebSphere Process Server.

Installing WebSphere Transformation Extender for WebSphere Process Server is a two-step process:

1. Run the WebSphere Transformation Extender for the Software Developers Kit (SDK). This installs the SDK and the OSGi bundle for WebSphere Transformation Extender in the WebSphere Process Server plug-ins.
2. Optional: Run the installer for WebSphere Transformation Extender Design Studio. This step provides you with all the tools required to design and create tree-types and maps.

Note: You must perform the first step on every node where you will utilize the WebSphere Transformation Extender data handler. The second step is required only on nodes where you want to create and edit maps.

The installation of WebSphere Transformation Extender should have configured the system path to point to WebSphere Transformation Extender. Verify that the installation location (for example, C:\Program Files\IBM\WebSphere Transformation Extender 8.2) is part of your system path.

The WebSphere Transformation Extender data handler calls WebSphere Transformation Extender using Java Native Interface (JNI). This may have performance implications on your system.

The WebSphere Transformation Extender data handler has the following memory requirements:

- For transforming from native data to business object, the memory required is at least twice the size of the native data plus twice the size of the serialized business object.
- For transforming from business object to native data, the memory required is at least twice the size of the serialized business object plus twice the size of native data.

The WebSphere Transformation Extender data handler is an ideal choice when you have non-XML data entering, or leaving, your WebSphere Process Server environment. For XML data, you should use the XML data handler for JMS, WebSphere MQ, and HTTP imports and exports, and use the XMLDataHandler for EIS bindings.

Additional considerations for using the WebSphere Transformation Extender data handler.

- If your data is in an industry standard format supported by WebSphere Transformation Extender, then the WebSphere Transformation Extender data handler is an ideal solution.
- For formats such as delimited, csv, and fixed-width, use the data bindings and data handlers that are supplied as part of WebSphere Process Server rather than the WebSphere Transformation Extender data handler.
- For COBOL, use the COBOL data binding generator to generate a COBOL data binding.
- For any other custom format, make an evaluation based on the memory requirements and the ease of developing a WebSphere Transformation Extender map compared to developing a custom data binding.

Setting the data binding descriptor

Use the WebSphere Transformation Extender data binding descriptor instead of the configuration properties of the WebSphere Transformation Extender data handler when you want to configure the WebSphere Transformation Extender data handler at an instance level. You must set the WebSphere Transformation Extender data binding descriptor in specific places for the various imports and exports.

About this task

The key to the invocation of WebSphere Transformation Extender is to determine the map name to be invoked. You can do this by configuring the properties of the WebSphere Transformation Extender data handler in WebSphere Integration Developer. The map set in the properties of the binding is then applicable to all instances of data that enter the export or import for that operation. For example, consider an export with an operation `Create(Customer)`. If you want every invocation of this operation to use the same WebSphere Transformation Extender map for each customer, then configuring the properties of the WebSphere Transformation Extender data handler in WebSphere Integration Developer is an appropriate choice.

However, you might want to invoke a different map depending on the incoming customer object. Such a scenario cannot be configured in the properties of the data handler and you must instead use the data binding descriptor.

You must put the WebSphere Transformation Extender data binding descriptor in the following locations, depending on what type of import or export you are employing:

- The WebSphere Transformation Extender data handler for JMS expects the descriptor to be set in the **JMSType** or in a **JMS property** called **DataBindingDescriptor**.

  For JMS Exports:

  – For a request, the client is required to either set the **JMSType** in the incoming message or set a **JMS property** called **DataBindingDescriptor**.

    These properties are set when the client sends a message to your server.

  – For a response, you cannot configure the binding at the instance level; you must use configuration properties to invoke WebSphere Transformation Extender.

  For JMS Imports:

  - For a request, you cannot configure the binding at the instance level; you must use configuration properties to invoke WebSphere Transformation Extender.
  - For a response, the client is required to either set the **JMSType** in the incoming message or set a **JMS property** called **DataBindingDescriptor**.

    These properties are set when the client sends a message to your server.
- The WebSphere Transformation Extender Native MQ data handler expects the descriptor to be set in the MQRFH2 headers.

  For Native MQ Exports:
  - For a request, the client must set the **DatabindingDescriptor** property in the MQRFH2 header.
  - For a response, you cannot configure the binding at the instance level; you must use configuration properties of the binding itself to invoke WebSphere Transformation Extender.

  For Native MQ Imports:
  - For a request, you cannot configure the binding at the instance level; you must use configuration properties of the binding itself to invoke WebSphere Transformation Extender.
  - For a response, the client must set the **DatabindingDescriptor** property in the MQRFH2 header.
- The WebSphere Transformation Extender HTTP data handler expects a custom header called **DatabindingDescriptor**. If the header is not present, the binding looks in the parameters of the URL of the BusinessObject and StreamType or the map name parameter. For example: https://host[*port*]/*sca module name*/*exportname*/?businessObject=*value1*&contentType=*value2*

  For HTTP Exports:
  - For a request, the custom header **DatabindingDescriptor** must be set by the client. Alternatively, you can set custom properties on the URL.
  - For a response, you cannot configure the binding at the instance level and must use configuration properties of the binding itself to invoke WebSphere Transformation Extender.

  For HTTP Imports:
  - For a request, you cannot configure the binding at the instance level and must use configuration properties of the binding itself to invoke WebSphere Transformation Extender.
  - For response, the custom header **DatabindingDescriptor** must be set by the client.

WebSphere Transformation Extender data binding descriptor

The WebSphere Transformation Extender data binding descriptor is a URI that describes the incoming message.

Purpose

The WebSphere Transformation Extender data binding descriptor is set by the client in the headers of the incoming message. It provides the information required to associate the incoming message with the correct map in the WebSphere Transformation Extender.

Using the WebSphere Transformation Extender data binding descriptor is one of two ways that you can set the invocation of the map. The alternative is to configure the properties of the WebSphere Transformation Extender data handler in

WebSphere Integration Developer. However, you should use the WebSphere Transformation Extender data binding descriptor when you want to use a different map for each invocation of the operation.

For example, consider an export with an operation `Create(Customer)`. If you want every invocation of this operation to use the same WebSphere Transformation Extender map for each customer, then configuring the properties of the WebSphere Transformation Extender data handler in WebSphere Integration Developer is an appropriate choice.

However, you might want to invoke a different map depending on the customer. In this case the operation is invoked with customer and uses a different map for every invocation. Such a scenario cannot be configured in the properties of the data binding and you must instead use the data binding descriptor.

Syntax

The WebSphere Transformation Extender data binding descriptor has the following syntax

databinding://*domain***/***property***?***queryParameters*

domain is **WTX** for this data binding.

property has the value **map** in this case.

queryParameters is either name=*mapname* where *mapname* is the name of the map required, or businessObject=*Customer*&contentType=*format*, where *Customer* is the name of the Business Object and *format* is the format of the data stream (for example, COBOL, EDI, and so on).

Sample

Examples of a WebSphere Transformation Extender data binding descriptor are shown below:

```
databinding://WTX/map?name=WTX/CustomerToCOBOL
```

Note: A directory called WTX must be in the top level of the module and must contain any maps that are required by this binding.

```
databinding://WTX/map?businessObject=Customer&contentType=COBOL
```

```
databinding://WTX/map?contentType=COBOL
```

Setting the binding descriptor for an EIS binding

The data binding descriptor cannot be set for an EIS import or export binding. If you are using EIS imports and exports, you must configure the interaction between your binding and WebSphere Transformation Extender using the configuration properties within the EIS binding definition. You cannot configure the connection at the instance level.

Setting the binding descriptor for a JMS Import binding

A JMS import binding can be passed the WebSphere Transformation Extender data binding descriptor in either the JMSType or JMSProperties part of the message.

Before you begin

Use the WebSphere Transformation Extender JMS data binding for JMS, Generic JMS, and MQ JMS imports.

You can set the WebSphere Transformation Extender data binding descriptor only on the response message for a JMS Import.

About this task

Set the data binding descriptor on the response message for a JMS import when you want to configure the messages individually.

If you do not need to configure messages individually, it is recommended that you use the custom properties within the binding itself to make the association with WebSphere Transformation Extender.

Procedure

1. The request message cannot be configured at the instance level. You must use the custom properties of the binding to make the association with the specific WebSphere Transformation Extender maps that you want to use.
2. Configure the response message. There are two ways to configure the response:

Option	Example

Set the JMSType in the incoming message	`jmsMessage.setJMSType("databinding:`
	`//WTX/map?businessObject=Customer&contentType=COBOL");`
	`Note that line breaks have been added to this example.`

Set the JMS property called DataBindingDescriptor	`jmsMessage. setStringProperty`
	`("DataBindingDescriptor","databinding:`
	`//WTX/map?businessObject=Customer&contentType=COBOL");`
	`Note that line breaks have been added to this example.`

These properties are set when the client sends a message to WebSphere Process Server.

Setting the binding descriptor for a JMS Export binding

A JMS export binding can be passed the WebSphere Transformation Extender data binding descriptor in either the JMSType or JMSProperties part of the message.

Before you begin

Use the WebSphere Transformation Extender JMS data binding for JMS, Generic JMS, and MQ JMS exports.

You can set the WebSphere Transformation Extender data binding descriptor only on the response message for a JMS export.

About this task

Set the data binding descriptor on the response message for a JMS export when you want to configure the messages individually.

If you do not need to configure messages individually, it is recommended that you use the custom properties within the binding itself to make the association with WebSphere Transformation Extender.

Procedure

1. Configure the request message. There are two ways to configure the request message:

Option	Example

Set the JMSType in the incoming message	`jmsMessage.setJMSType("databinding:`
	`//WTX/map?businessObject=Customer&contentType=COBOL");`
	`Note that line breaks have been added to this example.`

Set a JMS property called DataBindingDescriptor	`jmsMessage. setStringProperty`
	`("DataBindingDescriptor", "databinding:`
	`//WTX/map?businessObject=Customer&contentType=COBOL");`
	`Note that line breaks have been added to this example.`

2. The response message cannot be configured at the instance level. In order to use a WebSphere Transformation Extender transformation on a JMS export response message, you must set the configuration properties of your export.

Setting the binding descriptor for a native MQ Import

If you want to configure each message to interact in a specific way with WebSphere Transformation Extender, you must configure the message to use the WebSphere Transformation Extender MQ data binding via the WebSphere Transformation Extender data binding descriptor.

Before you begin

Use the WebSphere Transformation Extender MQ data binding for native MQ imports.

You can set the WebSphere Transformation Extender data binding descriptor only on the response message for a native MQ Import.

About this task

Set the data binding descriptor on the response message for a native MQ import when you want to configure the messages individually.

If you do not need to configure messages individually, it is recommended that you use the custom properties within the binding itself to make the association with WebSphere Transformation Extender.

Procedure

1. The request message cannot be configured at the instance level. You must use the custom properties of the binding to make the association with the specific WebSphere Transformation Extender maps that you want to use.
2. Configure the response message. Set the DataBindingDescriptor in the MQRFH2 header of the incoming message

Results

These properties are set when the MQ client sends a message to WebSphere Process Server.

Setting the binding descriptor for a native MQ Export

If you want to configure each message to interact in a specific way with WebSphere Transformation Extender, you must configure the message to use the WebSphere Transformation Extender MQ data binding via the WebSphere Transformation Extender data binding descriptor.

Before you begin

Use the WebSphere Transformation Extender MQ data binding for native MQ exports.

You can set the WebSphere Transformation Extender data binding descriptor only on the request message for a native MQ export.

About this task

Set the data binding descriptor on the request message for a native MQ export when you want to configure the messages individually.

If you do not need to configure messages individually, it is recommended that you use the custom properties within the binding itself to make the association with WebSphere Transformation Extender.

Procedure

1. Configure the request message. Set the DataBindingDescriptor in the MQRFH2 header of the incoming message
2. The request message cannot be configured at the instance level. You must use the custom properties of the binding to make the association with the specific WebSphere Transformation Extender maps that you want to use.

Results

These properties are set when the MQ client sends a message to WebSphere Process Server.

Setting the binding descriptor for an HTTP Import

If you want to configure each HTTP message that is sent to the same operation on the import to use a different WebSphere Transformation Extender map, you must configure the message to use the WebSphere Transformation Extender data binding descriptor.

Before you begin

Use the WebSphere Transformation Extender HTTP data binding for HTTP imports.

You can set the WebSphere Transformation Extender data binding descriptor only on the response message for an HTTP import.

About this task

Set the data binding descriptor on the response message for an HTTP import when you want to configure the messages individually.

If you do not need to configure messages individually, it is recommended that you use the custom properties within the binding itself to make the association with WebSphere Transformation Extender.

Procedure

1. The request message cannot be configured at the instance level. You must use the custom properties of the binding to make the association with the specific WebSphere Transformation Extender maps that you want to use.
2. Configure the response message.

   Create a custom HTTP header called DataBindingDescriptor in the HTTP import. For example:

   ```
   databinding://WTX/map?businessObject=Customer&contentType=EDI
   ```

Setting the binding descriptor for an HTTP Export

If you want to configure each HTTP message that is published to the same operation on the export to use a different WebSphere Transformation Extender map, you must configure the message to use the WebSphere Transformation Extender data binding descriptor.

Before you begin

Use the WebSphere Transformation Extender data binding for HTTP exports.

You can set the WebSphere Transformation Extender data binding descriptor only in the request message for an HTTP export.

The HTTP client sending requests to the HTTP export sets the data binding descriptor in the request message header or on the URL.

About this task

Set the data binding descriptor on the request message for an HTTP export when you want to configure the messages individually.

If you do not need to configure messages individually, it is recommended that you use the custom properties within the binding itself to make the association with WebSphere Transformation Extender.

Procedure

1. Configure the request message. There are two ways to set the data binding descriptor in the request message:

Option	Example
Create a custom HTTP header called DataBindingDescriptor in the HTTP export.	databinding://WTX/ map?businessObject=Customer& contentType =EDI
Add additional query parameters on the URL.	http://www.ibm.com/Export1/ map?businessObject=Customer &contentType=EDI

   In general, it is impractical to create the custom headers; setting query parameters on the URL is the recommended method.
2. The response message cannot be configured at the instance level. You must use the custom properties of the binding to make the association with the specific WebSphere Transformation Extender maps that you want to use.

Chapter 10. Adapters supported by the server

WebSphere Process Server supports two types of adapters: WebSphere Adapters and WebSphere Business Integration Adapters. Adapters enable business applications to act as services by connecting them to diverse enterprise information systems (EISs), such as databases, enterprise resource planning systems, file systems, and e-mail systems.

With the help of an adapter, the application and EIS can ″talk to each other,″ or, in other words, send and retrieve information in a consistent way. To allow your applications to operate as services, the adapter connects them to WebSphere Process Server, which powers your Service Oriented Architecture (SOA). With an adapter, you no longer need to provide proprietary connection utilities (or write custom connection utilities) for each EIS or application server.

Differences between WebSphere Adapters and WebSphere Business Integration Adapters

Both WebSphere Adapters and WebSphere Business Integration Adapters mediate communication between components and enterprise information systems. The two types of adapter differ in several respects including: their integration, their JCA-compliance, their data models, and the management of their connectivity.

There are several differences between WebSphere Adapters and WebSphere Business Integration Adapters. These distinctions are most important during development of applications. When deploying applications to a running server, the nature of the adapters used affects some of the steps that need to be followed.

Adapters provide communication mechanisms between enterprise information systems (EISs) and WebSphere applications. To illustrate the operation of the adapters, Figure 33 on page 280 and Figure 34 on page 281 provide details of the communication between the server and the EIS for the two types of adapters.

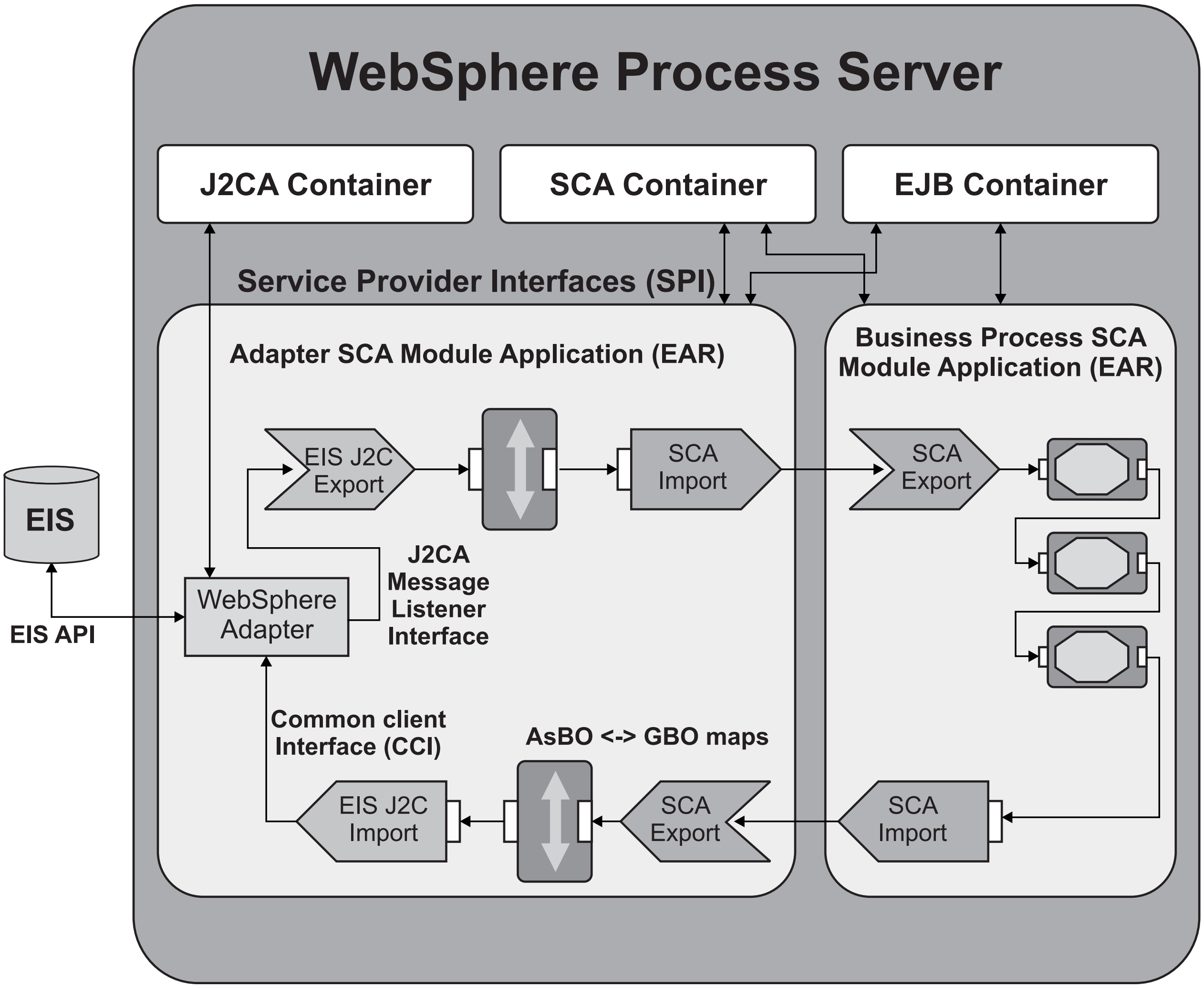


Figure 33. Detailed schematic of a WebSphere Adapter

Figure 33 depicts a WebSphere Adapter managing the connectivity between a J2EE component supported by the server and the EIS. The WebSphere Adapter resides inside the server.

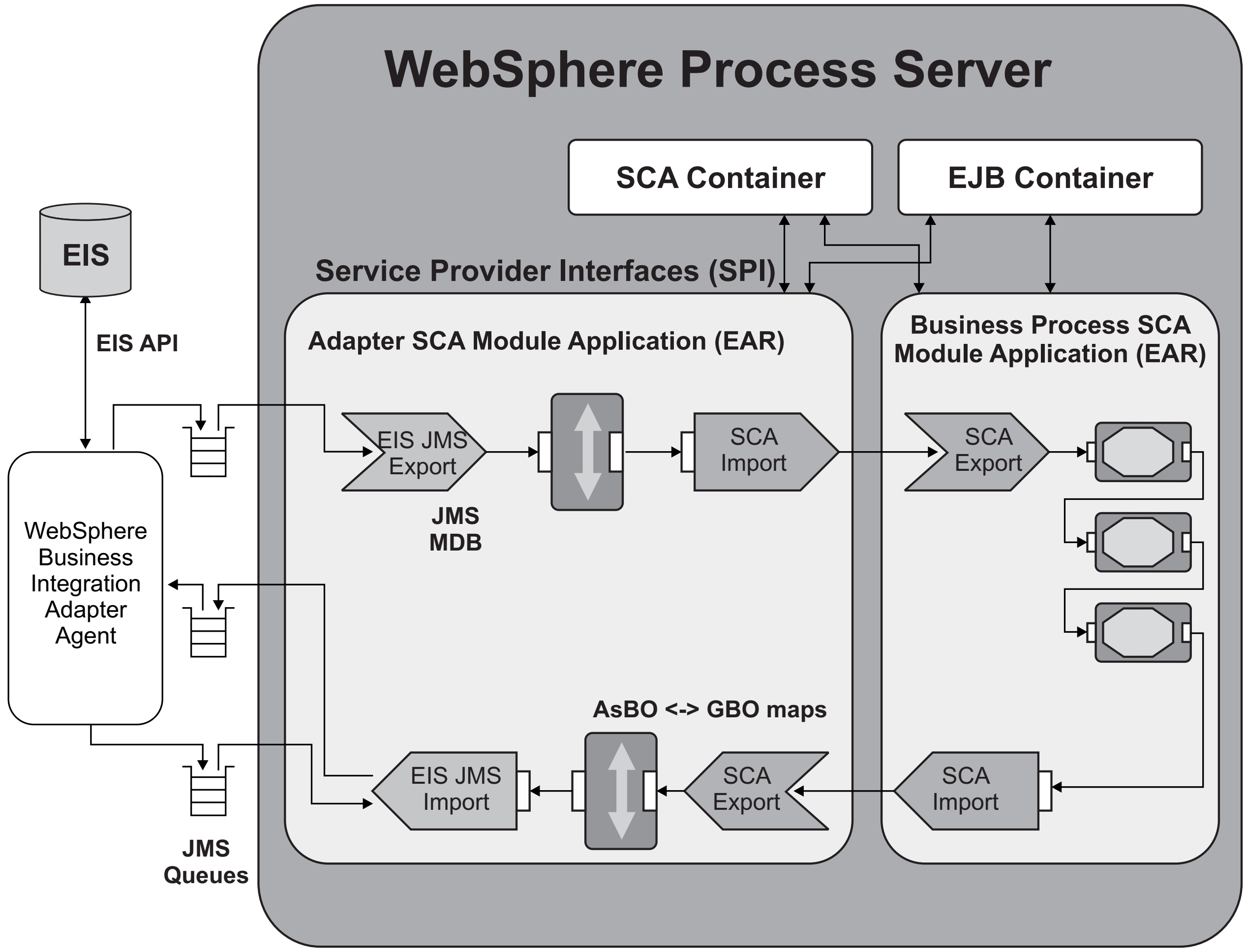


Figure 34. Detailed schematic of a WebSphere Business Integration Adapter.

Figure 34 shows a WebSphere Business Integration Adapter mediating communication between the WebSphere Integration Broker and the EIS. The integration broker communicates with the WebSphere Business Integration Adapter through the use of a Java Message Service (JMS) transport layer.

Table 38 shows the differences between the two types of adapter.

Table 38. Differences between WebSphere Adapters and WebSphere Business Integration Adapters

Feature	WebSphere Adapters	WebSphere Business Integration Adapters
JCA Compliance	Fully JCA compliant (version 1.5).	Not JCA-compliant.
Connectivity Manager	Rely on standard JCA contracts to manage life cycle tasks such as starting and stopping.	Rely on WebSphere Adapter Framework to manage connectivity.
Event Notification	Use an EventStore subclass to retrieve events from an EIS.	Manage event notification using a pollFor Events method.

Table 38. Differences between WebSphere Adapters and WebSphere Business Integration Adapters (continued)

Feature	WebSphere Adapters	WebSphere Business Integration Adapters
Request Processing	Clients directly invoke one of several interaction contracts to query or modify data in the EIS.	Rely on an integration server and the WebSphere Adapter Framework to initiate and help process requests.
Data Models	Use an Enterprise Metadata Discovery (EMD) utility to parse an EIS and develop Service Data Objects (SDOs) and other useful artifacts. The EMD is part of the WebSphere Adapter implementation.	Use a separate Object Discovery Agent (ODA) to introspect an EIS and generate business object definition schemas.
Integration	Run on the server.	Reside outside the server. The server or integration broker communicates with the adapter via a Java Message Service (JMS) transport layer.

WebSphere Adapters are the recommended product.

WebSphere adapters

WebSphere adapters, also known as resource adapters, enable managed, bidirectional connectivity between enterprise information systems (EISs) and J2EE components supported by the server.

WebSphere adapters, which are preferred over WebSphere Business Integration adapters, are covered elsewhere in this information library.

Where to find more information

To learn about configuring and using WebSphere adapters, see Configuring and using adapters in the WebSphere Integration Developer information center. In the adapter guide for your adapter, expand the navigation and click **Administering the adapter module**.

For general information about adapters, see Accessing external services with adapters in the WebSphere Integration Developer information center.

For information about EIS bindings, which are used with WebSphere adapters to provide connectivity between SCA components and an EIS, see “EIS bindings” on page 200.

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 administrative console.

Where to find more information

For more information about working with these adapters, see Using WebSphere Business Integration Adapters.

Managing the WebSphere Business Integration Adapter

You can manage a running WebSphere Business Integration Adapter from the administrative console.

Before you begin

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

About this task

Use the following procedures to manage your resources and to perform various administrative actions on them.

Procedure

1. Select the resource or resources to manage. From the top level of the administrative console follow these steps:
   a. Expand **Servers**.
   b. Select **Application Servers**.
   c. From the list of servers, select the server where the resources you intend to manage reside.

      Click on the name of the server that hosts the resources of interest.
   d. Select **WebSphere Business Integration Adapter Service**.

      Under the Business Integration subheading on the Configuration tab select **WebSphere Business Integration Adapter Service**.
   e. Select **Manage the WebSphere Business Integration Adapter resources**.
   f. From the list of resources, choose those that you want to manage.

      Select the check boxes associated with the resources you intend to manage.
2. Manage the selected resources.

   Click one of the command buttons to act upon the selected resources.

Command	Description
Deactivate	Changes the status of the selected resources from active to paused or inactive.
Activate	Changes the status of the selected resources from inactive to active.
Suspend	Changes the status of the selected resources from active to paused.
Resume	Changes the status of the selected resources from paused to active.
Shut down	Changes the status of the selected resources from active to unavailable.

Chapter 11. Working with events

Events are requests or responses sent from one component to another. You can process events in a specific sequence; in addition, you can view, discard, modify, or resubmit any events that fail.

Processing events in sequence

Event sequencing guarantees that WebSphere Process Server components process events from asynchronous invocations in the order in which they are delivered. Event order is maintained throughout the entire business integration scenario.

An *event* is a request or a response that is sent from one component to another. The event encapsulates data and invocation metadata (for example, the name of the target component, the operation, and the parameters).

Note: Event sequencing is supported only for requests sent with an asynchronous invocation.

Why use event sequencing?

Some implementations require the target component to process events in the same order in which they were sent by the source application; processing them out of order can cause errors or exceptions. For example, if a source application generates an event to create a business object and then generates an event to update that business object, the create event must be processed first.

In an asynchronous invocation, events are stored in destinations on a service integration bus and can be handled by multiple instances of Message Driven Beans (MDBs). As a result, they may be processed non-sequentially, which can cause failures. To avoid this problem, use event sequencing.

How does event sequencing work?

Enable event sequencing by using the *event sequencing qualifiers* available in WebSphere Integration Developer. The qualifiers must be set on each method that requires event sequencing; they tell the runtime environment that invocations to these methods need to be sequenced.

Each qualifier has an event sequencing key that determines how events are sequenced. The key's value is comprised of one or more attributes of the business objects associated with an invocation. All events that share the same key are grouped together and processed in sequence. Events that do not have an event sequencing key continue to be processed as normal, in parallel with the sequenced events.

A sequenced event acquires a lock before being sent to the target component for processing. As soon as the business logic for the event has executed, the lock is released and given to the next event with the same event sequencing key. If the event cannot acquire the necessary lock, the execution of the invocation is suspended until the lock is acquired.

Related concepts

Example: Event sequencing
To understand how event sequencing works, consider a situation in which a source application (Component A) asynchronously invokes a target application (Component B) to create new orders, and then updates those orders with revised data.

Considerations for implementing event sequencing
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">
            <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">
            <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
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
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.

Event sequencing declarations for components

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.

Important: When declaring event sequencing on a component, ensure that the component is invoked in a managed thread. The managed thread provides the session information required to properly sequence events.

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.

Use the event sequencing qualifier’s attributes to further extend sequencing functionality. For example, the sequencingGroup attribute groups methods that need to be sequenced together; all events that are generated by any method in the same group are processed sequentially.

WebSphere Integration Developer provides a setting to determine the event processing behavior of sequenced events when encountering unexpected runtime failures when authoring the Event Sequencing qualifier for the specific service operation. The configuration setting for the qualifier is controlled by the checkbox called ″Process requests when error encountered″. For further information about the configurable behavior of failed event processing of sequenced events, see the topic devoted to Failed Sequenced Events in this documentation.

When declaring event sequencing on a component, ensure that the component is invoked in a managed thread. The managed thread provides the session information required to properly sequence events.

Event sequencing with export bindings

Event sequencing is supported with EIS, JMS, WebSphere MQ, and WebSphere MQ JMS export bindings. To ensure that the exports process and deliver messages in the correct sequence, you must configure the export bindings appropriately.

Consider the following requirements when using event sequencing on a target component that handles export bindings:

- An adapter component must use the non-optimized path for a Java Message Service (JMS) export when event sequencing is used on the target component.
- To enable event sequencing for JMS export bindings, you must limit the number of concurrent Message Driven Beans (MDBs) that are processing incoming messages. Do this by setting the maxConcurrency custom property on the ActivationSpec to a value of 1.
- To enable event sequencing for a WebSphere MQ JMS export , you must limit the number of concurrent listener threads that will deliver messages to the Message Driven Bean. Do this by setting the maxSessions property to a value of 1.
- To enable event sequencing for a native MQ export, you must use WebSphere Integration Developer to set the eventSequencing property.

Related information

Enabling event sequencing for an EIS Export binding

Enabling event sequencing for a JMS Export binding

Enabling event sequencing for a WebSphere MQ JMS Export bindings

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 39. 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.

Failed sequenced events

Processing errors or unavailable resources can cause a sequenced event to fail. How any remaining events in the sequence are handled is determined by the setting of the ″Process requests when error encountered″ attribute of the event sequencing qualifier in WebSphere Integration Developer.

The ″Process requests when error encountered″ attribute of the event sequencing qualifier has two possible values.

Checked

Use this value if you want the processing of the sequence of events to ignore the failure and proceed to process the next event in the sequence.

Unchecked

Use this value if you want to halt the processing of dependent events in the sequence until the failure is resolved. You can use the failed event manager to quickly identify failed sequenced events and resubmit them for processing.

When this attribute is unchecked and a sequenced event is unsuccessfully processed and sent to the failed event manager, you can handle it in one of the following ways:

- resubmit it without modification
- resubmit it with modification (either with or without changing event sequencing key identifiers)
- delete it (the Recovery subsystem uses the event sequencing callback to delete the lock associated with the deleted event to allow remaining events in the sequence to be processed)

If the resubmission is successful, the event is processed in its original sequenced position within the queue.

Limitations in event sequencing

Certain types of components and invocations offer limited support for event sequencing.

Limitations for the current release of event sequencing include the following:

- Event sequencing on operations bound to a Business Process Execution Language (BPEL) process with a non-initiating receive is not recommended. In long-lived business processes, event sequencing relies on a work completion contract to determine when to release a lock; this work completion contract is activated whenever a new process instance is created. However, no new process instance is created when there is a non-initiating receive. As a result, it is difficult for the event sequencing runtime to accurately detect a completed work contract and it can release the lock either too early or too late.
- Event sequencing on operations bound to a Business State Machine with a non-initiating receive is not recommended.
- Work completion contracts are supported only for BPEL components. To effectively use event sequencing on any other type of component that has asynchronous invocations, it is recommended that you use the request-response method signature. The event sequencing runtime interprets a response as a signal that the work is complete and releases the lock.

  Note: If you cannot declare a method as a request-response operation, you might need to specify event sequencing on downstream components, making sure you use the same event sequencing key for all methods.

Event sequencing is not supported in the following scenarios:

- Using unmanaged threads or non-SCA bindings to send events to their destinations without proper session context.
- Using synchronous invocations to components with asynchronous invocations.

Enabling event sequencing in WebSphere Process Server

Event sequencing provides the ability to sequence incoming events to an SCA component in WebSphere Process Server. Export bindings are the entry points to the target SCA components. For event sequencing to be enabled, the exports must process and deliver messages in the same order in which those messages are received.

Enabling event sequencing: EIS exports

WebSphere adapters provide a mechanism to allow event sequencing in WebSphere Process Server, by specifying an activation specification property. Also, the export must process and deliver messages in the same order it received those messages.

About this task

For more details about the activation specification property; see the WebSphere adapter documentation. For JCA 1.5 resource adapters, consult the specific provider documentation for details on how to configure the adapter to enable the ordering or sequencing of events.

In general, if event sequencing is required in a network deployment environment, the module that has the export should be moved to a standalone server or to a cluster that has only one active server that is enabled for high availability.

Enabling event sequencing: JMS exports

To enable event sequencing for JMS export bindings in WebSphere Process Server, you must configure properties of the binding. Also, the export must process and deliver messages in the same order that it receives those messages.

Before you begin

Event sequencing for JMS export bindings is supported in a clustered environment only when the destinations are not partitioned. For event sequencing to function in a network deployment environment with clusters, there can be only one active messaging engine per cluster. For event sequencing in a network deployment environment with servers that are not in a cluster, each server can have an active messaging engine.

About this task

Event sequencing requires events to acquire a lock before being dispatched to the target component for processing. When processing is complete, the event releases the lock. If an event cannot acquire a lock, processing of the invocation is suspended. If the event subsequently acquires a lock, it will be dispatched.

You declare that event sequencing is required on a particular method for a particular component by adding an event sequencing qualifier to the method in the component definition.

- The keySpecification attribute defines the key that will be used to identify the events that need to be sequenced.
- The parameter attribute specifies the parameter from which the key attributes will be extracted.
- The name attribute is the name of the parameter.
- The xpath attribute is applied to the parameter to extract a value that will be part of the key.

You must specify a parameter element for each parameter that is going to contribute to the key.

The **esadmin** command line utility can be used to list locks and delete locks, both active and queued.

You enable event sequencing for a JMS export from WebSphere Integration Developer.

Procedure

1. In WebSphere Integration Developer, click the **Properties** tab for the export.
2. From the **Message configuration** section, select the **Event sequence required** check box.

Results

Event sequencing is enabled for your binding.

Note: Removing the exception destination means that any failure will stop all incoming messages.

Enabling event sequencing: Generic JMS exports

To enable event sequencing for Generic JMS export bindings in WebSphere Process Server, you must configure properties of the binding. Also, the export must process and deliver messages in the same order that it receives those messages.

Before you begin

If event sequencing for Generic JMS export bindings is required in a network deployment environment, the module that has the export should be moved to a standalone server or to a cluster that has only one active server that is enabled for high availability.

About this task

You enable event sequencing for a Generic JMS export from WebSphere Integration Developer.

Procedure

1. In WebSphere Integration Developer, click the **Properties** tab for the export.
2. From the **Message configuration** section, select the **Event sequence required** check box.

Enabling event sequencing: WebSphere MQ JMS exports

To enable event sequencing for WebSphere MQ JMS export bindings in WebSphere Process Server, you must configure properties of the binding. Also, the export must process and deliver messages in the same order that it receives those messages.

Before you begin

If event sequencing for these export bindings is required in a network deployment environment, the module that has the export should be moved to a standalone server or to a cluster that has only one active server that is enabled for high availability.

About this task

You enable event sequencing for WebSphere MQ JMS export bindings from WebSphere Integration Developer.

Procedure

1. In WebSphere Integration Developer, click the **Properties** tab for the export.
2. From the **Message configuration** section, select the **Event sequence required** check box.

Enabling event sequencing: WebSphere MQ exports

To enable event sequencing for WebSphere MQ export bindings in WebSphere Process Server, you must configure properties of the binding. Also, the export must process and deliver messages in the same order that it receives those messages.

Before you begin

If event sequencing for WebSphere MQ export bindings is required in a network deployment environment, the module that has the export should be moved to a standalone server or to a cluster that has only one active server that is enabled for high availability.

About this task

You enable event sequencing for a WebSphere MQ export from WebSphere Integration Developer.

Procedure

1. In WebSphere Integration Developer, click the **Properties** tab for the export.
2. From the **Message configuration** section, select the **Event sequence required** check box.

Listing, releasing, and deleting locks

The lock manager handles event sequencing locks. You can use the esAdmin command to list, delete, or unlock any lock in the lock manager.

The lock manager supports two operations on event locks:

- Lock: The lock operation attempts to acquire a lock and stores the lock request in a database. After a lock is granted, processing resumes for the invocation that requested the lock.
- Unlock: The unlock operation releases the current lock and grants the lock to the next lock request.

Requests for the same lock are put into a queue in the order in which they are received. Locks are persisted to the default WebSphere Process Server database and data source to ensure they can be recovered in the case of a server failure.

The esAdmin command enables you to administer the active and queued locks currently in the lock manager. The following sections provide more detail on using esAdmin.

Note: If you are using partitioned databases, run the esAdmin command once for each deployment target. In a clustered environment, you can run it on any cluster, but do not run it on the deployment manager.

Listing locks

The esAdmin command can list all active and queued locks in the lock manager, or only those locks associated with a specific module, component, or method.

Use one of the following methods with esAdmin:

- listAll: Lists all active and queued locks in the lock manager.
- listLocks: Lists a subset of the active and queued locks in the lock manager. Specify one or more of the following parameters to return a filtered list of locks:
  – moduleName
  – componentName
  – methodName

  For example, the following command returns a list of active and queued locks for the CustComp component that is part of the CusMod module.

  ```
  esAdmin listLocks CustMod CustComp
  ```

The command returns output that looks like the following:

Table 40. 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.
```

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.

For examples:

```
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.

Event sequencing callback fails to release a lock

While trying to delete a failed sequenced event in the Recovery subsystem, the event sequencing callback can fail to release the event's lock. This typically occurs when a target application has been removed or when other components of the system (for example, the database) are unavailable.

In this situation, the failed event manager generates an error message. Use the esAdmin command to manually delete the lock associated with the failed event.

Performance issues

If you are experiencing memory problems on the messaging engine server used for event sequencing components, try modifying the runtime event sequencing properties in the *install_root*/properties/eventsequencing.properties file.

The maxActiveMessages property defines the number of messages currently locked on a component destination; too many large messages can negatively affect performance and cause memory problems. Note that a value of 0 (zero) means that an unlimited number of messages are allowed. By default, the maxActiveMessages property is set to 100. When changing the value, consider using the following formula where *delta* is the standard deviation of the accuracy of the estimate for the anticipated number of sequenced events with the same sequencing key that can be simultaneously processed.

```
average_number_of_ES_keys * average_number_of_potential_queued_events_per_key
+ delta
```

The workItemRetryCount property sets the upper boundary for the verification work retry count. A verification work item is spawned when an asynchronous event is unlocked and there are dependent events waiting to be processed. In this situation the creation and deletion of the lock are done in separate units of work and the work verification task ensures that the processing of one unit of work is complete before the next event is processed. By default, workItemRetryCount is set to -1 (retry).

The workItemSleepTime property specifies the amount of time that elapses between work verification retry attempts. By default, workItemSleepTime is set to 10 seconds. Note that lowering the value can decrease performance.

To modify any of the properties, perform the following steps.

1. Open the eventsequencing.properties file in a text editor.
2. Make the appropriate modifications for your environment.
3. Save and close the file.
4. Stop and restart any applications that are part of the event sequencing component in order for the changes to take effect.

Managing failed events

The WebSphere Process Server Recovery service captures data about failed events. You can then use the failed event manager to view, modify, resubmit, or delete the failed event.

The WebSphere Process Server Recovery service manages failed operations between Service Component Architecture (SCA) components, failed JMS events, and failed operations within long-running business processes.

Failed SCA events

In the context of SCA, an event is a request or response that is received by a service 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 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 Recovery service handles the following types of failed SCA events:

- Event failures that occur during an asynchronous invocation of an 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 SCA 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 (callback) 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.

The Recovery service sends failed SCA 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.

Failed JMS events

The Java Message Service (JMS) binding type and configuration determine whether a failed event is generated and sent to the failed event manager.

JMS bindings

WebSphere Integration Developer provides a recovery binding property that allows you to enable or disabled recovery for each JMS binding, at authoring time. The `recoveryMode` property can be set to one of the following:

`bindingManaged`	Allow binding to manage recovery for failed messages
`unmanaged`	Rely on transport-specific recovery for failed messages

Recovery for JMS bindings is enabled by default. When it is enabled, JMS failed events are created in the following situations:

- The function selector fails
- The fault selector fails
- The fault selector returns the `RuntimeException` fault type
- The fault handler fails
- The data binding or data handler fails after a single retry in JMS

In addition, a Service Component Architecture (SCA) failed event is created when the `ServiceRuntimeException` exception is thrown in a JMS binding target component after a single retry in JMS.

These failures can occur during inbound or outbound communication. During outbound communication, JMSImport sends a request message and receives the response message; a failed event is generated if the JMS import binding detects a problem while processing the service response. During inbound communication, the sequence of events is as follows:

1. JMSExport receives the request message.
2. JMSExport invokes the SCA component.
3. The SCA component returns a response to JMSExport.
4. JMSExport sends a response message.

A failed event is generated if the JMS export binding detects a problem while processing the service request.

The Recovery service captures the JMS message and stores it in a Recovery table in the Common database. It also captures and stores the module name, component name, operation name, failure time, exception detail, and JMS properties of the failed event. You can use the failed event manager to manage failed JMS events, or you can use a custom program.

To disable recovery, you must explicitly disable it in WebSphere Integration Developer by setting the `recoveryMode` property to `unmanaged`.

Note: If the `recoveryMode` property is missing (for earlier versions of applications), the recovery capability is regarded as enabled.
When recovery is disabled, a failed message is rolled back to its original destination and retried. The system does not create a failed event.

WebSphere MQ JMS bindings and generic JMS bindings

WebSphere MQ JMS bindings and generic JMS bindings handle failures in a different way than JMS bindings. Problems during request and response handling do not generate a failed JMS event. Instead, they generate a failed SCA event if the following two conditions are met:

- The underlying messaging system is configured to automatically redeliver a failed message.
- The failure occurs in the export binding's target SCA component, not in the binding itself.

When both conditions are true, the Recovery system generates a failed SCA event that you can manage with the failed event manager.

In all other situations, the failed message rolls back to its original destination, where it is handled according to the messaging system configuration. No failed event is created.

Failed Business Process Choreographer events

In the context of Business Process Choreographer, exceptions can occur that, if not handled by the process logic, cause an activity to stop or the process instance to fail. A failed event is generated when a long-running Business Process Execution Language (BPEL) process fails and one of the following happens:

- The process instance enters the failed or terminated state
- An activity enters the stopped state

The Recovery service captures the module name and component name for failed Business Process Choreographer events. Failed event data is stored in the Business Process Choreographer database (BPEDB) database.

Note that the Recovery service does not handle failures from business process and human task asynchronous request/reply invocations.

Business Flow Manager hold queue messages

You can use the failed event manager to manage navigation messages that are stored in the Business Flow Manager hold queue. A navigation message might be stored in the hold queue if:

- An infrastructure, such as a database, is unavailable.
- The message is damaged.

In a long-running process, the Business Flow Manager can send itself request messages that trigger follow-on navigation. These messages trigger either a process-related action (for example, invoking a fault handler) or an activity-related action (for example, continuing process navigation at the activity). A navigation message always contains its associated process instance ID (piid). If the message triggers an activity-related action, it also contains the activity template ID (atid) and the activity instance ID (aiid).

You can use the failed event manager to manage Business Flow Manager hold queue messages, or you can use a custom program.

Business Flow Manager hold queue messages cannot be deleted directly in the failed event manager. If the related process instance does not exist, replaying the hold queue message will result in deletion of the message.

How are failed events managed?

An administrator uses the failed event manager to browse and manage 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

Session monitoring
You can monitor multiple events that are part of the same session, by using the Common Base Event browser to find all events on the Common Event Infrastructure database that contain the identical session ID attribute.

Recovery from infrastructure failures
A long-running process spans multiple transactions. If a transaction fails because of an infrastructure failure, Business Flow Manager provides a facility for automatically recovering from these failures.

Security considerations for recovery

If you have enabled security for your WebSphere Process Server applications and environment, it is important to understand how role-based access and user identity affect the Recovery subsystem.

Role-based access for the failed event manager
The failed event manager uses role-based access control for the failed event data and tasks. Only the administrator and operator roles are authorized to perform tasks within the failed event manager. Users logged in as either administrator or operator can view all data associated with failed events and can perform all tasks.

Event identity and user permissions
A failed event encapsulates information about the user who originated the request. If a failed event is resubmitted, its identity information is updated to reflect the user who resubmitted the event. Because different users logged in as administrator or operator can resubmit events, these users must be given permissions to the downstream components required to process the event.

For more information about implementing security, see Securing applications and their environment.

Finding failed events

Failed events are stored in a database and are retrieved through the search functionality of the failed event manager. You can search for all failed events on all the servers within the cell, or for a specific subset of events.

Before you begin

If administrative security is enabled, you must be logged in as administrator or operator to perform this task.

About this task

This topic describes how to find all failed events in the cell. This default query returns all SCA and JMS failed events.

If Business Process Choreographer is installed, the query also returns failed, terminated, and stopped Business Process Choreographer events.

To retrieve a complete list of failed events, use the following procedure.

Procedure

1. Ensure the administrative console is running.
2. Click **Integration Applications** → **Failed Event Manager** to enter the failed event manager.
3. From the **Failed events on this server** box, click **Get all failed events**.

Results

The Search Results page opens, displaying a list of all the WebSphere Process Server failed events in the cell.

What to do next

You can now view (and in some cases, modify) data in a failed event, resubmit it, or delete it.

Searching for events by criteria

Use the failed event manager Search page to find only those events that meet specified criteria. You can search by failed event type and by criteria such as failure time, event destination or source, exception or business object type, session ID or, for WebSphere Process Server only, event sequencing qualifier.

Before you begin

If administrative security is enabled, you must be logged in as administrator or operator to perform this task.

About this task

To search for a specific subset of failed events on the server, perform the following steps.

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 **Search failed events**.
4. From the **Event type** box on the Search failed events page, select one or more types of events to search for:
   - SCA
   - JMS
   - Business Process Choreographer
   - Business Flow Manager hold queue messages
5. If you are searching for Business Process Choreographer events, verify the event status selected in the Event status box. By default, the failed event manager returns all failed, stopped, and terminated Business Process Choreographer events, but you can modify the search to return only events with a particular status.
6. Optional: Specify any additional search criteria. The following table describes the available options. If you specify multiple criteria, the AND operator is used during the query; the failed event manager returns only events that meet all of the criteria.

Table 41. Search criteria

Search criteria	Field or fields to use	Supported event types	Usage notes
The module, component, or method the event was en route to when it failed.	**Module** **Component** **Operation**	SCA JMS Business Process Choreographer Business Flow Manager hold queue	Use one or more of these fields to search for failed events associated with a specific module, component, or method.
The time period during which the event failed	**From date** **To date**	SCA JMS Business Process Choreographer Business Flow Manager hold queue	Formats for date and time are locale-specific. An example is provided with each field. If the value you provide is not formatted correctly, 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 the individual workstations running the administrative console.
The session in which the event failed	**Session ID**	SCA	None
The module or component from which the event originated	**Source module** **Source component**	SCA	Use one or both of these fields to find only those failed events that originated from a specific source module or component. The failed event manager determines the source based on the point of failure, regardless of interaction type.
The type of business object in the failed event	**Business object type**	SCA	None
Whether the event had the event sequencing qualifier specified	**Event sequencing qualified**	SCA	This search criteria is applicable only for WebSphere Process Server.

Table 41. Search criteria (continued)

Search criteria	Field or fields to use	Supported event types	Usage notes
The exception thrown when the event failed	**Exception text**	SCA	Specify all or part of the exception text in the field to find all events associated with that exception.

For detailed information about each field and the values it accepts, see the online help for the failed event manager Search page.

7. Click **OK** to begin the search.

What to do next

You can now view (and in some cases, modify) data in a failed event, resubmit it, or delete it.

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 type of failed event has the following data associated with it:

- SCA events
  - The event ID, type, status, and session ID
  - The service invocation type between SCA components
  - The names of the module and component from which the event originated (the source).
  - The names of the destination module, component and method for the event
  - The time the event failed
  - The deployment target associated with the event
  - Whether an event sequencing qualifier has been declared for this event
  - The destination module where the event has been or will be resubmitted
  - The correlation ID, if one exists
  - The exception thrown when the event failed
  - The expiration date for resubmitted events (this data can be edited)
  - The trace control set for the event (this data can be edited)
- JMS events:
  - The event ID, type, and status
  - The interaction type
  - The names of the destination module, component and method for the event
  - The time the event failed
  - The deployment target associated with the event
  - The exception thrown when the event failed
  - The destination module where the event has been or will be resubmitted
  - The correlation ID, if one exists

  - The expiration date for resubmitted events (this data can be edited)
  - The JMS-specific properties associated with the failed event: redelivered count, delivery mode, message priority, type, `replyTo` destination, and redelivered indicator (`true` or `false`).
- Business Process Choreographer events:
  - The event ID, type, and status
  - The names of the destination module and component for the event
  - The time the event failed
  - The deployment target associated with the event
  - The process instance name associated with the event
  - The top-level process ID associated with the event
- Business Flow Manager hold queue events:
  - The event ID, type, and status
  - The time the event failed
  - The deployment target associated with the event
  - The process instance ID (if the process instance does not exist, `0` is returned)
  - The name and state of the process instance
  - The name of the associated process template
  - The activity instance name and ID
  - The activity template ID

Business data

SCA and Business Process Choreographer failed 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 for SCA failed events can be edited with the business data editor available in the failed event manager.

Browsing data in failed events

Use the failed event manager to view failed event data and any business data associated with the event.

Before you begin

If administrative security is enabled, you must be logged as administrator or operator to perform this task.

About this task

To browse failed event data, use the following procedure.

Procedure

1. Ensure that the failed event manager is open and that you have retrieved a list of the failed events on your system.
2. From the Search Results page of the failed event manager, click the ID (found in the Event 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 your failed event has business data, you can browse it by clicking **Edit business data**.

The Business Data Editor collection page opens, displaying the business data associated with the failed event. Each parameter name in the hierarchy is a link. If the parameter is a simple data type, clicking its name will open up a form so you can edit the parameter's value. If the parameter is a complex data type, clicking its name will expand the hierarchy further.

Editing trace or expiration data in a failed SCA event

The Failed Event Details page enables you to set or modify values for the trace control and expiration date associated with a failed event.

Before you begin

You must be logged in as administrator or operator to perform this task.

About this task

Important: Any edits you make to the trace or expiration data are only saved locally until you resubmit the event. If you perform any other action before resubmitting the event, all edits are lost.

Failed SCA 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 SCA events also have an expiration. If a user has specified an expiration with the asynchronous call that sends the event, that data persists even if the event fails, and the expiration time appears in the **Resubmit Expiration Time** field on the Failed Event Details page. Expired failed events cannot be resubmitted successfully. To prevent a second failure, you can edit the expiration date for the event to ensure that it is not expired when it is resubmitted.

To edit trace or expiration data in a failed event, use the following procedure.

Procedure

1. Ensure that the failed event manager is open and that you have retrieved a list of the failed events on your system.
2. From the failed event manager's Search Results page, click the ID (found in the Event 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 Business Process Management 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 301
Failed events are stored in a database and are retrieved through the search functionality of the failed event manager. You can search for all failed events on all the servers within the cell, or for a specific subset of events.

Editing business data in a failed SCA event

Business data can be encapsulated into a business object, or it can be simple data that is not part of a business object. A failed event can have both simple data and a business object associated with it. Use the business data editor to edit the business data associated with a failed event before you resubmit it.

Before you begin

If administrative security is enabled, you must be logged in as administrator or operator to perform this task.

About this task

For each failed event, the editor displays the associated business data in a hierarchical format; the navigation tree at the top of the table is updated as you navigate through the parameters to give you a clear picture of where you are in the hierarchy.

You can edit only simple data types (for example, String, Long, Integer, Date, Boolean). If a data type is complex (for example, an array or a business object), you must navigate through the business data hierarchy until you reach the simple data types that make up the array or business object. Complex data is denoted by an ellipsis (`...`) in the Parameter Value column.

Note that you cannot use the failed event manager to edit business data for a Business Process Choreographer event. Instead, click the **Open calling process in Business Process Explorer** link from the failed event detail page and use Business Process Choreographer Explorer to make any permitted modifications.

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 SCA event, use the following procedure.

Procedure

1. Ensure that the failed event manager is open and that you have retrieved a list of the failed events on your system.
2. From the failed event manager's Search Results page, click the ID (found in the Event ID column) of the failed event whose data you want to edit.
3. From the failed event detail 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 send an event again, you must resubmit it from the failed event manager. You can resubmit an event without changes, or, in some cases, 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 SCA 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 event ID to track its success or failure. If a resubmitted event fails again, it is returned to the failed event manager with its original event ID and an updated failure time.

Resubmitting an unchanged failed event

You can resubmit one or more unchanged failed events to be processed again. Processing resumes only for the failed branch, not for the entire event.

About this task

If administrative security is enabled, you must be logged in as administrator or operator to perform this task.

Procedure

1. Ensure that the failed event manager is open and that you have retrieved a list of the failed events on your system.
2. From the Search Results page, select the check box next to each failed event you want to resubmit.
3. Click **Resubmit**.

Results

Each selected event is resubmitted for processing and is removed from the failed event manager.

Resubmitting a failed SCA event with trace

You can monitor the resubmission of a failed SCA event to determine whether it now succeeds. The failed event manager provides optional tracing for all failed events.

About this task

Tracing can be set for a service or a component, and it can be output to a log or to the Common Event Infrastructure (CEI) server. For detailed information about setting and viewing trace, see the Monitoring topics in the information center.

If administrative security is enabled, you must be logged in as administrator or operator to perform this task.

Procedure

1. Ensure that the failed event manager is open and that you have retrieved a list of the failed events on your system.
2. From the Search Results page, select the check box next to each failed event you want to resubmit.
3. Click **Resubmit with trace**.
4. From the Resubmit with Trace page, specify the level of trace you want to use in the **Trace control** field.

   By default, the value is SCA.LOG.INFO;COMP.LOG.INFO. With this setting, no trace occurs when the session calls an SCA service or executes a component.
5. Click **OK** to resubmit the failed event and return to the Search Results page.

What to do next

To view the trace log for a resubmitted event, open the corresponding component logger or use the CEI log viewer.

Managing failed JMS events

When problems processing a JMS request or response message create a failed JMS event in the Recovery subsystem, you must decide how to manage that event. Use the information in this topic to help you identify and fix the error and clear the event from the Recovery subsystem.

About this task

To manage a failed JMS event, perform the following steps.

Procedure

1. Use the failed event manager to locate information about the failed JMS event, taking note of the exception type.
2. Locate the exception type in Table 42 on page 310 to determine the location and possible causes of the error, as well as suggested actions for managing the failed event.

Table 42. Failed JMS events

Exception type	Location of error	Possible cause of error	Suggested action
FaultServiceException	Fault handler or fault selector	There is malformed data in the JMS message.	1. Inspect the JMS message and locate the malformed data. 2. Repair the client that originated the message so it creates correctly formed data. 3. Resend the message. 4. Delete the failed event.
		There was an unexpected error in the fault handler or fault selector.	1. Debug the custom fault selector or fault handler, fixing any errors identified. 2. Resubmit the failed event.
ServiceRuntimeException	Fault handler	The fault selector and runtime exception handler are configured to interpret the JMS message as a runtime exception. This is an expected exception.	Look at the exception text to determine the exact cause, and then take appropriate action.
DataBindingException or DataHandlerException	Data binding or data handler	There is malformed data in the JMS message.	1. Inspect the JMS message and locate the malformed data. 2. Repair the client that originated the message so it creates correctly formed data. 3. Resend the message. 4. Delete the failed event.
		There was an unexpected error in the data binding or data handler.	1. Debug the custom data binding or data handler, fixing any errors identified. 2. Resend the message. 3. Delete the failed event.

Table 42. Failed JMS events (continued)

Exception type	Location of error	Possible cause of error	Suggested action
SelectorException	Function selector	There is malformed data in the JMS message.	1. Inspect the JMS message and locate the malformed data. 2. Repair the client that originated the message so it creates correctly formed data. 3. Resend the message. 4. Delete the failed event.
		There was an unexpected error in the function selector.	1. Debug the custom function selector, fixing any errors identified. 2. Resend the message. 3. Delete the failed event.

Managing stopped Business Process Choreographer events

Use the failed event manager and Business Process Choreographer Explorer to manage stopped Business Process Choreographer events in any process state. Stopped events occur if a Business Process Execution Language (BPEL) instance encounters an exception and one or more activities enter the Stopped state.

About this task

You can view, compensate, or terminate the process instance associated with a stopped Business Process Choreographer event. In addition, you can work with the activities associated with the event. viewing, modifying, retrying, or completing them as appropriate.

To manage stopped events originating from a long-running BPEL process, perform the following steps.

Procedure

1. Ensure the administrative console is running.
2. Open the failed event manager by clicking **Integration Applications** → **Failed Event Manager**.
3. Perform a search to find the stopped Business Process Choreographer event or events you want to manage.
4. For each stopped event you want to manage, do the following:
   a. Click the stopped event ID in the Event ID column of the Search Results page.
   b. From the event detail page, click **Open calling process in Business Process Choreographer Explorer**.

c. Use Business Process Choreographer Explorer to manage the event and its associated activities.

Finding business process instances related to a failed event

If a failed event is generated from a business process, the failed event manager provides a link to view that business process instance in Business Process Choreographer Explorer.

Before you begin

You must be logged in as administrator or operator to perform this task.

About this task

Examining the business process instance that generated the failed event can give you additional information about how or why the event failed. The business process instance and the failed event are linked by a common session ID.

Note: Not all failed events are generated from a business process instance.

To find and examine a business process instance related to a failed event, use the following procedure.

Procedure

1. From within the administrative console, use the failed event manager to locate the failed event you want to investigate. See “Finding failed events” on page 301 for instructions on how to search for failed events.
2. From the Failed Event Details page for that event, click **Open calling process in Business Process Choreographer Explorer**.

Results

The Business Process Choreographer Explorer opens in a new browser window and displays information about the related process instance.

Finding Common Base Events related to a failed event

A failed event can be related to one or more Common Base Events. The failed event manager provides a link to view related Common Base Events in the Common Base Event Browser.

Before you begin

You must be logged in as administrator or operator to perform this task.

About this task

Examining related Common Base Events can give you additional information about how or why the original event failed. The failed event and any related Common Base Events are linked by the same session ID.

To find and view related Common Base Events, use the following procedure.

Procedure

1. From within the administrative console, use the failed event manager to locate the failed event you want to investigate. See "Finding failed events" on page 301 for instructions on how to search for failed events.
2. From the Failed Event Details page for that event, click **Browse Related Common Base Events**.

Results

The Common Base Event Browser opens in a new browser window and lists any Common Base Events related to the original failed event.

Deleting failed events

If you do not want to resubmit a failed event, or if you have failed events that have expired, use the failed event manager to delete them from the server. The failed event manager provides three options for deleting failed events.

Before you begin

You must be logged in as administrator or operator to perform this task.

About this task

To delete one or more failed events, use the following procedure.

Procedure

1. Ensure that the failed event manager is open and that you have retrieved a list of the failed events on your system.
2. From the failed event manager's Search Results page, do one of the following:
   - If you want to delete one or more specific failed events, select the check box next to each event and then click **Delete**.
   - If you want to delete only those failed events that have expired, click **Delete expired events**. Note that this deletes only the expired events in the current set of search results.
   - If you want to delete all failed events on the server, click **Clear all**.

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 entering values in the Search page's **By Date** tab	"Values in the By Date and From Date field automatically change to default if entered incorrectly" on page 314
I am having trouble deleting expired events	"Using the Delete Expired Events function appears to suspend the failed event manager" on page 314

Problem	Refer to the following
I am having trouble with failed events not being created	"Failed events are not being created"

Values in the By Date and From Date field automatically change to default if entered incorrectly

The Search page's **From Date** and **To Date** fields require correctly formatted locale-dependent values. 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 workstation with an en_US locale in the Eastern Standard Time (EST) time zone.
The default value for the **To Date** field is always the current date and time, formatted for your locale and time zone.

To avoid this problem, always enter your dates and times carefully, following the example provided above each field.

Using the Delete Expired Events function appears to suspend the failed event manager

If you use the Delete Expired Events button in situations where there are many failed events in the current search results, or where those events contain a large amount of business data, the failed event manager can appear to be suspended indefinitely.

In this situation, the failed event manager is not suspended: it is working through the large data set, and will refresh the results set as soon as the command completes.

Failed events are not being created

If the Recovery subsystem is not creating failed events, go through the following checklist of potential causes:

- Ensure that the wpsFEMgr application is running. If necessary, restart it.
- Ensure that the failed event manager's database has been created, and that the connection has been tested.
- Ensure that the necessary failed event destination has been created on the SCA system bus. There should be one failed event destination for each deployment target.

- Ensure that the Quality of Service (QoS) **Reliability** qualifier has been set to `Assured` for any Service Component Architecture (SCA) implementation, interface, or partner reference that participates in events you want the Recovery service to handle.

Chapter 12. Troubleshooting 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 or within the service applications you are administering.

For information on troubleshooting Business Process Choreographer or Common Event Infrastructure components, see one of the following locations:

- The WebSphere Process Server for Multiplatforms, version 6.1, information center
- The *Business Process Choreographer* PDF
- The *Common Event Infrastructure* PDF

Troubleshooting administration tasks and tools

Use the information in this group of topics to identify and resolve problems that can occur while you are administering the runtime environment.

Profile-specific log files

There are log files detailing the characteristics and runtime activities of individual profiles. These log files are located within the profile directory for each profile.

There are a number of log files that are created for each profile. Some of these logs describe the parameters used for the creation of the profile. These types of log files generally remain unchanged once the profile is fully configured. Other profile-specific logs are continually updated to capture error, warning, and information messages emitted during runtime. Some of these log files are also used to capture a Common Base Event (that may include business object data) that is selected for monitoring.

The table below specifies the different types of profile-specific log files and the locations where you can find them within the product. Within the table, the variable *install_root* represents the installation directory of WebSphere Process Server. The variable *profile_root* represents the root location of a profile.

i5/OS **On i5/OS platforms:** The variable *user_data_root* represents the default user data directory.

For more information see Default installation directories for the product, profiles, and tools.

Table 43. Profile-specific log files updated during runtime

Log	Contents
First failure data capture (ffdc) log and exception files (common to all profile types) are found in these directories: • Linux UNIX **On Linux® and UNIX® platforms:** *profile_root*/logs/ffdc • Windows **On Windows platforms:** *profile_root*\logs\ffdc • i5/OS **On i5/OS platforms:** *profile_root*/logs/ffdc	Contains the ffdc log and exception files for individual profiles. There are two types of ffdc logs: a single log file with a compilation of all the errors encountered during the profile runtime, and numerous text files with details such as stack traces and other information. The naming conventions for the different types of profiles are given for both files, as follows: • **Deployment manager profile:** – Log file — *deployment_manager_name*_exception.log. – Text files — *deployment_manager_name_hex_id_date_time*.txt. • **Custom profile:** • – Log file(s) — *node_agent_name*_exception.log and *server_name*_exception.log . – Text files — *node_agent_name*(or)*server_name_hex_id_date_time*.txt. • **Stand-alone profile:** – Log file — *server_name*_exception.log. – Text files — *server_name_hex_id_date_time*.txt.
Deployment manager logs (deployment manager profiles only) are found in these directories: • Linux UNIX **On Linux and UNIX platforms:** *profile_root*/logs/*deployment_manager_name* • Windows **On Windows platforms:** *profile_root*\logs*deployment_manager_name* • i5/OS **On i5/OS platforms:** *profile_root*/logs/*deployment_manager_name*	You will work primarily with four log files in this directory: • startServer.log — Contains the system parameters detected on the system and the messages emitted by the deployment manager during the start process • stopServer.log — Contains the system parameters detected on the system and the messages emitted when the deployment manager is shut down. • SystemErr.log — Contains error and exception messages generated by the deployment manager during runtime. Continually updated while server is running. • SystemOut.log — Contains all messages, including error, warning, and information messages generated by the deployment manager during runtime. Continually updated while server is running.

Table 43. Profile-specific log files updated during runtime (continued)

Log	Contents
Node agent logs (custom profiles only) are found in these directories: • Linux UNIX **On Linux and UNIX platforms:** *profile_root*/logs/*node_agent_name* • Windows **On Windows platforms:** *profile_root*\logs*node_agent_name* • i5/OS **On i5/OS platforms:** *profile_root*/logs/*node_agent_name*	You will work primarily with four log files in this directory: • startServer.log — Contains the system parameters detected on the system and the messages emitted by the node agent during the start process • stopServer.log — Contains the system parameters detected on the system and the messages emitted when the node agent is shut down. • SystemErr.log — Contains error and exception messages generated by the node agent during runtime. Continually updated while node agent is running. • SystemOut.log — Contains all messages, including error, warning, and information messages generated by the node agent during runtime. Continually updated while the node agent is running.
Server logs (custom and stand-alone profiles only) are found in these directories: • Linux UNIX **On Linux and UNIX platforms:** *profile_root*/logs/*server_name* • Windows **On Windows platforms:** *profile_root*\logs*server_name* • i5/OS **On i5/OS platforms:** *profile_root*/logs/*server_name*	You will work primarily with four log files in this directory: • startServer.log — Contains the system parameters detected on the system and the messages emitted by the server during the start process • stopServer.log — Contains the system parameters detected on the system and the messages emitted when the server is shut down. • SystemErr.log — Contains error and exception messages generated by the server during runtime. Continually updated while server is running. • SystemOut.log — Contains all messages, including error, warning, and information messages generated by the server during runtime. Also contains any events being monitoring that are emitted from the Common Event Infrastructure (CEI), in Common Base Event format. These events may also include the level of business object data (FINE, FINER, or FINEST) that is specified for the monitor. Continually updated while the server is running.

Table 43. Profile-specific log files updated during runtime (continued)

Log	Contents
Node federation log files are found in these directories (only applies to non-deployment manager profiles): • Linux UNIX **On Linux and UNIX platforms:** *profile_root*/logs • Windows **On Windows platforms:** *profile_root*\logs • i5/OS **On i5/OS platforms:** *profile_root*/logs	Two log files are generated when you attempt to federate a custom, augmented, or stand-alone profile to a deployment manager: • addNode.log — contains the pertinent server environment information and messages generated when you attempt to federate the profile. • isFederated.log — lists the commands used by the deployment manager to federate the profile.
The location of the Integrated Solutions Console application deployment log file is listed here (only for deployment manager and stand-alone profiles): • Linux UNIX **On Linux and UNIX platforms:** *profile_root*/logs/iscinstall.log • Windows **On Windows platforms:** *profile_root*\logs\iscinstall.log • i5/OS **On i5/OS platforms:** *profile_root*/logs/iscinstall.log	The iscinstall.log file contains information regarding the deployment of the administrative console application in a deployment manager or stand-alone profile.
The location of the Installation Verification Tool log file is listed here (only for deployment manager and stand-alone profiles): • Linux UNIX **On Linux and UNIX platforms:** *profile_root*/logs/ivtClient.log • Windows **On Windows platforms:** *profile_root*\logs\ivtClient.log • i5/OS **On i5/OS platforms:** *profile_root*/logs/ivtClient.log	This log file contains the output generated by the Installation Verification Tool. You can start this program from the First Steps console after you create a deployment manager or stand-alone profile. The log contains basic configuration information and the messages that are displayed when you run the tool.
The location of the log file detailing the commands generated for a profile creation is listed here: • Linux UNIX **On Linux and UNIX platforms:** *profile_root*/logs/updateserverpolicy.log • Windows **On Windows platforms:** *profile_root*\logs\updateserverpolicy.log • i5/OS **On i5/OS platforms:** *profile_root*/logs/updateserverpolicy.log	This file contains the sequence of commands used by the product to set server environment variables and create a profile. All profile types will contain this file.

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 entering values in the Search page's **By Date** tab	"Values in the By Date and From Date field automatically change to default if entered incorrectly" on page 314
I am having trouble deleting expired events	"Using the Delete Expired Events function appears to suspend the failed event manager" on page 314
I am having trouble with failed events not being created	"Failed events are not being created" on page 314

Values in the By Date and From Date field automatically change to default if entered incorrectly

The Search page's **From Date** and **To Date** fields require correctly formatted locale-dependent values. 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 workstation with an en_US locale in the Eastern Standard Time (EST) time zone.
The default value for the **To Date** field is always the current date and time, formatted for your locale and time zone.

To avoid this problem, always enter your dates and times carefully, following the example provided above each field.

Using the Delete Expired Events function appears to suspend the failed event manager

If you use the Delete Expired Events button in situations where there are many failed events in the current search results, or where those events contain a large amount of business data, the failed event manager can appear to be suspended indefinitely.

In this situation, the failed event manager is not 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 the business rules manager

Some of the problems you might encounter using the business rules manager are login errors, login conflicts, and access conflicts.

You can take various steps to troubleshoot these problems.

Resolving login errors

A log in error occurs upon logging in.

Before you begin

About this task

The login error message is as follows:

```
Unable to process login. Please check User ID and password and try again.
```

Note: Login errors occur only when administrative security is enabled and either the user ID, password, or both, are incorrect.

To resolve login errors, perform the following steps.

Procedure

1. Click **OK** on the error message to return to the Login page.
2. Enter the valid **User ID** and **Password**.
   - If passwords are case sensitive, make sure that Caps Lock key is not on.
   - Make sure the user ID and password are spelled correctly.
   - Check with the system administrator to be sure that the user ID and password are correct.
3. Click **Login**.

What to do next

If you resolve the login error, you will now be able to log in to the business rules manager. If the error is not resolved, contact your system administrator.

Resolving login conflict errors

A login conflict error occurs when another user with the same user ID is already logged in to the application.

Before you begin

About this task

The login conflict message is as follows:

```
Another user is currently logged in with the same User ID. Select from the
following options:
```

Usually this error occurs when a user closed the browser without logging out. When this condition occurs, the next attempted login before the session timeout expires results in a login conflict.

Note: Login conflict errors occur only when administrative security is enabled.

To resolve login conflict errors, select from the following three options:

- Return to the Login page.

  Use this option if you want to open the application with a different user ID.
- Log out the other user with the same user ID.

  Use this option to log out the other user and start a new session.

  Note: Any unpublished local changes made in the other session will be lost.
- Inherit the context of the other user with the same user ID and log out that user.

  Use this option to continue work already in progress. All unpublished local changes in the previous session that have been saved will not be lost. The business rules manager will open to the last page displayed in the previous session.

Resolving access conflict errors

An access conflict error occurs when a business rule is updated in the data source by one user at the same time another user is updating the same rule.

Before you begin

This error is reported when you publish your local changes to the repository.

About this task

To correct access conflict errors, perform the following actions:

- Find the source of the business rule that is causing the error and check if your changes on the local machine are still valid. Your change may no longer be required after the changes done by another user.
- If you choose to continue working in the business rules manager, you must reload those business rule groups and rule schedules in error from the data source as your local changes of business rule groups and rule schedules in error are no longer usable. Reload a business rule group or rule schedule page, by clicking **Reload** in the Publish and Revert page of the rule for which the error was reported. You can still use local changes in other business rule groups and rule schedules that are not in error.

Troubleshooting deployed service applications

Use the information in this group of topics to identify and resolve errors in service applications deployed to the run time.

Using cross-component tracing for applications

Cross-component tracing allows you to identify trace.log data that is associated with WebSphere Process Server and WebSphere Enterprise Service Bus modules and components. The trace.log data can include error and event information, such as corrupted data or runtime exceptions, captured during Service Component Architecture (SCA) processing. The input and output data passing between WebSphere Process Server and WebSphere Enterprise Service Bus components can also be captured and used for problem determination using WebSphere Integration Developer.

Enabling Cross-Component Trace

Enable cross-component tracing to collect error and event information associated with WebSphere Process Server and WebSphere Enterprise Service Bus modules and components captured during processing.

Before you begin

You must be logged in as administrator to perform this task.

About this task

To enable cross-component tracing, use the following procedure.

Procedure

1. Ensure the administrative console is running, and then click **Troubleshooting** → **Cross-Component Trace** to display the Cross-Component Trace page.
2. Select a server for which you want to enable tracing. Note the status of the selected servers in the **Status** column: **running** or **not running**. If the status is **not running**, runtime tracing for that server is disabled and you can specify only the configuration value. In this case, the trace level takes effect only when the server starts or restarts.
3. In the **Configuration** or **Runtime** columns, select **enable** from the dropdown box for each server for which you want to enable tracing. Enable tracing in **Configuration** to collect data when the server starts or restarts. Enable tracing in **Runtime** to collect data when the server is currently running.
4. After you have specified the settings, click **OK** to save the settings.

Results

Collected data is added to the trace.log file and is purged as those files are purged. See "Troubleshooting Service Component Architecture (SCA) processing and call chains" on page 326 for more information.

Enabling Cross-Component Tracing with data snapshot

Enable cross-component tracing with data snapshot to collect data associated with Service Component Architecture (SCA) processing and call chain data associated with WebSphere Process Server and WebSphere Enterprise Service Bus modules and components.

Before you begin

You must be logged in as administrator to perform this task.

About this task

To enable cross-component tracing, use the following procedure.

Procedure

1. Ensure the administrative console is running, and then click **Troubleshooting** → **Cross-Component Trace** to display the Cross-Component Trace page.
2. Select the server for which you want to enable tracing. Note the status of the selected servers in the **Status** column: **running** or **not running**. If the status is **not running**, runtime tracing for that server is disabled and you can specify only the configuration value. In this case, the trace level takes effect only when the server starts or restarts.
3. In the **Configuration** or **Runtime** columns, select **enable with data snapshot** from the dropdown box for each server for which you want to enable tracing. Enable tracing in **Configuration** to collect data when the server starts or restarts. Enable tracing in **Runtime** to collect data when the server is currently running.
4. After you have specified the settings, click **OK** to save the settings.

Results

Collected SCA data is added to the systemout.log and trace.log files and is purged as those files are purged. Input and output data passing between WebSphere Process Server and WebSphere Enterprise Service Bus components is captured and additional files are created in the logs\XCT directory. This data can be used for problem determination by WebSphere Integration Developer. Deleting these files when they are no longer needed is a task for the administrator. See “Troubleshooting Service Component Architecture (SCA) processing and call chains” on page 326 for more information.

Disabling Cross-Component Trace

Disable cross-component tracing to stop the collection of error and event information associated with WebSphere Process Server and WebSphere Enterprise Service Bus modules and components captured during Service Component Architecture (SCA) processing.

Before you begin

You must be logged in as administrator to perform this task.

About this task

To disable cross-component tracing, use the following procedure.

Procedure

1. Ensure the administrative console is running, and then click **Troubleshooting** → **Cross-Component Trace** to display the Cross-Component Trace page.
2. Select a server for which you want to disable tracing. Note the status of the selected servers in the **Status** column: **running** or **not running**. If the status is **not running**, runtime tracing for that server is disabled and you can specify only the configuration value. In this case, the disabling tracing takes effect only when the server starts or restarts.
3. In the **Configuration** or **Runtime** columns, select **disable** from the dropdown box for each server for which you want to disable tracing.

4. After you have specified the settings, click **OK** to save the settings.

Results

Cross-component tracing is disabled for the selected servers. No data is collected.

Deleting data collected by Cross-Component Trace

When Cross-Component trace adds data to log files, and if enabled with data capture, additional files are created in the logs\XCT directory. The data added to the systemout.log and trace.log files does not need to be deleted as these files are automatically deleted by WebSphere Application Server. The data capture files need to be deleted from the logs\XCT directory manually when they are no longer needed. Delete these files after using WebSphere Integration Developer for problem determination or after the log files that refer to the files have been deleted by WebSphere Application Server.

Before you begin

You must have read and write access to the logs directories of each server.

About this task

To delete data collected by cross-component tracing, use the following procedure.

Procedure

1. Go to the logs\XCT directory in which the data was captured and move the contents of the directory to a location where it can be viewed by WebSphere Integration Developer for problem determination.
2. If you determine that the captured data is not needed for problem determination, then manually delete the contents of the logs\XCT directory.

Results

The captured data is deleted.

Troubleshooting Service Component Architecture (SCA) processing and call chains

Cross-component tracing allows you to identify systemout.log or trace.log data that is associated with WebSphere Process Server and WebSphere Enterprise Service Bus modules and components. The trace.log data can include error and event information, such as corrupted data or runtime exceptions, captured during SCA processing. The input and output data passing between WebSphere Process Server and WebSphere Enterprise Service Bus components can also be captured and used for problem determination using WebSphere Integration Developer.

Events that can be captured include:

- Errors that occur during processing because of corrupted data.
- Errors when resources are not available, or are failing.
- Interpretation of code paths.

You can access the Cross-Component Trace page from the administrative console and then clicking **Troubleshooting** → **Cross-Component Trace**. On this page, you can select the servers from which you want to collect trace data. Use the

Configuration column to specify trace settings for servers that will be used when the server starts or restarts. Use the **Runtime** column to specify trace settings for servers that are running.

For each server, you can use any of the following settings:

enable
: This setting enables tracing for SCA processing. Data collected from this setting is added to the systemout.log and trace.log files and is purged as those files are purged.

enable with data snapshot
: This setting enables tracing for SCA processing and input and output data that passes between WebSphere Process Server and WebSphere Enterprise Service Bus components. Data from SCA processing is added to the systemout.log and trace.log files. Input and output data from WebSphere Process Server and WebSphere Enterprise Service Bus components is placed in files that are created in the logs\XCT directory.

disable
: This setting disables tracing on the selected server.

Handling and deleting collected data

- Data collected from SCA processing is added to added to the systemout.log and trace.log files and is purged as those files are purged.
- Input and output data added as files in the logs\XCT directory can be moved to a location where it can be viewed by WebSphere Integration Developer for problem determination. The files can then be manually deleted. These input and output files in logs\XCT are related to the systemout.log and trace.log files that were created at the same time. When WebSphere Application Server deletes old systemout.log and trace.log files, the associated input and output files in logs\XCT can also be deleted. Generally, there will be many input and output files for a given systemout.log file. The timestamps in the systemout.log and trace.log files can be used to identify what input and output files to delete. It is safe to delete all input and output files older than the oldest date in the systemout.log and trace.log files.

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.

Event sequencing callback fails to release a lock

While trying to delete a failed sequenced event in the Recovery subsystem, the event sequencing callback can fail to release the event's lock. This typically occurs when a target application has been removed or when other components of the system (for example, the database) are unavailable.

In this situation, the failed event manager generates an error message. Use the esAdmin command to manually delete the lock associated with the failed event.

Performance issues

If you are experiencing memory problems on the messaging engine server used for event sequencing components, try modifying the runtime event sequencing properties in the *install_root*/properties/eventsequencing.properties file.

The maxActiveMessages property defines the number of messages currently locked on a component destination; too many large messages can negatively affect performance and cause memory problems. Note that a value of `0` (zero) means that an unlimited number of messages are allowed. By default, the maxActiveMessages property is set to 100. When changing the value, consider using the following formula where *delta* is the standard deviation of the accuracy of the estimate for the anticipated number of sequenced events with the same sequencing key that can be simultaneously processed.

```
average_number_of_ES_keys * average_number_of_potential_queued_events_per_key
+ delta
```

The workItemRetryCount property sets the upper boundary for the verification work retry count. A verification work item is spawned when an asynchronous event is unlocked and there are dependent events waiting to be processed. In this situation the creation and deletion of the lock are done in separate units of work and the work verification task ensures that the processing of one unit of work is complete before the next event is processed. By default, workItemRetryCount is set to -1 (retry).

The workItemSleepTime property specifies the amount of time that elapses between work verification retry attempts. By default, workItemSleepTime is set to 10 seconds. Note that lowering the value can decrease performance.

To modify any of the properties, perform the following steps.

1. Open the eventsequencing.properties file in a text editor.
2. Make the appropriate modifications for your environment.
3. Save and close the file.
4. Stop and restart any applications that are part of the event sequencing component in order for the changes to take effect.

Troubleshooting Service Component Architecture and WebSphere MQ communications

Communication between Service Component Architecture (SCA) modules and WebSphere MQ queue managers depends on the binding between the imports and exports within the SCA module and the queues in WebSphere MQ servers. Use this information to determine the servers that are not processing WebSphere MQ messages.

Before you begin

This task assumes that you have noticed requests dependant on WebSphere MQ are not being processed and that you have access to the administrative console. You should also either have the ability to make changes to the WebSphere MQ queue manager or be in contact with the WebSphere MQ administrator.

About this task

Service Component Architecture (SCA) modules depend on the bindings between the server and the WebSphere MQ queue manager. Communications between the two entities could keep messages from processing completely. The following steps should help you discover the cause of the disruption and what to do to get the messages processed again.

Procedure

1. Display the SCA module communicating with WebSphere MQ to make sure it is still processing. Navigate to this page using **Applications > SCA Modules**.
2. Display the queue manager to make sure it is still operational. Use WebSphere MQ administrative tools to perform this task.
3. Display the bindings between the SCA module and the queue manager to make sure the binding is correct. If the binding is incorrect, change the binding. Navigate to this page using **Applications** → **SCA modules** >*moduleName* > **Imports|Exports** > *importName|exportName* > **Bindings** > *bindingName* [type].
4. Locate any messages that may indicate failed transactions. You will have to investigate system, SCA-specific message areas, WebSphere MQ-specific message areas, the failed event queue and other locations to determine what has failed.

a. Examine SystemOut.log for any messages that would indicate processing failures.

   If there is an WebSphere MQ error, there will be an MQException linked somewhere in the stack trace with a WebSphere MQ reason code (for example, 2059 is “queue manager unavailable”).

b. Check AMQERR*xx*.LOG and the WebSphere MQ FFDC files to determine the cause of a WebSphere MQ error.

c. Examine the application queues to determine if there are any unprocessed messages. Make sure you examine both WebSphere MQ and Service Integration Bus (SIB) queues.

d. Examine the WebSphere MQ dead letter queue and the SIB exception destination.

e. Examine the failed event queue to determine if there are any messages related to the applications of interest. See Finding failed events for information about locating the failed events. See “Managing failed events” for information about locating the failed events.

Troubleshooting the object request broker (ORB) service settings

Setting **Pass by reference** to *true* on the Object Request Broker (ORB) service page of the administrative console might cause problems with serializing and de-serializing objects.

Object serialization problems and the *Pass by reference* property

SCA calls the ORB method `javax.rmi.CORBA.Util.copyObject()` to copy objects. If you enabled **Pass by reference** processing by checking the check box, a deep copy **is not made**, which causes problems with serializing and de-serializing objects.

Object serializing and de-serializing problems can result in communication issues between Service Component Architecture (SCA) modules. For example, if a `ServiceBusinessException` exception is thrown, it might not be reflected as such in the client end and, instead, might result in a `ServiceRuntimeException` exception.

Resolving object serialization problems caused by setting *Pass by reference* to True

To avoid object serialization problems in WebSphere Process Server, make sure **Pass by reference** is set to the default value. The default setting for **Pass by reference** is *false*, meaning that the check box for **Pass by reference** is not selected.

The following steps describe how to verify the **Pass by reference** property setting.

1. Navigate to the ORB service page of the administrative console.

   Application servers → **[ServerName]** → **Container Services** → **ORB Service**.

2. Make sure the check box for **Pass by reference** is NOT selected.

The Service Component Architecture depends on the setting of **Pass by reference** property to make a message copy.

Troubleshooting messaging bindings

Various error conditions can occur with bindings that are specific to the type of binding.

About this task

The manner in which error conditions are handled depends upon the type of binding concerned.

Troubleshooting JMS bindings

You can diagnose and fix problems with JMS bindings.

Implementation exceptions

In response to various error conditions, the JMS import and export implementation can return one of two types of exceptions:

- Service Business Exception: this exception is returned if the fault specified on the service business interface (WSDL port type) occurred.
- Service Runtime Exception: raised in all other cases. In most cases, the cause exception will contain the original exception (JMSException).

  For example, an import expects only one response message for each request message. If more than one response arrives, or if a late response (one for which the SCA response expiration has expired) arrives, a Service Runtime Exception is thrown. The transaction is rolled back, and the response message is backed out of the queue or handled by the failed event manager.

Primary failure conditions

The primary failure conditions of JMS bindings are determined by transactional semantics, by JMS provider configuration, or by reference to existing behavior in other components. The primary failure conditions include:

- Failure to connect to the JMS provider or destination.

  A failure to connect to the JMS provider to receive messages will result in the MDB Listener Port failing to start. This condition will be logged in the WebSphere Application Server log. Persistent messages will remain on the destination until they are successfully retrieved (or expired).

  A failure to connect to the JMS provider to send outbound messages will cause rollback of the transaction controlling the send.
- Failure to parse an inbound message or to construct an outbound message.

  A failure in the data binding or data handler causes rollback of the transaction controlling the work.
- Failure to send the outbound message.

  A failure to send a message causes rollback of the relevant transaction.
- Multiple or unexpected late response messages.

  The import expects only one response message for each request message. Also the valid time period in which a response can be received in determined by the SCA Response Expiration qualifier on the request. When a response arrives or the expiration time is exceeded, the correlation record is deleted. If response messages arrive unexpectedly or arrive late, a Service Runtime Exception is thrown.

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.

Troubleshooting Generic JMS bindings

You can diagnose and fix problems with Generic JMS bindings.

Implementation exceptions

In response to various error conditions, the Generic JMS import and export implementation can return one of two types of exceptions:

- Service Business Exception: this exception is returned if the fault specified on the service business interface (WSDL port type) occurred.
- Service Runtime Exception: raised in all other cases. In most cases, the cause exception will contain the original exception (JMSException).

Troubleshooting Generic JMS message expiry

A request message by the JMS provider is subject to expiration.

Request expiry refers to the expiration of a request message by the JMS provider when the JMSExpiration time on the request message is reached. As with other JMS bindings, the Generic JMS binding handles the request expiry by setting expiration on the callback message placed by the import to be the same as for the outgoing request. Notification of the expiration of the callback message will indicate that the request message has expired and the client should be notified by means of a business exception.

If the callback destination is moved to the third-party provider, however, this type of request expiry is not supported.

Response expiry refers to the expiration of a response message by the JMS provider when the JMSExpiration time on the response message is reached.

Response expiry for the generic JMS binding is not supported, because the exact expiry behavior of a third-party JMS provider is not defined. You can, however, check that the response is not expired if and when it is received.

For outbound request messages, the JMSExpiration value will be calculated from the time waited and from the requestExpiration values carried in the asyncHeader, if set.

Troubleshooting Generic JMS connection factory errors

When you define certain types of connection factories in your Generic JMS provider, you might receive an error message when you try to start an application. You can modify the external connection factory to avoid this problem.

When launching an application, you might receive the following error message:

```
MDB Listener Port JMSConnectionFactory type does not match
JMSDestination type
```

This problem can arise when you are defining external connection factories. Specifically, the exception can be thrown when you create a JMS 1.0.2 Topic Connection Factory, instead of a JMS 1.1 (unified) Connection Factory (that is, one that is able to support both point-to-point and publish/subscribe communication).

To resolve this issue, take the following steps:

1. Access the Generic JMS provider that you are using.
2. Replace the JMS 1.0.2 Topic Connection Factory that you defined with a JMS 1.1 (unified) Connection Factory.

When you launch the application with the newly defined JMS 1.1 Connection Factory, you should no longer receive an error message.

JMS-based events

If SCA messages originated through a JMS interaction fail, use the administrative facilities of the JMS provider to manage the events.

Troubleshooting WebSphere MQ bindings

You can diagnose and fix faults and failure conditions that occur with WebSphere MQ bindings.

Primary failure conditions

The primary failure conditions of WebSphere MQ bindings are determined by transactional semantics, by WebSphere MQ configuration, or by reference to existing behavior in other components.

Note: MQ bindings only support CLIENT transport mode.

The primary failure conditions include:

- Failure to connect to the WebSphere MQ queue manager or queue.

  A failure to connect to WebSphere MQ to receive messages will result in the MDB Listener Port failing to start. This condition will be logged in the WebSphere Application Server log. Persistent messages will remain on the WebSphere MQ queue until they are successfully retrieved (or expired by WebSphere MQ).

  A failure to connect to WebSphere MQ to send outbound messages will cause rollback of the transaction controlling the send.
- Failure to parse an inbound message or to construct an outbound message.

  A failure in the data binding causes rollback of the transaction controlling the work.
- Failure to send the outbound message.

  A failure to send a message causes rollback of the relevant transaction.
- Multiple or unexpected response messages.

  The import expects only one response message for each request message. If more than one response arrives, or if a late response (one for which the SCA response expiration has expired) arrives, a Service Runtime Exception is thrown. The transaction is rolled back, and the response message is backed out of the queue or handled by the failed event manager.

Misusage scenarios: comparison with WebSphere MQ JMS bindings

The WebSphere MQ import and export are principally designed to interoperate with native WebSphere MQ applications and expose the full content of the WebSphere MQ message body to mediations. The WebSphere MQ JMS binding, however, is designed to interoperate with JMS applications deployed against WebSphere MQ, which exposes messages according to the JMS message model.

The following scenarios should be built using the WebSphere MQ JMS binding, not the WebSphere MQ binding:

- Invoking a JMS message-driven bean (MDB) from an SCA module, where the MDB is deployed against the WebSphere MQ JMS provider. Use a WebSphere MQ JMS import.
- Allowing the SCA module to be called from a J2EE component servlet or EJB by way of JMS. Use a WebSphere MQ JMS export.
- Mediating the contents of a JMS MapMessage, in transit across WebSphere MQ. Use a WebSphere MQ JMS export and import in conjunction with the appropriate data binding.

There are situations in which the WebSphere MQ binding and WebSphere MQ JMS binding might be expected to interoperate. In particular, when you are bridging between J2EE and non-J2EE WebSphere MQ applications, use a WebSphere MQ export and WebSphere MQ JMS import (or vice versa) in conjunction with appropriate data bindings or mediation modules (or both).

Undelivered messages

If WebSphere MQ cannot deliver a message to its intended destination (because of configuration errors, for example), it sends the messages instead to a nominated dead-letter queue.

In doing so, it adds a dead-letter header to the start of the message body. This header contains the failure reasons, the original destination, and other information.

Troubleshooting a failed deployment

Use the information in this group of topics to identify and resolve errors in your deployment environment.

Troubleshooting your deployment environment

When processing appears sluggish or requests fail, use a focused approach to determine the source of the problem in the environment. The approach described is for non-standalone server environments.

Before you begin

You must be logged into the administrative console of the deployment manager to perform this task.

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

About this task

Investigate the state of your deployment environment if you notice any of the following symptoms:

- Unavailable applications
- Sluggish applications
- Stopped applications
- Decreased throughput
- Sluggish performance

Procedure

1. Display the topology layout that describes this deployment environment to determine the status of the topology.
2. Display the topology to determine the state of the various roles in the topology. Note the roles with unexpected states or warning for further investigation.
3. Locate the nodes that are causing the error state for each role.
4. Make sure all nodes are synchronized.

   On the Nodes page in the administrative console select any unsynchronized nodes and click **Synchronize**.
5. Make sure that the messaging engines associated with all the buses are running.

   If they are not running, stop and start the messaging engines.
6. Locate the logs associated with the nodes in error and view the logs for error messages.
7. Take any actions prescribed by the error messages to affect the correction.
8. Correct any errors and restart the affected nodes.

Results

The nodes previously in error start and the status of the topology becomes "running."

What to do next

Restart any affected applications

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 character set (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
1001 Hillsdale Blvd., Suite 400
Foster City, CA 94404
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows: (c) (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. (c) Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks and service marks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (^{R} or ^{TM}), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at ″Copyright and trademark information″ at www.ibm.com/legal/copytrade.shtml.

Microsoft and Windows are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Java and JavaScript are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

This product includes software developed by the Eclipse Project (http://www.eclipse.org).

IBM WebSphere Process Server for Multiplatforms, Version 6.2

Printed in USA