WebSphere Process Server for z/OS

IBM

Administering WebSphere Process Server

Version 7.0.0

Contents

Administering Business Process Choreographer. 167

Configuring and administering the Common Event Infrastructure 169

Administering service components 171

Working with adapters 217

Working with events 223

Troubleshooting administration . . . 259

Tables

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 architecture

The WebSphere Process Server administrative architecture consists of software processes called servers, topological units referenced as nodes and cells, and the configuration repository used for storing configuration information.

The server, node agent server, deployment manager, administrative agent, and job manager interact to perform system administration.

An administrator uses the administrative console to manage the entities that make up the administrative architecture.

Cells

Cells are logical groupings of one or more nodes in a WebSphere Process Server distributed network.

A cell is a configuration concept, a way for administrators to logically associate nodes with one another. Administrators define the nodes that make up a cell, according to the specific criteria that make sense in their organizational environments.

Administrative configuration data is stored in XML files. A cell retains master configuration files for each server in every node in the cell. Each node and server also have their own local configuration files. Changes to a local node or to a server configuration file are temporary, if the server belongs to the cell. While in effect, local changes override cell configurations. Changes to the master server and master node configuration files made at the cell level replace any temporary changes made at the node when the cell configuration documents are synchronized to the nodes. Synchronization occurs at designated events, such as when a server starts.

Servers

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

A process server can be either a *stand-alone server* or a *managed server*. A managed server can optionally be a member of a *cluster*. A collection of managed servers, clusters of servers, and other middleware is called a *deployment environment*. In a deployment environment, each of the managed servers or clusters is configured for a specific function within the deployment environment (for example, destination host, application module host, or Common Event Infrastructure server). A stand-alone server is configured to provide all of the required functions.

Servers provide the runtime environment for Service Component Architecture
(SCA) modules, for the resources that are used by those modules (data sources,
activation specifications, and JMS destinations), and for IBM-supplied resources
(message destinations, Business Process Choreographer containers, and Common
Event Infrastructure servers).

A *node agent* is an administrative agent that represents a node to your system and
manages the servers on that node. Node agents monitor servers on a host system
and route administrative requests to servers. The node agent is created when a
node is federated to a deployment manager.

A *deployment manager* is an administrative agent that provides a centralized
management view for multiple servers and clusters.

A stand-alone server is defined by a stand-alone profile; a deployment manager is
defined by a deployment manager profile; managed servers are created within a
managed node, which is defined by a custom profile.

Stand-alone server

A stand-alone server provides an environment for deploying Service Component
Architecture (SCA) modules in one server process. This server process includes,
but is not limited to, an administrative console, a deployment target, the messaging
support, the business rules manager, and a Common Event Infrastructure server.

You can deploy your own solutions to a stand-alone server, but a stand-alone
server cannot provide the capacity, scalability, or robustness that is required of a
production environment. For your production environment, it is better to use a
network deployment environment.

It is possible to start off with a stand-alone server and later include it in a network
deployment environment, by federating it to a deployment manager cell, *provided
that no other nodes have been federated to that cell*. It is not possible to federate
multiple stand-alone servers into one cell. To federate the stand-alone server, use
the administrative console of the deployment manager or the addNode command.
The stand-alone server must not be running when you federate it using the
addNode command.

A stand-alone server is defined by a stand-alone server profile.

Clusters

Clusters are groups of servers that are managed together and participate in
workload management.

A cluster can contain nodes or individual application servers. A node is usually a
physical computer system with a distinct host IP address that is running one or
more application servers. Clusters can be grouped under the configuration of a
cell, which logically associates many servers and clusters with different
configurations and applications with one another depending on the discretion of
the administrator and what makes sense in their organizational environments.

Clusters are responsible for balancing workload among servers. Servers that are a
part of a cluster are called cluster members. When you install an application on a
cluster, the application is automatically installed on each cluster member.

Because each cluster member contains the same applications, you can distribute client tasks in distributed platforms according to the capacities of the different machines by assigning weights to each server.

In distributed platforms, assigning weights to the servers in a cluster improves performance and failover. Tasks are assigned to servers that have the capacity to perform the task operations. If one server is unavailable to perform the task, it is assigned to another cluster member. This reassignment capability has obvious advantages over running a single application server that can become overloaded if too many requests are made.

Profiles

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

Using profiles, you can have more than one runtime environment on a system, without having to install multiple copies of the WebSphere Process Server binary files.

When you install WebSphere Application Server for z/OS®, you configure the product using the WebSphere Customization Tools application, which creates a profile. After you have installed WebSphere Process Server, you must augment the profile to define the node either as a deployment manager or stand-alone server for WebSphere Process Server. If you have created a deployment manager, you can then create one or more managed nodes. This automatically creates a profile to define the runtime environment for each node.

Note: On distributed platforms, each profile has a unique name. On the z/OS platform, all profiles are named "default"; you cannot rename, edit, copy, or delete profiles on z/OS.

The profile directory

Every profile in the system has its own directory containing all of its files. You specify the location of the profile directory when you create the profile. By default, it is in the `profiles` directory in the directory where WebSphere Process Server is installed. For example, `installation_file_system_root/AppServer/profiles/default`.

Deployment managers

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

You use the administrative console of the deployment manager to manage the servers and clusters in the cell. This includes configuring servers and clusters, adding servers to clusters, starting and stopping servers and clusters, and deploying Service Component Architecture (SCA) modules to them.

Although the deployment manager is a type of server, you cannot deploy modules to the deployment manager itself.

Nodes

A *node* is a logical grouping of managed servers.

A node usually corresponds to a logical or physical computer system with a distinct IP host address. Nodes cannot span multiple computers. Node names usually are identical to the host name for the computer.

Nodes in the network deployment topology can be managed or unmanaged. A managed node has a node agent process that manages its configuration and servers. Unmanaged nodes do not have a node agent.

Managed nodes

A *managed node* is a node that is federated to a deployment manager and contains a node agent and can contain managed servers. In a managed node, you can configure and run managed servers.

The servers that are configured on a managed node make up the resources of your deployment environment. These servers are created, configured, started, stopped, managed and deleted using the administrative console of the deployment manager.

A managed node has a node agent that manages all servers on a node.

When a node is federated, a node agent process is created automatically. This node agent must be running to be able to manage the configuration of the profile. For example, when you do the following tasks:

- Start and stop server processes.
- Synchronize configuration data on the deployment manager with the copy on the node.

However, the node agent does not need to be running in order for the applications to run or to configure resources in the node.

A managed node can contain one or more servers, which are managed by a deployment manager. You can deploy solutions to the servers in a managed node, but the managed node does not contain a sample applications gallery.

Unmanaged nodes

An unmanaged node does not have a node agent to manage its servers.

Unmanaged nodes in the Network Deployment topology can have server definitions such as Web servers , but not Application Server definitions. Unmanaged nodes can never be federated. That is, a node agent can never be added to an unmanaged node. Another type of unmanaged node is a stand-alone server. The deployment manager cannot manage this stand-alone server because it is not known to the cell. A stand-alone server can be federated. When it is federated, a node agent is automatically created. The node becomes a managed node in the cell.

Node agents

Node agents are administrative agents that route administrative requests to servers.

A node agent is a server that runs on every host computer system that participates in the Network Deployment configuration. It is purely an administrative agent and is not involved in application serving functions. A node agent also hosts other

important administrative functions such as file transfer services, configuration synchronization, and performance monitoring.

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 product-specific Welcome pages for each product you have installed. Use these pages to see detailed information on using each product.

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

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.
- **Servers → Server Types → WebSphere 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
  - REST Services 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** → **WebSphere application server 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
  – REST Services 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 for a specific task, surrounded by the following information to help you:
- An introduction to the task and its essential concepts
- A description of when and why to do this task
- A list of other tasks to do before and after 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 on page 8 shows an example of the administrative console displaying a guided activity.
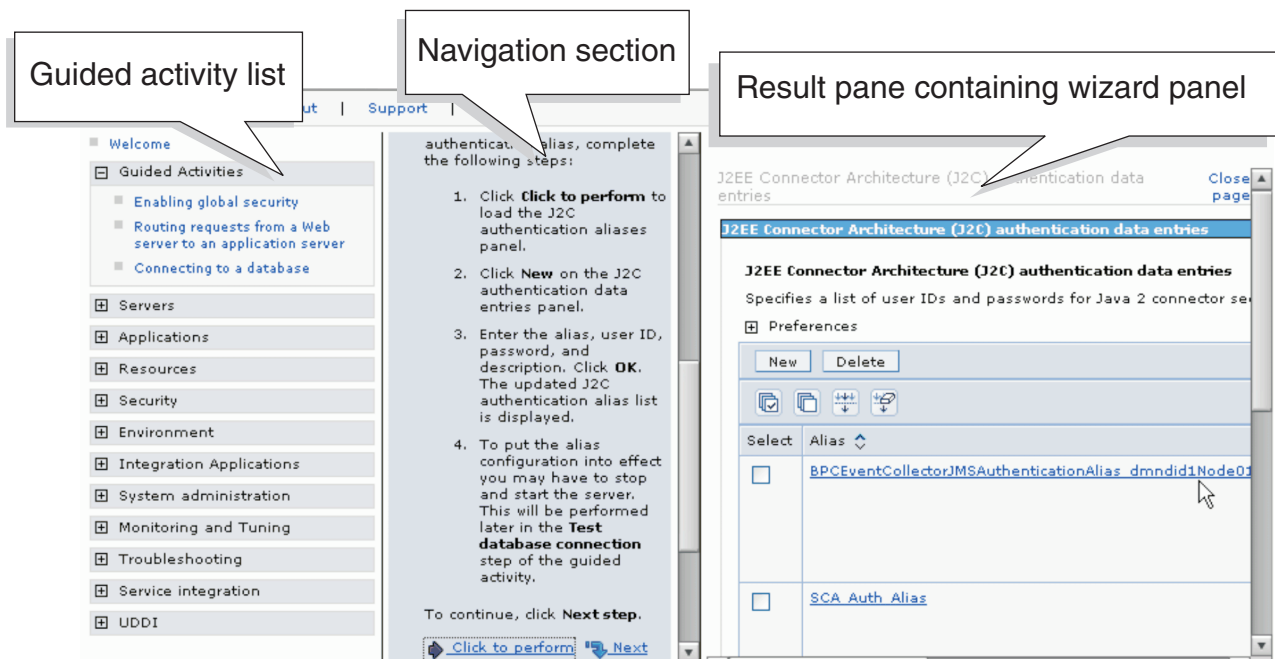
Figure 1. A guided activity

Administrative console pages

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

- "Collection pages"
- "Detail pages" on page 9
- "Wizard pages" on page 9

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 9. 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 consisting 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 7.

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 1. 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 17.
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 2. 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 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 programming interfaces

Administrative programming interfaces 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 Programming information 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 3. 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.

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 17.

- **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 17.

- **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 and running a script, as described in Changing the console session expiration.

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

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

      where *your_fully_qualified_server_name* specifies the fully qualified host name for the system that contains the administrative server and *portNumber* is the administrative console port number.

      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.
   d. Check the `job 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`.

   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" 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 developerWorks.

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.
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 corner of 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 information

Administrative console actions with command assistance

↪ 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 concepts

"Business Process Choreographer Explorer user interface"
Business Process Choreographer Explorer is a standalone 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.

Business Process Choreographer Explorer user interface

Business Process Choreographer Explorer is a standalone 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.

Taskbar	
Navigation pane	**Page title**

Action bar

| Start | Stop | Complete |

Content...
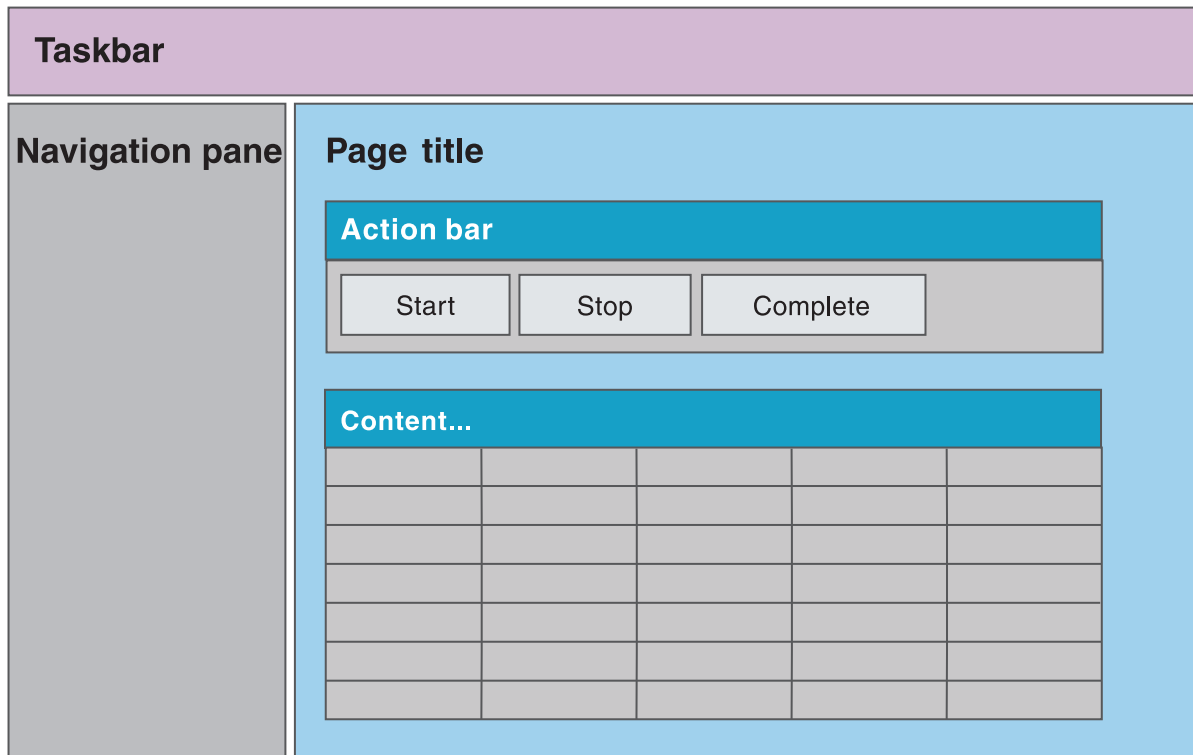
The user interface has the following main areas.

Taskbar

For all users, the taskbar offers options for logging out of Business Process Choreographer Explorer 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.

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 (![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:

Currently Valid
This view shows a list of process templates are the currently valid version of each process. That is, they are the newest started version of a process whose valid from date is not in the future. 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.

All Versions

This view shows a list of process templates for all process versions. From this view you can display information about a process template for a process version 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:

Stopped Activities

This view shows the activities that are in the stopped state.

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.

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 that is shown after you log in. If you have one of the system administrator roles, you can customize both the predefined views that are shown to your users and the default view that they see when they log in.

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 set of views, the default login view, and to define new views for 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. Define new 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, select a query table for your view.

      A default query table is set for your view definition. You can either select a different query table, or choose not to use query tables in your view definition.

      Note: If you use a query table, you cannot specify additional search criteria here for the view. All the search criteria must be defined in the query table definition.

      If you are not using a query table, specify 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. If you are using query tables and the query table definition has parameters, specify the query parameters that are needed on the Query Properties tab.

The parameter names that you specify must match the names in the query table definition. You can also provide default values for the parameters, and specify whether a default value can be overwritten when the query for the view is run.

e. 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**.

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

The search is run to check for errors. If it runs without errors, the view is saved.

The new view appears in your navigation pane. Users see the new view when they next log into 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 **Check and Save**.

   The search is run to check for errors. If it runs without errors, the view is saved.

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 **Check and Save**.

  The search is run to check for errors. If it runs without errors, the view is saved.

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 Customized Views page, where ... is the type of view, for example Process Templates, select a query table for your view.

   A default query table is set for your view definition. You can either select a different query table, or choose not to use query tables in your view definition.

   Note: If you use a query table, you cannot specify additional search criteria here for the view. All the search criteria must be defined in the query table definition.

   If you are not using a query table, specify 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. If you are using query tables and the query table definition has parameters, specify the query parameters that are needed on the Query Properties tab.

   The parameter names that you specify must match the names in the query table definition. You can also provide default values for the parameters, and specify whether a default value can be overwritten when the query for the view is run.

4. 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**.

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

   The search is run to check for errors. If it runs without errors, the view is saved. 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" on page 22
Business Process Choreographer Explorer is a standalone 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" on page 37
- "Menu bar" on page 37
- "Login page" on page 37
- "Navigator" on page 37
- "Content panels" on page 38
- "Command bar" on page 38
- "Lists" on page 38
- "Details panel" on page 38
- "Message data" on page 39
- "Tabbed panes" on page 39
- "Search pages" on page 39
- "Error details" on page 39

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.

Style name	Description
.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.
.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.

Style name	Description
.errorDetailsMessage	Text style for error message.

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** → **Server Types** → **WebSphere 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

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 the servers you need.

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 configuration.

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:
   - Use the startManager command.
   - Use the z/OS START command.
2. Verify that the deployment manager started successfully by checking the SYSPRINT DD statement of the WebSphere Process Server started task for the message ExtendedMessage: BBO00222I: WSVR0001I: Server CONTROL PROCESS *server_name* open for e-business.

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:
   - From the administrative console, click **System administration** ⟩ **Deployment manager** ⟩ **Stop** ⟩ **OK**. The administrative console closes before the server stops running.
   - Use the stopManager command.
   - Use the z/OS STOP command.

2. If you used the STOP command, verify that the deployment manager stopped successfully by checking the SYSPRINT DD statement of the WebSphere Process Server started task 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 SYSPRINT DD statement of the WebSphere Process Server started task. If the server started successfully, the SYSPRINT DD statement contains 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 node agent.

About this task

To restart 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 **Restart**. The node agent is stopped and then restarted.

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 configurator 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 reference

➡ Cluster, single server and node status
Describes the status of specific entities within a deployment environment.

➡ Deployment Environment function status
Describes the state of the minimum required entities and the redundant entities of a configured deployment environment.

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

Related information

➡ Using the Administration Thin Client

startDeploymentEnv command

stopDeploymentEnv command

Starting the deployment environment using the command line

You can start the deployment environment using the wsadmin.sh command.

Before you begin

Ensure the wsadmin.sh 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.sh command, perform the following steps.

Procedure
1. Open a command window.
2. At the command prompt, enter the wsadmin.sh command to enter the command environment. The wsadmin.sh 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.sh client from the deployment manager bin folder, you do not need to include the `-host` and `-port` parameters in the command.

```
wsadmin.sh -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.sh command.

Before you begin

Ensure the wsadmin.sh 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.sh command, perform the following steps.

Procedure
1. Open a command window.

2. At the command prompt, enter the wsadmin.sh command to enter the command environment. The wsadmin.sh 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.sh -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** › **Server Types** › **WebSphere 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** → **WebSphere application server clusters**.
2. Select the cluster you want to start.
3. Click **Start** or **Ripplestart** to 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.

   • **Start** launches the server process of each member of the cluster by calling the node agent for each server to start the servers. If your servers are stopped, select the **Start** option. If a call to a node agent for a server fails, the server does not start.

   • **Ripplestart** combines stopping and starting operations. If your servers are running, select the **Ripplestart** option. It first stops and then restarts each member of the cluster. For example, your cluster contains 3 cluster members:
     – server_1
     – server_2
     – server_3

   When you click **Ripplestart**, server_1 stops and restarts, then server_2 stops and restarts, and finally server_3 stops and restarts.

   Use **Ripplestart** instead of manually stopping and then starting all of the application servers in the cluster.

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

   Attention:　Do not perform a ripplestart on multiple clusters simultaneously. If plan on using **Ripplestart** to start clusters, do so on one cluster at a time.

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** → **WebSphere application server 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.

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 configurator 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

Configuring host aliases

Configuring authentication aliases for a deployment environment

Configuring custom deployment environments

Generating 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.

Use the procedure described in this task to modify the deployment topology in the following ways:

- Before the deployment environment is generated, so that only the deployment environment definition is updated
- After the deployment environment has been generated, as a means to create or delete cluster members

  If the deployment environment has already been generated, then the changes you make to the topology configuration are used to update the number of cluster members for the corresponding deployment environment clusters. For example, if you use the Deployment Topology panel to add a node, then the deployment environment definition is updated by that one node and all clusters that are managed by the deployment environment are adjusted by one more cluster member on that node (or the number of cluster members as defined by this panel). This adjustment is made to the topology configuration when you click **OK** or **Apply** on the Deployment Topology panel.

Procedure

- Select an objective and perform the associated actions.

Objective	Actions
Add node	Select a node from the pull-down list and click **Add**.

Objective	Actions
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.
Reset number of clusters assigned to each function	If you modified the number of clusters assigned to each cluster type, and you want to reset the parameters to the original value, click **Reset**
Remove a node from the deployment topology	For the node that you want to remove, select the check box in the Select column of the table and click **Remove**

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

What to do next

Either save the changes or discard them.

Related concepts

Custom deployment environment layout configuration
This overview describes two major configuration considerations for custom deployment environments: selecting clusters and single servers to use with the environment and specifying the deployment environment configuration. An understanding of these considerations enables you to plan and implement a deployment environment effectively.

Related reference

Cluster, single server and node status
Describes the status of specific entities within a deployment environment.

Deployment Environment function status
Describes the state of the minimum required entities and the redundant entities of a configured deployment environment.

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

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 configurator 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.

Configuration area	Available actions
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 tasks

"Modifying the deployment topology" on page 52
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.

Related information

Configuring custom deployment environments

Configuring data sources for a topology

Configuring authentication aliases 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
Configuring databases
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
configurator 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
configurator 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

When you make configuration changes, you must stop and restart a cluster member.You can stop and restart a cluster member by using one of three methods: the administrative console, the MVS console or by issuing USS commands.

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.

You can use any of the following methods to stop and restart a server:
* Administrative console
* MVS console
* USS command

Procedure

1. To stop and restart a server using the administrative console complete the following steps:

   a. In the administrative console, navigate to **Servers** ⇒ **Server Types** ⇒ **WebSphere application servers**.

   b. Select the servers or cluster members that you want to stop. Click **Stop**, then wait for the cluster members to stop.

   c. Select the servers or cluster members that you want to restart. Click **Start**, then wait for the servers to start.

   Note: Alternatively, you can stop and restart the complete cluster from the administrative console cluster panel by selecting **Servers** ⇒ **Clusters** ⇒ **WebSphere application server clusters** ⇒ *cluster_name*.

2. To use the MVS console to stop and restart a server complete the following steps:

   a. From the MVS console, issue the following **STOP** command:

      STOP *appserver_proc_name*,JOBNAME=*server_shortname*

      appserver_proc_name
      > Is the JCL procedure name in the proclib that is used to start the server.

      server_shortname
      > Is the short name of the server which is also used as the process jobname.

      For example:

      STOP BBO7ACR,JOBNAME=BBOS001

   b. To restart a server, issue the START command from the MVS console.

      START *appserver_proc_name*,JOBNAME=*server_shortname*,ENV=*cell_shortname.node_shortname.server*

      appserver_proc_name
      > The JCL procedure name in the proclib that is used to start the server.

      server_shortname
      > The short name of the server, which is also used as the process jobname.

      cellname.nodename.servername
      > This element of the ENV parameter is a concatenation of the cell short name, the node short name, and the server short name.

      For example:

      START BBO7ACR,JOBNAME=BBOS001,ENV=BBOBASE.BBONODE.BBOS001

      The following messages indicate that the controller is running:

      ```
      $HASP100 BBO7ACR  ON STCINRDR
      $HASP373 BBO7ACR  STARTED
      BBOO0001I WEBSPHERE FOR Z/OS CONTROL PROCESS BBODMNB/SY1/BBOC001/BBOS001
                IS STARTING.
      IRR812I PROFILE BBO*.* (G) IN THE STARTED CLASS WAS USED TO START BBOS001S
              WITH JOBNAME BBOS001S.
      ```

```
                    $HASP100 BBOS001S ON STCINRDR
                    $HASP373 BBOS001S STARTED
                    +BBO00004I WEBSPHERE FOR Z/OS SERVANT PROCESS BBODMNB/SY1/BBOC001/BBOS001
                               IS STARTING.
                    +BBO00020I INITIALIZATION COMPLETE FOR WEBSPHERE FOR Z/OS SERVANT PROCESS
                               BBOS001.
                    BBO00019I INITIALIZATION COMPLETE FOR WEBSPHERE FOR Z/OS CONTROL PROCESS
                               BBOS001.
```

3. To stop and restart a server using a USS command complete the following
   steps:

   a. Issue a **stopServer** command from the command line.

   b. Issue a **startServer** command.

      You can check that the server has successfully started by checking the
      startServer.log file. If the server has successfully started, the last two lines
      of the startServer.log file read:

      ```
      Server launched.  Waiting for initialization status.
      Server server1 open for e-business; process id is 1932.
      ```

      The startServer.log file is located in the profile_root/logs/server1/ directory

Related information

➙ Directory conventions: WebSphere Application Server

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
configurator 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 reference

▶ Cluster, single server and node status
Describes the status of specific entities within a deployment environment.

▶ Deployment Environment function status
Describes the state of the minimum required entities and the redundant entities of a configured deployment environment.

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

Related information

▶ Using the Administration Thin Client

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 configurator 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.sh command. You can use the wsadmin.sh 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.sh reduces the time for performing the task.

Procedure

1. Open a command window.

   The wsadmin.sh 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.sh command to enter the wsadmin.sh 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.sh -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.

Importing deployment environment definitions based on design documents using the administrative console

You can import an existing deployment environment definition based on a design document 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 design document from another deployment manager.
- You must be able to access the deployment environment design document (an XML file) from the deployment manager into which you are importing the deployment environment design.
- The deployment manager that imports the deployment environment definition must support at least all of the functions that are defined in the deployment environment design document. For example, you can import a deployment environment design 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 configurator to perform this task.

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

About this task

Importing an existing deployment environment design 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 **Create a deployment environment based on an imported design** selected.

2. Click **Browse** to open a file dialog and select the deployment environment design document (XML 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.

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 available if the nodes selected do not meet the constraints imposed by the imported deployment environment design. 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. Optional: On the Clusters page, assign the required number of cluster members on each node for each cluster *type* (Application Deployment Target, Messaging Infrastructure and Supporting Infrastructure) of the deployment environment.

   By default one cluster member is assigned on each node for each function. You change the number by replacing the number in each column. If you are unfamiliar with the different cluster roles and functions provided by each type of cluster, see "Topology types and deployment environment patterns."

   A 0 (zero) value for a node means that the node does not contribute to the selected function, based on features that you have selected.

   After assigning cluster members, you can click **Next** to display the Cluster naming pages for each cluster type of the deployment environment. The Cluster naming sub-steps that display will vary depending on the deployment environment pattern selected.

   The system generates default values for cluster names and cluster member names. The system also generates default values for the cluster short name and cluster member short name.

   If you do not want to customize cluster names or cluster member names, you can use the wizard navigation pane to go directly to the REST Services page in a following step.

   Each substep page is structured in the same fashion, and is described in Customize the cluster names and cluster member names.

   a. Optional:

      Use the Cluster Naming page to customize cluster names or cluster member names for the cluster type. You can also modify cluster short names and cluster member short names. There is one substep page for each cluster *type* in the pattern that you have selected. For example, if you selected a **Remote messaging and remote support pattern**, there are 3 sub-steps, one for each type of cluster (Application Deployment Target, Messaging Infrastructure and Supporting Infrastructure) in that pattern.

      The information on each substep page is as follows:

      Cluster
         A read-only field specifying the functional role of the cluster.

         The value varies depending on the cluster type, as follows:
         • Application Deployment Target
         • Supporting Infrastructure
         • Messaging Infrastructure

         For information on the functional role provided by each cluster type, see doc/cpln_top_types.dita

      Cluster Name
         Contains the system-generated default value for the cluster name.

Cluster Short Name

You can leave this field blank or enter a short name of your choosing.

Cluster Member Name

Accept the system-generated default value or specify a name of your choosing.

The default value for the cluster member name is based on the following naming convention: `<cluster name>.<node name>.<node number sequence>` .

The number of cluster member names that display in the table match the number of cluster members that you entered for the cluster type column and node row on the Clusters page. See the preceding step for the Clusters page.

Cluster Member Short Name

Accept the system-generated default value or specify name of your choosing.

The system-generated value for cluster member short name is based on a naming convention of `<deployment environment name>[0:5]<cluster type name>`.

The cluster member short name is limited to 7 characters and **MUST BE UNIQUE**.

If the cluster member short name is not unique, the system appends a unique number to it.

As an example, for a deployment environment named `DEMO`, the system-generated short name for the *application target* cluster member is `DEMO`**AT**.

The option for cluster member short name displays when the following configuration conditions exist:

- If any one known node in the cell is on a z/OS platform, then the cluster member short name displays. The node metadata should support the platform on which the node resides.
- If the Deployment Manager resides on a z/OS platform.

6. On the REST Services page, configure service endpoints for Representational State Transfer (REST) application programming interfaces (APIs).

   If you want widgets to be available in Business Space, you must configure the REST service endpoints for those widgets.

   a. Configure a full URL path for all REST services by selecting either **https://** or **http://** from the **Protocol** list.

   b. Enter a name in the **Host Name or Virtual Host in a Load-Balanced Environment** field.

   c. In the **Port** field, enter the port that a client needs to communicate with the server or cluster.

   d. In the table of REST services, if you want to modify the description of the REST service endpoint, overtype the entry in the Description field. The other fields are read-only.

   e. Click **Next** to go to the Import the database configuration page.

7. Optional: On the Import the database configuration page, click **Browse** to go the database design document or enter the path to the database design document and then click **Next** to go to the Data sources page. The design document can be based on a database design that you created using the

database design tool (DDT), or it can be the supplied design document based on the pattern and feature that you have selected.

Note: The database design document that you import for the deployment environment does not change the commonDB created at Profile Creation time.

8. Conditional optional: Database page, configure the database parameters for data sources of the deployment environment, then click **Next** to go to the Security page.

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.

Note: If you imported a database design document, the information on the Database page reflects the data source configuration as it exists in the database design document that you imported.

Whether or not this step displays for a fast path deployment environment configuration is conditional. This step displays for a fast path deployment environment configuration if more than one database has been defined.

This step always displays if you are using DB2 for z/OS or an Oracle database provider.

The default schema names that are displayed on this page might conflict with your site naming convention or might conflict with existing schemas. As such, it is likely that you will need to change the schema name.

Oracle database considerations:

- If you do not want to provide a DBA user name and password for all components when using Oracle, clear **Create tables** and specify preexisting and unique user names and passwords for each component. If you are able to provide a DBA user name and password for all the components, select **Create tables** and allow the configuration process to create the required schemas and users.

For a production environment, you should set the same values for **User name** and **Schema name** and you should deselect **Create tables**. For a production environment, create the required schemas manually and use the SQL files generated to create the tables.

Note: You cannot select **Create tables** for Business Space (the option is unavailable for selection). The SQL files for Business Space need to be run manually. For information on running the SQL manually for Business Space, see *Configuring Business Space database tables*.

You can edit all key parameters, such as the database name, whether or not to create tables, the data source runtime user name, and the password for the deployment environment.

You can select which database to use for the given component.

DB2 for z/OS: The **Create tables** option cannot be used if you are using a DB2 for z/OS database provider.

Steps that cannot be completed through the Deployment Environment Configuration wizard, and which need to be completed manually, are listed on the Deferred Configuration page.

9. On the Security page, configure the authentication aliases WebSphere uses when accessing secure components

You can change the authentication alias user name and password on this page. These aliases are used to access secure components but do not provide access to data sources

10. On the Business Process Choreographer page, set parameters for the Business Process Choreographer configuration and then click **Next** to display the System web applications page. On this page you specify the values for:
   - Security roles
   - Authentication aliases

11. Optional: On the System web applications page, set the context root for component-based web applications in your deployment environment or accept the system-provided default values for the context roots. Then click **Next** to display the Summary page.

   The System web applications page displays for deployment environments using the Remote messaging, support and web applications pattern. The Remote messaging, support and web applications pattern applies if the deployment environment is for a deployment manager that has been augmented to include WebSphere Business Monitor.

   The table contains the following control information.

   Web Application
   The name of the Web application.

   Some of the components that are part of the deployment environment you are creating contain web applications. The **Web application** column can include the following components:
   - Business Space
   - Business Process Choreographer Explorer
   - Business Rules Manager

   Context Root
   The current value of the context root for the component.

   By default, the default context root for the web application applies. You can change the context roots by typing over the value in the **Context Root** field.

   Note: The Business Space context root is read only and cannot be edited.

   Description
   The description of the Web application context root.

12. 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**.

   Clicking **Finish** saves the deployment environment configuration - but does not generate it.

   Click **Cancel** cancels the deployment configuration and does not save the configuration.

   a. Check for deferred configuration steps

      Select **Deployment Environments** → *name of deployment environment* → **Deferred Configuration**

      You need to address any existing deferred configuration steps before starting the Deployment Environment.

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" on page 61
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.sh command. You can use the wsadmin.sh 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.
- 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 from which you are importing the deployment environment definition must at least support all the functions defined in the deployment environment design. 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 designs; using wsadmin reduces the time for performing the task.

Procedure

1. Telnet into USS (Unix System Services).
2. Copy the deployment environment definition XML file you are importing to your chosen directory
3. Enter the wsadmin.sh command from the <WAS_HOME>/profiles/<dmgr profile>/bin or the <WAS_HOME>/bin directory.
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.sh -connType SOAP -host myDmgr -port 8879 -user dmgrAdmin -password -dmgrPass
> $AdminTask importDeploymentEnvDef {-filePath
/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 do 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

Generating deployment environments using the command line

📤 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 configurator 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" on page 61
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.

Administering applications and application services

Applications for WebSphere Process Server involve similar administration tasks and interfaces as Java EE 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.

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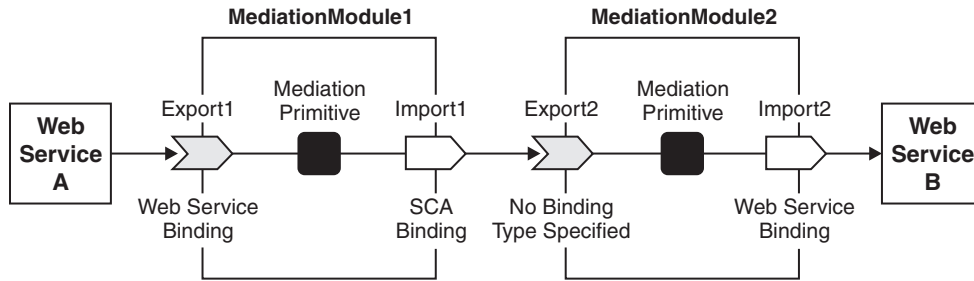
Figure 2. Example showing one mediation module interacting with another mediation module.
Mediation Module1 connects to Mediation Module2

Administering service modules in the administrative console

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*. **Note:** Make sure you stop any running Business Process Execution Language (BPEL) instances before stopping the application.

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.

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.

Working with imports and exports

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

Displaying an import or export interface:

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

About this task

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

Procedure
1. From the console navigation pane, click **Applications** → **SCA Modules** →
   moduleName to display the SCA Modules detail page for that module.
2. From the SCA Modules detail page, do one of the following tasks, depending
   on the type of interface you want to view.

Option	Description
Viewing an import interface	1. In the content pane, expand **Imports** to list all imports associated with the module. 2. Expand the import you want to view, and then expand **Interfaces** to display the import interfaces. 3. Select the interface you want to display.
Viewing an export interface	1. In the content pane, expand **Exports** to list all exports associated with the module. 2. Expand the export you want to view, and then expand **Interfaces** to display the export interfaces. 3. Select the interface you want to display.

Results

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

Displaying an import or export binding:

You can display details about import and export bindings after you deploy service
modules to WebSphere Process Server.

About this task

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

Procedure
1. From the console navigation pane, click **Applications** → **SCA Modules** →
   moduleName to display the SCA Modules detail page for that module.
2. From the SCA Modules detail page, do one of the following tasks, depending
   on the type of binding you want to view.

Option	Description
Viewing an import binding	1. In the content pane, expand **Imports** to list all imports associated with the module. 2. Expand the import you want to view, and then expand **Bindings** to display the import bindings. 3. Select the binding you want to display.
Viewing an export binding	1. In the content pane, expand **Exports** to list all exports associated with the module. 2. Expand the export you want to view, and then expand **Bindings** to display the export bindings. 3. Select the binding you want to display.

Results

The content pane displays the import or export binding details.

Administering bindings:

You can display information about import and export bindings and, in some cases, you can update the properties of the bindings. You use the administrative console to display and change information related to bindings. You can also use commands to show and modify import and export binding information.

Administering SCA bindings:

You can view information about the SCA import and export bindings of a module after the module has been deployed to the server. You can also reconfigure selected properties of an SCA import binding.

Viewing and updating SCA import bindings:

Using the administrative console, you can view information about a Service Component Architecture (SCA) import binding and change the target of the associated module.

Before you begin

To perform this task, you must have permission to change the master configuration.

About this task

To view information about an SCA import binding or to change the target of the associated module, use the administrative console to complete the following steps.

Procedure

1. Select the module that contains the 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 binding to view information about its properties.

- **Module** identifies the module that contains the import with this import binding.

- **Version** displays the SCA module version, if the module is versioned.

- **Cell ID** identifies the SCA module instance in the cell.

- **Import** identifies the import that contains the selected import binding.

- **Import interfaces** contains the list of interfaces for the import of this module.

3. To select a new target SCA module, perform the following steps:

a. Select a module from the **Target** list.

   Selecting another SCA module changes the exports and export interfaces that are displayed.

b. Select an export from the **Export** list.

4. Save your changes to the master configuration.

Results

If you made any updates, the binding is changed for the selected module. The changes take effect after you restart the SCA module.

Restriction: If the module is redeployed, the configuration changes you made are replaced by the original settings.

To ensure that the changes you made remain with the module across deployments, use WebSphere Integration Developer to make the changes in the source code for the module.

Viewing SCA export bindings:

Using the administrative console, you can view information about a Service Component Architecture (SCA) export binding, such as the name of the associated module and the name of the Web Services Description Language (WSDL) file.

About this task

To view information about an SCA export binding, use the administrative console to complete the following steps.

Procedure

1. Select the module that contains the 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 binding to view information about its properties.

- **Module** identifies the module that contains the export with this export binding.

- **Version** displays the SCA module version, if the module is versioned.

- **Cell ID** identifies the SCA module instance in the cell.

- **Export** identifies the export that contains the selected export binding.
- **Export interfaces** contains the list of interfaces for the export of this module.

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 import bindings and configure policy sets for the bindings.

Viewing and updating Web service import bindings:

Using the administrative console, you can view information about a Web service import binding and change the endpoint URL. For the Java API for XML Web Services (JAX-WS) bindings, you can also 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

The steps for administering Web service bindings depend on the type of binding:
- For JAX-RPC bindings, you can view attributes of the binding and edit the target endpoint.
- For JAX-WS bindings, you can view attributes of the binding, edit the target endpoint, and configure policy sets.

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: The **Policy set attachments** section of the administrative console page appears only for JAX-WS bindings. It does not appear for JAX-RPC service bindings.

Procedure
1. Select the module that contains the 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 binding to view information about its properties.
3. Change the value of **Target endpoint address**, which is the location of the Web service, and then click **Apply** or **OK**.
4. For JAX-WS bindings only, configure policy sets for import bindings by performing the following tasks:

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, and click **Attach** to attach a policy set to the binding, or click **Detach** to remove the policy set.

d. To assign a policy set binding, select the import binding, click **Assign Binding**, and provide a name for the policy set binding.

e. Repeat steps 4c and 4d for each binding you want to configure.

5. Save your changes to the master configuration.

Results

If you made any updates, the binding is changed for the selected module. The changes take effect after you restart the SCA module.

Restriction: If the module is redeployed, the configuration changes you made are replaced by the original settings.

To ensure that the changes you made remain with the module across deployments, use WebSphere Integration Developer to make the changes in the source code for the module.

Viewing and updating Web service export bindings:

Using the administrative console, you can view information about a Web service export binding (including the WSDL file) and configure properties of the associated Web module. For the Java API for XML Web Services (JAX-WS) bindings, you can also 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

The steps for administering Web service bindings depend on the type of binding:
- For JAX-RPC bindings, you can view attributes of the binding.
- For JAX-WS bindings, you can view attributes of the binding and configure policy sets.

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: The **Policy set attachments** section of the administrative console page appears only for JAX-WS bindings. It does not appear for JAX-RPC service bindings.

Procedure

1. Select the module that contains the 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 binding to view information about its properties.
      - 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 Web 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. For JAX-WS bindings only, configure policy sets for export bindings by performing the following tasks:
   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, and click **Attach** to attach a policy set to the binding, or click **Detach** to remove the policy set.
   d. To assign a policy set binding, select the export binding, click **Assign Binding**, and provide a name for the policy set binding.
   e. Repeat steps 4c and 4d for each binding you want to configure.
   f. Save the changes to the master configuration.

Results

If you made any updates, the binding is changed for the selected module. The changes take effect after you restart the SCA module.

Restriction: If the module is redeployed, the configuration changes you made are replaced by the original settings.

To ensure that the changes you made remain with the module across deployments, use WebSphere Integration Developer to make the changes in the source code for the module.

Administering HTTP bindings:

You can view information about the HTTP import and export bindings of a module after the module has been deployed to the server. You can also tune, or set, special features of the import and export bindings.

You use WebSphere Integration Developer to create HTTP imports and exports.

Viewing and updating HTTP import bindings:

Using the administrative console, you can change the configuration of HTTP import bindings without changing the original source and then redeploying the application.

About this task

Change HTTP import bindings when the binding properties of an HTTP application used by a Service Component Architecture (SCA) module change.

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.

Procedure
1. Select the module that contains the 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 binding to view information about its properties.
3. Select the scope for the changes you want to make.
   - 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. Make changes to one or more of the following properties:
   - **Select method** (Method scope only)

     Choose the method that you want to review or configure. Click on the arrow in the **Select method** field to see the list of methods that can be configured.
   - **Endpoint URL**

     Specifies the URL for the target service.
   - **HTTP Method**

     Specifies the method to use for the endpoint URL.
   - **HTTP version**

     Specifies the HTTP version to use for this endpoint URL. The choices are **1.0** and **1.1**. The default is **1.1**.
   - **Number of connection retries**

Specifies the number of times the request is retried when the system receives an error response. The default is **0**, which means that, after a failure, no attempt is made.

- **Basic HTTP authentication**

  Specifies the authentication alias to use with the HTTP server on this binding.

  To choose the authentication alias, select the alias name from the list. To change the attributes of a selected authentication alias, click **Edit**. To create a new authentication alias, click **New**.

- **SSL authentication**

  Specifies the Secure Sockets Layer (SSL) configuration to use for this binding.

  To edit an existing configuration, select the name from the list and click **Edit**. To create a new configuration, click **New**.

- **Transfer Encoding**

  Specifies how information is transferred between the endpoints. Choices are **chunked** or **identity**.

  The chunked encoding modifies the body of a message in order to transfer it as a series of chunks, each with its own size. This allows dynamically produced content to be transferred along with the information necessary for the recipient to verify that it has received the full message.

  Important: If you set this parameter to **chunked**, Content Encoding is set to **identity** and you will be unable to change Content Encoding.

- **Content Encoding**

  Specifies how the content that traverses the binding is encoded. Choose either **gzip**, **x-gzip**, **deflate**, or **identity**.

- **HTTP proxy settings** or **HTTPS proxy settings**

  Specifies the settings for bindings that do not require security authorization for access (**HTTP proxy settings**) or that do require authorization for access (**HTTPS proxy settings**).

  - **Proxy host**

    Specifies the host name or IP address of an HTTP proxy server through which to connect to the endpoint URL.

  - **Proxy port**

    Specifies the port used to connect to an HTTP proxy server for this binding.

  - **Proxy credentials**

    Specifies the Java2 Connectivity (J2C) authentication alias to use for the proxy settings.

    To change an existing alias, select the alias from the list and click **Edit**. To add a new alias, click **New**.

  - **Non-proxied hosts**

    Specifies a list of hosts on this binding that do not use proxies. Enter each host on a separate line (use the Enter key).

    To add a host to the list, type the host at the end of the list, separating it from the previous entry by clicking the Enter key. To remove a host from the list, delete the host from the list.

- **Response read timeout**

  Specifies the time, in seconds, that the binding waits to read data while receiving a response message.

Note: Setting this field to **0** causes the binding to wait indefinitely.

5. Save your changes to the master configuration.

Results

If you made any updates, the binding is changed for the selected module. The changes take effect after you restart the SCA module.

Restriction: If the module is redeployed, the configuration changes you made are replaced by the original settings.

To ensure that the changes you made remain with the module across deployments, use WebSphere Integration Developer to make the changes in the source code for the module.

Viewing and updating HTTP export bindings:

Using the administrative console, you can change the configuration of HTTP export bindings without changing the original source and then redeploying the application.

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.

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.

Procedure

1. Select the module that contains the 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 binding to view information about its properties.
3. Select the scope for the changes you want to make.
   - 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. Make changes to one or more of the following properties:
   - **Select method** (Method scope only)

     Choose the method that you want to review or configure. Click on the arrow in the **Select method** field to see the list of methods that can be configured.
   - **HTTP Methods**

     Lists the methods and the current configuration for the methods. You can set whether the method is pingable and the return code for the method.
     – **Method**

The name of the method. The methods are GET, POST, PUT, DELETE, TRACE, OPTIONS, and HEAD.

– **Pingable**

Whether or not an HTTP client can ping the method. When selected, you must specify the return code the binding returns to the client. The default for this is unchecked.

– **Return code**

An integer returned when an HTTP client pings the method.

- **Transfer Encoding**

Specifies how information is transferred between the endpoints. Choices are **chunked** or **identity**.

The chunked encoding modifies the body of a message in order to transfer it as a series of chunks, each with its own size. This allows dynamically produced content to be transferred along with the information necessary for the recipient to verify that it has received the full message.

Important: If you set this parameter to **chunked**, Content Encoding is set to **identity** and you will be unable to change Content Encoding.

- **Content Encoding**

Specifies how the content that traverses the binding is encoded. Choose either **gzip**, **x-gzip**, **deflate**, or **identity**.

5. Save your changes to the master configuration.

Results

If you made any updates, the binding is changed for the selected module. The changes take effect after you restart the SCA module.

Restriction: If the module is redeployed, the configuration changes you made are replaced by the original settings.

To ensure that the changes you made remain with the module across deployments, use WebSphere Integration Developer to make the changes in the source code for the module.

Administering EJB bindings:

You can view information about the EJB import and export bindings of a module after the module has been deployed to the server. You can also reconfigure selected properties of import bindings.

Viewing and updating EJB import bindings:

You can view information about EJB import bindings using the WebSphere administrative console. You can also modify the JNDI name associated with the binding.

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

You can modify only the **JNDI name** property. All other properties for an EJB binding import are read-only.

EJBs invoked by an EJB import can be running in any of the following combinations. For each one of these scenarios, make sure you consider the information in the JNDI configuration information column when you modify the JNDI name:

Table 4. EJB import JNDI name configurations

EJB scenario	JNDI configuration information
WebSphere Process Server in a different Java EE 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 Java EE module bindings file. **Note:** The JNDI name for local invocations, which apply only to the EJB 3.0 programming model, take the form `ejblocal:` followed by the fully qualified class name of the local interface. You can find more information in the "JNDI name" topic.
Remote WebSphere Process Server or WebSphere Application Server	Create a namespace binding (of EJB binding type) using the WebSphere Process Server administrative console. To create the namespace binding, click **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 Java EE server (other than WebSphere Process Server or WebSphere Application Server)	Create a namespace binding using the WebSphere Process Server administrative console. • If the Java EE server provides a COSNaming interface, create a namespace binding of type CORBA. • If the Java EE server does not provide a COSNaming interface, create a namespace binding of the indirect type. To create the namespace binding, click **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 might be required.

To view or configure EJB import properties using the WebSphere Process Server administrative console, complete the following steps:

Procedure

1. Select the module that contains the 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 binding to view information about its properties.
3. Change the JNDI name.
4. Save your changes to the master configuration.

Results

If you made any updates, the binding is changed for the selected module. The changes take effect after you restart the SCA module.

Restriction: If the module is redeployed, the configuration changes you made are replaced by the original settings.

To ensure that the changes you made remain with the module across deployments, use WebSphere Integration Developer to make the changes in the source code for the module.

Viewing EJB export bindings:

You can view EJB export bindings using the WebSphere administrative console.

Before you begin

To see an EJB export binding, it must be installed as part of a Service Component Architecture (SCA) application in your server profile.

Procedure

1. Select the module that contains the 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 binding to view information about its properties.

Administering EIS bindings:

EIS bindings are installed in the server as part of your SCA applications. Administer your bindings 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 or export 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. Select the module that contains the 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** or **Exports**.
   b. Expand the import or export, and then expand **Binding**.
   c. To view the WSDL, 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.
   d. To view the binding, expand **Bindings** and click the import or export binding that you want to view. You can change the port or the name of the imported or exported service
3. Optional: Change the port or the name of the imported or exported service.
4. Save your changes to the master configuration.

Results

If you made any updates, the binding is changed for the selected module. The changes take effect after you restart the SCA module.

Restriction: If the module is redeployed, the configuration changes you made are replaced by the original settings.

To ensure that the changes you made remain with the module across deployments, use WebSphere Integration Developer to make the changes in the source code for the module.

Administering JMS bindings:

You can view information about the JMS import and export bindings of a module after the module has been deployed to the server. You can also tune, or set, special features of the import and export bindings.

Use the administrative console to configure and administer JMS import and export bindings.

For detailed instructions on generating JMS imports and exports, see "Generating a JMS import binding" and "Generating a JMS export binding" in the WebSphere Integration Developer information center.

Viewing and updating 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 import or export must be installed as part of a Service Component Architecture (SCA) application in your server profile.

Procedure
1. Select the module that contains the 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** or **Exports**.
   b. Expand the import or export, and then expand **Binding**.
   c. Click the binding to view information about its properties. 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.
3. Make any desired changes to the resources:
   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 page referred to by the JNDI name.
4. Save your changes to the master configuration.

Results

If you made any updates, the binding is changed for the selected module. The changes take effect after you restart the SCA module.

Restriction: If the module is redeployed, the configuration changes you made are replaced by the original settings.

To ensure that the changes you made remain with the module across deployments, use WebSphere Integration Developer to make the changes in the source code for the module.

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.

Table 5 and Table 6 show examples of the resources you specify when you select **Use pre-configured messaging provider resources**.

Note: The JNDI name takes the form:
```
moduleName/importName_resourceAbbreviation
```

or
```
moduleName/exportName_resourceAbbreviation
```

For example, for a module named `Inventory` and an import named `Import1`, the JNDI name for the connection factory would be:
```
Inventory/Import1_CF
```

The fields and associated values for import bindings are shown in the following table:

Table 5. Example values for import bindings

Property	Example
JNDI name for connection factory	*moduleName*/*importName*_CF
JNDI name for send destination	*moduleName*/*importName*_SEND_D
JNDI name for receive destination	*moduleName*/*importName*_RECEIVE_D

The fields and associated values for export bindings are shown in the following table:

Table 6. Example values for export bindings

Property	Example
JNDI name for activation specification	*moduleName*/*exportName*_AS
JNDI name for receive destination	*moduleName*/*exportName*_RECEIVE_D
JNDI name for send destination	*moduleName*/*exportName*_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.

Viewing or changing the state of an endpoint:

You can use the administrative console to manage the state of endpoints associated with JMS bindings, WebSphere MQ JMS bindings, and WebSphere MQ bindings. For example, you can pause or resume an endpoint associated with one of those bindings.

About this task

The import or export must be installed as part of a Service Component Architecture (SCA) application in your server profile.

Note: This procedure applies only to Version 7 applications deployed to Version 7 of the runtime environment.

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** or **Exports**.
3. Expand the import or export and then expand **Binding**. Make sure that you select one of the following bindings:
   - JMS
   - WebSphere MQ JMS
   - WebSphere MQ
4. Click on the binding to be administered.
5. To view or change the state of a binding endpoint, perform the following tasks:
   a. Click the **Runtime** tab.
   b. In the **Receiving Endpoints** table, select the check box for the endpoint.
   c. Click **Pause** or **Resume** to temporarily stop or restart the endpoint.

Results

The endpoint is paused or resumed.

Administering Generic JMS bindings:

You can view information about the Generic JMS import and export bindings of a module after the module has been deployed to the server. You can also tune, or set, special features of the import and export bindings.

Use the administrative console to configure and administer Generic JMS import and export bindings.

For detailed instructions on generating Generic JMS imports and exports, see "Generating a generic JMS import binding" and "Generating a generic JMS export binding" in the WebSphere Integration Developer information center.

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.

Viewing and updating 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 module that contains the 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** or **Exports**.
   b. Expand the import or export, and then expand **Binding**.
   c. 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.
3. 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 page referred to by the JNDI name. While most resources can be configured at cluster scope, selecting the **Configure** option at Listener Port displays a page 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 page will open.
4. Save your changes to the master configuration.

Results

If you made any updates, the binding is changed for the selected module. The changes take effect after you restart the SCA module.

Restriction: If the module is redeployed, the configuration changes you made are replaced by the original settings.

To ensure that the changes you made remain with the module across deployments, use WebSphere Integration Developer to make the changes in the source code for the module.

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 use.

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 7. 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 8. 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.

Administering WebSphere MQ JMS bindings:

You can view information about the WebSphere MQ JMS import and export bindings of a module after the module has been deployed to the server. You can also tune, or set, special features of the import and export bindings.

Use the administrative console to access WebSphere MQ JMS bindings.

For detailed instructions on generating WebSphere MQ imports and exports, see "Generating an MQ JMS import binding" and "Generating an MQ JMS export binding" in the WebSphere Integration Developer information center.

Viewing and updating 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 module that contains the 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** or **Exports**.
   b. Expand the import or export, and then expand **Binding**.
   c. Click the binding to view information about its properties. 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.
3. Make any desired changes to the resources:

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 page referred to by the JNDI name.

   Note: Most resources can be configured at the cluster scope. However, when you select the **Configure** option for the activation specification, a page is displayed that shows all activation specifications for all members of the given cluster; you can then select one activation specification.

4. Save your changes to the master configuration.

Results

If you made any updates, the binding is changed for the selected module. The changes take effect after you restart the SCA module.

Restriction: If the module is redeployed, the configuration changes you made are replaced by the original settings.

To ensure that the changes you made remain with the module across deployments, use WebSphere Integration Developer to make the changes in the source code for the module.

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 use.

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 9. 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 Activation Specification	mqjms.module	my/import	mqjms.module/my/import_AS

Table 9. MQ JMS imports: Names and JNDI names of resources created at installation on the server (continued)

Resource	Module name	Import name	Resource global JNDI name
Failed Event Replay 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 10. MQ JMS exports: Names and JNDI names of resources created at installation on the server

Resource	Module name	Export name	Resource global JNDI name
Request Activation Specification	mqjms.module	my/export	mqjms.module/my/export_AS
Failed Event Replay 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.

Viewing or changing the state of an endpoint:

You can use the administrative console to manage the state of endpoints associated with JMS bindings, WebSphere MQ JMS bindings, and WebSphere MQ bindings. For example, you can pause or resume an endpoint associated with one of those bindings.

About this task

The import or export must be installed as part of a Service Component Architecture (SCA) application in your server profile.

Note: This procedure applies only to Version 7 applications deployed to Version 7 of the runtime environment.

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** or **Exports**.
3. Expand the import or export and then expand **Binding**. Make sure that you select one of the following bindings:
   - JMS
   - WebSphere MQ JMS
   - WebSphere MQ
4. Click on the binding to be administered.
5. To view or change the state of a binding endpoint, perform the following tasks:
   a. Click the **Runtime** tab.
   b. In the **Receiving Endpoints** table, select the check box for the endpoint.
   c. Click **Pause** or **Resume** to temporarily stop or restart the endpoint.

Results

The endpoint is paused or resumed.

Administering WebSphere MQ bindings:

You can view information about the WebSphere MQ import and export bindings of a module after the module has been deployed to the server. You can also tune, or set, special features of the import and export bindings.

Use the administrative console to access WebSphere MQ bindings.

For detailed instructions on generating WebSphere MQ imports and exports, see "Generating an MQ JMS import binding" and "Generating an MQ JMS export binding" in the WebSphere Integration Developer information center.

Viewing and updating 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 or export must be installed as part of a Service Component Architecture (SCA) application in your server profile.

Procedure

1. Open the default messaging provider settings page 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 page 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 page 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 JCA pages where you can find these properties.

3. Optional: Administer WebSphere MQ queue connection factories.

   Click **WebSphere MQ queue connection factories** in the list of additional properties. This page 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 page 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 page 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.

   New WebSphere MQ queue destinations must be configured with the custom properties in the following table.

Table 11. Custom properties for WebSphere MQ queue destinations

Destination Type	Property Name	Property Value	Property Type
Send destination	MDWRITE	YES	java.lang.String
	MSGBODY	MQ	java.lang.String
Receive destination	MDREAD	YES	java.lang.String
	MSGBODY	MQ	java.lang.String

Use the subsequent page 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. Select the module that contains the binding by navigating to **Applications** → **SCA Modules** and clicking the module name.

6. Select the binding by performing the following steps:

   a. In the **Module components** section, expand **Imports** or **Exports**.

   b. Expand the import or export, and then expand **Binding**.

   c. Click the binding to view information about its properties. 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.

7. Make any desired changes to the resources:

   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 page referred to by the JNDI name.

      Note: Most resources can be configured at the cluster scope. However, when you select the **Configure** option for the activation specification, a page is displayed that shows all activation specifications for all members of the given cluster; you can then select one activation specification.

8. Save your changes to the master configuration.

Results

If you made any updates, the binding is changed for the selected module. The changes take effect after you restart the SCA module.

Restriction: If the module is redeployed, the configuration changes you made are replaced by the original settings.

To ensure that the changes you made remain with the module across deployments, use WebSphere Integration Developer to make the changes in the source code for the module.

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 use.

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 12. 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 Activation Specification	mq.module	my/import	mq.module/my/import_AS
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 13. WebSphere MQ export: Names and JNDI names of resources created at installation on the server

Resource	Module name	Export name	Resource global JNDI name
Request Activation Specification	mq.module	my/export	mq.module/my/export_AS
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

Table 13. WebSphere MQ export: Names and JNDI names of resources created at installation on the server (continued)

Resource	Module name	Export name	Resource global JNDI name
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.

Viewing or changing the state of an endpoint:

You can use the administrative console to manage the state of endpoints associated with JMS bindings, WebSphere MQ JMS bindings, and WebSphere MQ bindings. For example, you can pause or resume an endpoint associated with one of those bindings.

About this task

The import or export must be installed as part of a Service Component Architecture (SCA) application in your server profile.

Note: This procedure applies only to Version 7 applications deployed to Version 7 of the runtime environment.

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** or **Exports**.
3. Expand the import or export and then expand **Binding**. Make sure that you select one of the following bindings:
   - JMS
   - WebSphere MQ JMS
   - WebSphere MQ
4. Click on the binding to be administered.
5. To view or change the state of a binding endpoint, perform the following tasks:
   a. Click the **Runtime** tab.
   b. In the **Receiving Endpoints** table, select the check box for the endpoint.
   c. Click **Pause** or **Resume** to temporarily stop or restart the endpoint.

Results

The endpoint is paused or resumed.

Migrating WebSphere MQ Bindings from version 6 to version 7:

Migration is only required for WebSphere MQ bindings that contain pre-configured resources.

Specifying an activation specification

In WebSphere ESB version 7, the WebSphere MQ binding uses the WebSphere MQ resource adapter to receive messages, which requires an activation specification. If a WebSphere MQ binding has pre-configured WebSphere MQ resources, you need to define an additional activation specification JNDI name in the end-point configuration of the binding. This JNDI name must refer to an existing activation specification JMS resource on the server.

Modifying connection factory properties

Pre-configured connection factories need these custom properties removed:
- SENDEXIT
- RECEXIT
- SENDEXITINIT
- RECEXITINIT

Modifying destination properties

Pre-configured destinations need these custom properties added:

Table 14. Custom properties for WebSphere MQ queue destinations

Destination Type	Property Name	Property Value	Property Type
Send destination	MDWRITE	YES	java.lang.String
	MSGBODY	MQ	java.lang.String
Receive destination	MDREAD	YES	java.lang.String
	MSGBODY	MQ	java.lang.String

Administering enterprise applications

Use the console's Enterprise application page to view and administer enterprise applications installed on the server.

To view the values specified for an application's configuration, open the page (click **Applications > Application Types> WebSphere enterprise applications**) and select the application name from the list. The application details page opens and displays the application's configuration properties and, if appropriate, local topology. From this page, you can modify existing values and link to additional console pages for configuring the application.

To administer an enterprise application, select it by clicking the check box next to its name and then use one of the following buttons:

Table 15. Buttons for administering enterprise applications

Button	Resulting action
Start	Attempts to run the application. After the application starts successfully, the state of the application changes to one of the following: • `Started`: The application has started on all deployment targets • `Partial Start`: The application is still starting on one or more of the deployment targets
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 .sar or .rar file) onto a server or cluster.
Uninstall	Deletes the application from the product 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 and synchronized with the nodes.
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 relative path 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.

Table 15. Buttons for administering enterprise applications (continued)

Button	Resulting action
Rollout Update	Sequentially updates an application installed on multiple cluster members across a cluster. After you update an application's files or configuration, click **Rollout Update** to install the application's updated files or configuration on all cluster members of a cluster on which the application is installed. **Rollout Update** does the following for each cluster member in sequence: 1. Saves the updated application configuration. 2. Stops all of the cluster members on one node. 3. Updates the application on the node by synchronizing the configuration. 4. Restarts the stopped cluster members. 5. Repeats steps 2 through 4 for all of the nodes that have cluster members. Use **Rollout Update** if the application is deployed on one or more clusters spread across multiple nodes. This action reduces the amount of time that any single cluster member is unavailable to serve requests to the smallest interval possible. Pending IIOP transactions will complete before a cluster member stops; in-flight HTTP and JMS transactions might be lost while the cluster member is stopping. For an application server without clusters, use **Update** and then save and synchronize the node instead. For a standalone application server, simply update and save.
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. If the application or module is deployed on a cluster, after removing a file click **Rollout Update** to roll out the changes across the entire cluster.
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.
Export File	Accesses the Export a file from an application page, which you use to export a file of an enterprise application or module to a location of your choice. If the browser does not prompt for a location to store the file, click **File** ▸ **Save as** and specify a location to save the file that is shown in the browser.

For more information, see Deploying and administering enterprise applications in the WebSphere Application Server Information Center.

Administering the throughput of SCA requests

For each Service Component Architecture (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.

Procedure

- 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**

- 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 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.

Administering applications and application services **111**

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 in the administrative console" on page 76. 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.

Note: JNDI resources must not be shared across clusters. A cross-cluster module requires that each cluster have different JNDI resources. For more information, see Considerations for installing service applications on clusters.

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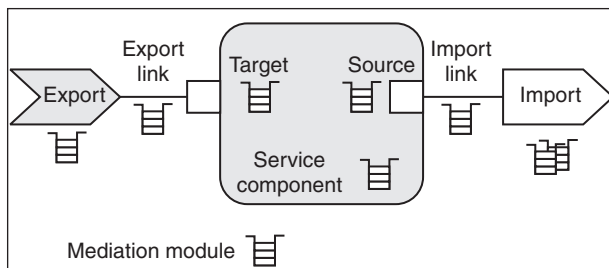
For each of the destinations, a queue point is also created, and defined on the messaging engine of the relevant bus member.

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.

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.
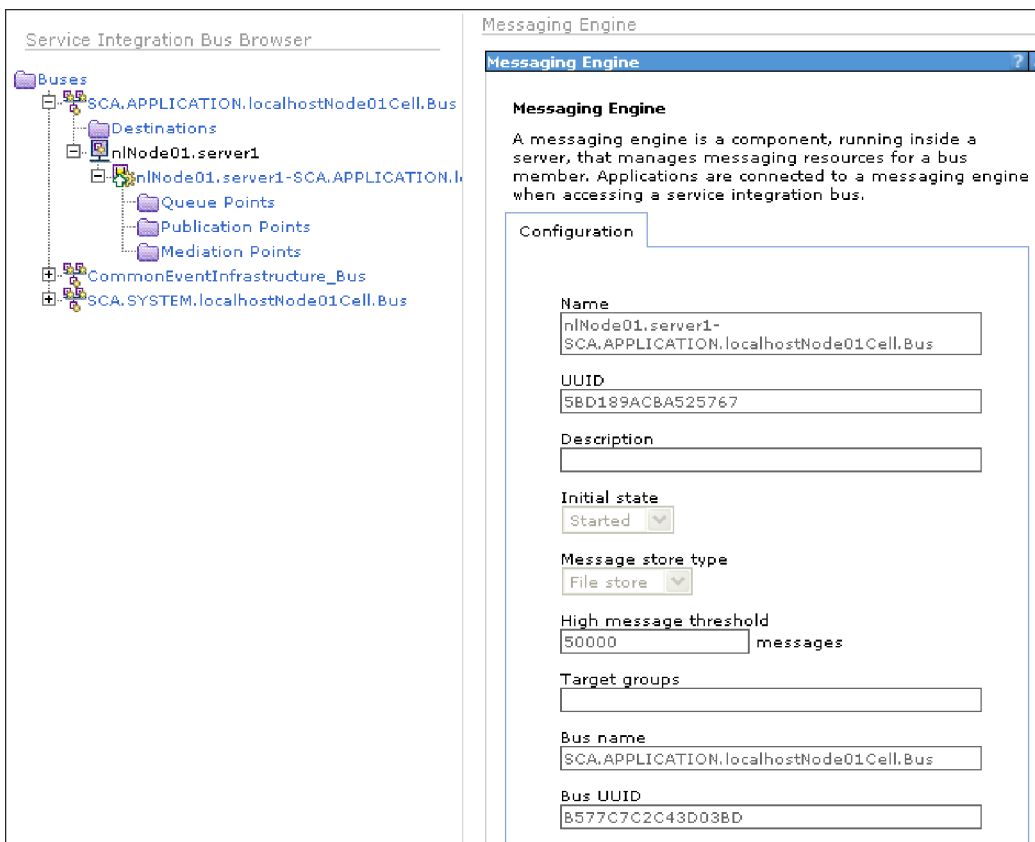
Table 16 lists and describes the icons associated with each item in the navigation tree.

Table 16. 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.

Table 16. Icons in the service integration bus browser (continued)

Icon	Description
	Indicates a messaging engine.
	Indicates a service integration bus member.

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**.

Type of invocation	How to change
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.

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 that change the details of import and export bindings.

Administering service modules using commands:

You can use a command to list the service modules that have been deployed to WebSphere Process Server. You can also view information about a service module and make changes to properties associated with the service module.

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.

About this task

Use the wsadmin tool to list service modules.

Note: The following procedure uses Jython syntax. See the individual command descriptions for information about using Jacl.

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 format makes it easier for scripts to parse the output and extract names for use in subsequent commands.

Displaying the properties of a service module using commands:

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

About this task

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

Note that you can use these commands individually, or you can use commands in combination. For example, you can parse the information returned by the first command (such as the module name) to determine the value to use in the second command.

Note: The following procedure uses Jython syntax. See the individual command descriptions for information about using Jacl.

Procedure
1. Optional: List the 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 and other optional parameters. Providing an application name improves performance.

Results

The properties of the specified SCA module are displayed.

The output of the listSCAModules command is in this format:
```
myModule:myModuleApp: :myModule: :\nyourModule_v1_0_0_yourCellId:
yourModule_v1_0_0_yourCellIdApp:6.0.0:yourModule:yourCellId:
```

The output of the showSCAModuleProperties command is in this format:
```
[myGroup]myProperty1=myValue1\n[myGroup]myProperty2=myValue2
```

Changing a service module property using commands:

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

About this task

Use the wsadmin tool to change property values. To change the properties, you need to know the module name.

Note that you can use these commands individually, or you can use commands in combination. For example, you can parse the information returned by the first command (such as the module name) to determine the value to use in the second command.

Note: The following procedure uses Jython syntax. See the individual command descriptions for information about using Jacl.

Procedure

1. Optional: List the deployed SCA modules by using the following wsadmin scripting command: .

   ```
   AdminTask.listSCAModules()
   ```

2. Optional: List the properties for a particular SCA module by using the following wsadmin scripting command:

   ```
   AdminTask.showSCAModuleProperties('-moduleName moduleName')
   ```

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

3. Modify a module property for the SCA module by using the following wsadmin scripting command:

   ```
   AdminTask.modifySCAModuleProperty('-moduleName moduleName
   -propertyName propertyName -newPropertyValue newpropertyValue')
   ```

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

Results

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

Working with imports:

You can use a command to list the imports of any service module deployed to WebSphere Process Server. You can also use commands to list the details of an import or the bindings associated with the import and to change the settings of import bindings.

Displaying imports and associated details using a command:

You can use a wsadmin scripting command to list the imports associated with a service module deployed to WebSphere Process Server. You can also display details of a particular import.

About this task

Use the wsadmin tool to list the imports associated with a service module and to display the details of a service module import.

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

Note that you can use these commands individually, or you can use commands in combination. For example, you can parse the information returned by the first command (such as the module name) to determine the value to use in the second command.

Note: The following procedure uses Jython syntax. See the individual command descriptions for information about using Jacl.

Procedure

1. Optional: List the deployed SCA modules by using the following wsadmin scripting command:

   ```
   AdminTask.listSCAModules()
   ```

2. List the imports for a particular SCA module by using the following wsadmin scripting command:

   ```
   AdminTask.listSCAImports('-moduleName moduleName')
   ```

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

3. Show the details of a particular SCA module import by using the following wsadmin scripting command:

   ```
   AdminTask.showSCAImport('-moduleName moduleName
   -import importName')
   ```

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

Results

Displays the imports associated with a module and the details for a particular SCA module import.

Displaying an import binding using a command:

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

About this task

Use the wsadmin tool to display the bindings of a particular service module import.

Note: It is possible for an SCA module not to have any imports.

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

You can also display information about a specific type of import binding, such as Web service or JMS.

Note that you can use these commands individually, or you can use commands in combination. For example, you can parse the information returned by the first command (such as the module name) to determine the value to use in the second command.

Note: The following procedure uses Jython syntax. See the individual command descriptions for information about using Jacl.

Procedure

1. Optional: List the deployed SCA modules by using the following wsadmin scripting command:

   ```
   AdminTask.listSCAModules()
   ```

2. Optional: List the imports for a particular SCA module by using the following wsadmin scripting command:

   ```
   AdminTask.listSCAImports('[-moduleName moduleName]')
   ```

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

3. Show the import binding for a particular import by using the following wsadmin scripting command:

   ```
   AdminTask.showSCAImportBinding('[-moduleName moduleName
   -import importName]')
   ```

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

Results

Displays the import binding for a particular SCA module import.

The output of the showSCAImportBinding command depends upon the type of binding. For example, for an adapter (EIS) import binding, the output would be in the following format:

```
importBinding:type=AdapterImportBinding
```

Changing an import binding using commands:

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

About this task

Use the wsadmin tool to change import binding properties associated with a service module.

Note: It is possible for an SCA module not to have any imports.

To change the properties, you need to know the module name.

Note that you can use these commands individually, or you can use commands in combination. For example, you can parse the information returned by the first command (such as the module name) to determine the value to use in the second command.

Note: The following procedure uses Jython syntax. See the individual command descriptions for information about using Jacl.

Procedure

1. Optional: List the deployed SCA modules by using the following wsadmin scripting command:

   ```
   AdminTask.listSCAModules()
   ```

2. Optional: List the imports for a particular SCA module by using the following wsadmin scripting command:

   ```
   AdminTask.listSCAImports('[-moduleName moduleName]')
   ```

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

3. Modify an import binding using the modifySCAimport*bindingType*Binding command, where *bindingType* is the actual type of binding, as in the following list:
   - modifySCAImportEJBBinding
   - modifySCAImportHttpBinding
   - modifySCAImportJMSBinding

     Note: The modifySCAImportJMSBinding command is used for the JMS binding, the generic JMS binding, and the WebSphere MQ JMS binding.
   - modifySCAImportMQBinding
   - modifySCAImportSCABinding
   - modifySCAImportWSBinding

   Each of these commands has its own set of parameters. For example, the modifySCAImportSCABinding has the following parameters:

```
AdminTask.modifySCAImport('[-moduleName moduleName
-import importName -targetModule targetModuleName
-targetExport  targetExportName]')
```

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

Results

Changes the import binding for a particular import.

When using the modifySCAImportSCABinding command to change the SCA
export referred to by an SCA import binding, WebSphere Process Server 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 the import and
export. If they are not the same, a warning is issued. However, if the port type
names do match, WebSphere Process Server assumes that the operations provided
are equivalent and no warning is issued.

Example

To change an EJB import binding:
```
AdminTask.modifySCAImportEJBBinding('[-moduleName myModule
-import myImport -jndiName newjndiName
-applicationName myApplication]')
```

To change a Web service import binding:
```
AdminTask.modifySCAImportWSBinding('[-moduleName myModule
-applicationName myApplication -import myImport
-endpoint http://myTargetEndpoint]')
```

Working with exports:

You can use a command to list the exports of any service module deployed to
WebSphere Process Server. You can also use commands to list the details of an
export or the bindings associated with the export and to change the settings of
HTTP, JMS, and WebSphere MQ export bindings.

Displaying exports and associated details using a command:

You can use a wsadmin scripting command to list the exports associated with a
service module deployed to WebSphere Process Server. You can also display details
of a particular export.

About this task

Use the wsadmin tool to list the exports associated with a service module and to
display the details of a service module export.

Note: It is possible for an SCA module not to have any exports.

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

Note that you can use these commands individually, or you can use commands in combination. For example, you can parse the information returned by the first command (such as the module name) to determine the value to use in the second command.

Note: The following procedure uses Jython syntax. See the individual command descriptions for information about using Jacl.

Procedure

1. Optional: List the deployed SCA modules by using the following wsadmin scripting command:

   ```
   AdminTask.listSCAModules()
   ```

2. List the exports for a particular SCA module by using the following wsadmin scripting command:

   ```
   AdminTask.listSCAExports('[-moduleName moduleName]')
   ```

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

3. Show the details of a particular SCA module export by using the following wsadmin scripting command:

   ```
    AdminTask.showSCAExport('[-moduleName moduleName
   -export exportName]')
   ```

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

Results

Displays the exports associated with a module and the details for a particular SCA module export.

The output of the showSCAExport command is in this format:

```
export:name=exportName,description=null
interface:type=type,portType=portType
```

Displaying an export binding using a command:

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

About this task

Use the wsadmin tool to display the bindings of a service module export.

Note: It is possible for an SCA module not to have any exports.

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

You can also display information about a specific type of export binding, such as Web service or JMS.

Note that you can use these commands individually, or you can use commands in combination. For example, you can parse the information returned by the first command (such as the module name) to determine the value to use in the second command.

Note: The following procedure uses Jython syntax. See the individual command descriptions for information about using Jacl.

Procedure

1. Optional: List the deployed SCA modules by using the following wsadmin scripting command:

   `AdminTask.listSCAModules()`

2. Optional: List the exports for a particular SCA module by using the following wsadmin scripting command:

   `AdminTask.listSCAExports('[-moduleName moduleName]')`

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

3. Show the export binding for a particular export by using the following wsadmin scripting command:

   `AdminTask.showSCAExportBinding('[-moduleName moduleName -export exportName]')`

   Note: In addition to specifying the module name, you have the option of specifying the application name and other optional parameters. Providing an application name 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, the run time assumes that the binding is of type SCA.

The output of the showSCAExportBinding command depends upon the type of binding. For example, for Web service export bindings, the output is in the following format:

`exportBinding:type=WebServiceExportBinding,port=_:portType,service=_:serviceName`

Changing an export binding using commands:

You can use a wsadmin scripting command to change the export bindings of service modules deployed to WebSphere Process Server. You can change the properties of HTTP, JMS, and WebSphere MQ export bindings.

About this task

Use the wsadmin tool to change export binding properties associated with a service module.

Note: It is possible for an SCA module not to have any exports.

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

Note that you can use these commands individually, or you can use commands in combination. For example, you can parse the information returned by the first command (such as the module name) to determine the value to use in the second command.

You can change the following types of export bindings:
- HTTP
- JMS (which applies to generic JMS and MQ JMS as well)
- MQ

Note: The following procedure uses Jython syntax. See the individual command descriptions for information about using Jacl.

Procedure

1. Optional: List the deployed SCA modules by using the following wsadmin scripting command:

   ```
   AdminTask.listSCAModules()
   ```

2. Optional: List the exports for a particular SCA module by using the following wsadmin scripting command:

   ```
   AdminTask.listSCAExports('[-moduleName moduleName]')
   ```

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

3. Modify an export binding using the modifySCAexport*bindingType*Binding command, where *bindingType* is the actual type of binding, as in the following list:
   - modifySCAExportHttpBinding
   - modifySCAExportJMSBinding
   - modifySCAExportMQBinding

   Each of these commands has its own set of parameters. For example, the modifySCAExportJMSBinding has the following parameters:

   ```
   AdminTask.modifySCAExportJMSBinding
   ('[-moduleName moduleName -export exportName
   -type JMS -sendDestination  sendDestinationName]')
   ```

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

Results

Changes the export binding for a particular export.

Example

To change an MQ export binding:

```
AdminTask.modifySCAExportMQBinding('[-moduleName myModule
-export exportName -sendDestination sendDestinationName]')
```

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" on page 117
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 JCA activation specifications

The system builds JCA 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 JCA 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 JCA 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 JCA 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 JCA activation specifications related to the module that created these destinations.

Administering enterprise applications

Use the console's Enterprise application page to view and administer enterprise applications installed on the server.

To view the values specified for an application's configuration, open the page (click **Applications > Application Types> WebSphere enterprise applications**) and select 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 17. 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 .sar or .rar file) onto a server or cluster.
Uninstall	Deletes the application from the product 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 and synchronized with the nodes.
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 relative path 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.

Table 17. Buttons for administering enterprise applications (continued)

Button	Resulting action
Rollout Update	Sequentially updates an application installed on multiple cluster members across a cluster. After you update an application's files or configuration, click **Rollout Update** to install the application's updated files or configuration on all cluster members of a cluster on which the application is installed. **Rollout Update** does the following for each cluster member in sequence: 1. Saves the updated application configuration. 2. Stops all of the cluster members on one node. 3. Updates the application on the node by synchronizing the configuration. 4. Restarts the stopped cluster members. 5. Repeats steps 2 through 4 for all of the nodes that have cluster members. Use **Rollout Update** if the application is deployed on one or more clusters spread across multiple nodes. This action reduces the amount of time that any single cluster member is unavailable to serve requests to the smallest interval possible. Pending IIOP transactions will complete before a cluster member stops; in-flight HTTP and JMS transactions might be lost while the cluster member is stopping. For an application server without clusters, use **Update** and then save and synchronize the node instead. For a standalone application server, simply update and save.
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. If the application or module is deployed on a cluster, after removing a file click **Rollout Update** to roll out the changes across the entire cluster.
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.
Export File	Accesses the Export a file from an application page, which you use to export a file of an enterprise application or module to a location of your choice. If the browser does not prompt for a location to store the file, click **File** ⇒ **Save as** and specify a location to save the file that is shown in the browser.

For more information, see Deploying and administering enterprise applications in the WebSphere Application Server Information Center.

Removing SCA destinations

By default, when you uninstall a module, the server deletes all the Service Component Architecture (SCA) destinations that are no longer active. If you have changed the default processing for SCA destinations and you uninstall a module, you must manually remove any inactive SCA destinations (those that are no longer used by any currently deployed module).

Before you begin

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

Procedure

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

Results

The destinations are removed from the server.

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.

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.

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** ⟶ **Server Types** ⟶ **WebSphere application servers** ⟶ *server name*.
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.
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
- Export an existing static relationship instance (from one platform) to an RI or CSV file, then use the relationship manager to import the RI or CSV file into a new environment (a different platform from the first)
- 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

You can 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.

About this task

To view the list of relationships in the system, do the following:

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

You can 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.

About this task

To view detailed information for the selected relationship, do the following:

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

You can 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.

About this task

To view detailed information for the selected role, do the following:

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 make 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.

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, do the following:

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.

Option	Description
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 using 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 18. Relationship database view columns

Name	Data type	Value	Nullable?
INSTANCEID	Integer	The ID number used to correlate instance data between different applications.	No
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 are 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 19. Clarify customer

Given Name	Last Name	Home Phone	ID
Jessica	Reed	111 111 11111	clarify_1
Tara	McLean	333 333 33333	clarify_2

Table 20. SAP customer

First Name	Last Name	Home Phone	ID
Jessica	Reed	111 111 11111	sap_10
Tara	McLean	333 333 33333	sap_8

Table 21. 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 22. 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 23. 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 24. 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 25. 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 26. 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" on page 140
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

You can 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.

About this task

To view a list of relationship instances that match the relationship query, do the following:

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 143.

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

You can 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.

About this task

To view detailed information for the selected relationship instance, do the following:

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

You can 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.

About this task

To create a new relationship instance, do the following:

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

You can 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.

About this task

To delete a selected relationship instance, do the following:

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

You can 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.

About this task

The following actions take place 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, do the following:

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.

Importing relationships

Perform this task to import data from an existing static relationship into your system. Importing an existing relationship is useful in situations when you want to incorporate a relationship from another platform into your solution, but do not want to write code or use relationship manager to add instance details one by one.

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.

The relationship manager can import static relationships that have been exported in RI or CSV file format. For more information on exporting a relationship instance so it can be used on another platform, see "Exporting relationships."

About this task

Only static relationships are supported. If there are existing relationship instances in the database, existing relationship instances and newly imported relationship instances will be merged. If the relationship definition that you are importing does not exist, a RelationshipUserException will be thrown.

To import an existing 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. Click **Import**.
5. **Browse** for the import file name.
6. Select the radio button next to the correct import file format - either **RI** or **CSV**.
7. Click **OK**.

Results

The relationship instance is imported into the system.

Exporting relationships

Perform this task to export data from an existing static relationship to an RI or CSV file. Exporting a relationship is useful in situations when you want to incorporate an existing relationship from one platform into a system running on another platform, but do not want to write code or use relationship manager to add instance details one by one.

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

Only static relationships are supported.

To export an existing 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 **Export**.
5. Type an export file name. By default, the export file name consists of the relationship name and the extension name (based on your format selection).
6. Select the radio button next to the correct import file format - either **RI** or **CSV** and click **OK**.
7. Select **Save file** and click **OK**.

Results

The relationship instance is exported into an RI or CSV file.

What to do next

Import the RI or CSV file into your solution. For more information on importing a relationship instance so it can be used on another platform, see "Importing relationships" on page 153.

Viewing role instance details

You can 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.

About this task

To view detailed information for the selected role instance, do the following:

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

You can 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.

About this task

To edit the property values for the selected role instance, do the following:

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

You can 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.

About this task

To create a new role instance for a relationship, do the following:

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

You can 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.

About this task

To delete a selected role instance of a relationship, do the following:

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, 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® or DB2 for IBM i, 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 for i5/OS or DB2 for IBM i, 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`
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 Java EE 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.Samples of integration application artifacts are available in the Samples Gallery.

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**, enter901
   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 Java EE 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.

Relationship definitions that are contained in user-defined shared libraries will be created in the server at startup time, even if those shared libraries are not associated with any server or application. You can use the relationship manager to manage these relationships.

Removing relationship database artifacts

When you install an application that uses relationships, the relationship service creates many database artifacts, including a set of database artifacts for each relationship and relationship role definition. In a unit test environment (UTE), the relationship database artifacts are removed when you uninstall the application. In a non-UTE environment, the relationship database artifacts are not removed, but are kept unchanged. The next time the application is installed, any new static relationship data in the application is not populated, and if the relationship definition is changed, the relationship table is not recreated. This can result in errors.

To force the relationship database artifacts to be removed, use the RelationshipDatabaseSchemaDrop script or the dropRelationshipDatabaseSchema API before you uninstall the application.

Note: Neither of these interfaces will detect when a relationship is shared by other applications.

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 with 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 is set to the old version. Otherwise, it is set to the current version.
5. To return to the Relationship collection page, click **Back**.

RelationshipDatabaseSchemaDrop script

Use the RelationshipDatabaseSchemaDrop.py script to remove all relationship database artifacts that are associated with a relationship, including all database artifacts and instance data that are associated with the roles defined for that relationship.

The RelationshipDatabaseSchemaDrop.py script is located in the `WPS_HOME/util/RelService` directory.

Make sure that wsadmin is connected to the server (in a stand-alone environment) or to the deployment manager (in a Network Deployment environment).

Note: The RelationshipDatabaseSchemaDrop.py script causes all the relationship database artifacts that are associated with the relationship to be forcibly removed, even if other applications are using the relationship. The system does not detect shared relationships.

Note: Do not uninstall the application that contains the relationship whose database artifacts you plan to drop before you run the RelationshipDatabaseSchemaDrop.py script. Uninstall the application after you run the script.

Required parameter

relationshipFullName
> The full name of the relationship, which has the following format: `relationship target namespace` plus "/" plus `relationship short name`. For example, with relationship target namespace `http://RelationshipSample` and relationship short name `CountryRelationship`, the full name would be `http://RelationshipSample/CountryRelationship`.

Example

This example removes all the relationship database artifacts that are associated with relationship full name `http://RelationshipSample/CountryRelationship`.

```
wsadmin -f ${WPS_HOME}/util/RelService/RelServicRelationshipDatabaseSchemaDrop.py http://RelationshipSample/CountryRelationship
```

Administering Business Process Choreographer

For information on how to administer Business Process Choreographer, go to the WebSphere Process Server for z/OS 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.

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 z/OS, 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.

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.

Considerations for database configuration

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

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

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

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

Removing business rule and selector data from the repository

When you uninstall an application that uses business rules or selectors, the server does not remove these artifacts from the repository. 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 make up 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 > Server Types > WebSphere 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 > Server Types > WebSphere 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 > Server Types > WebSphere 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.
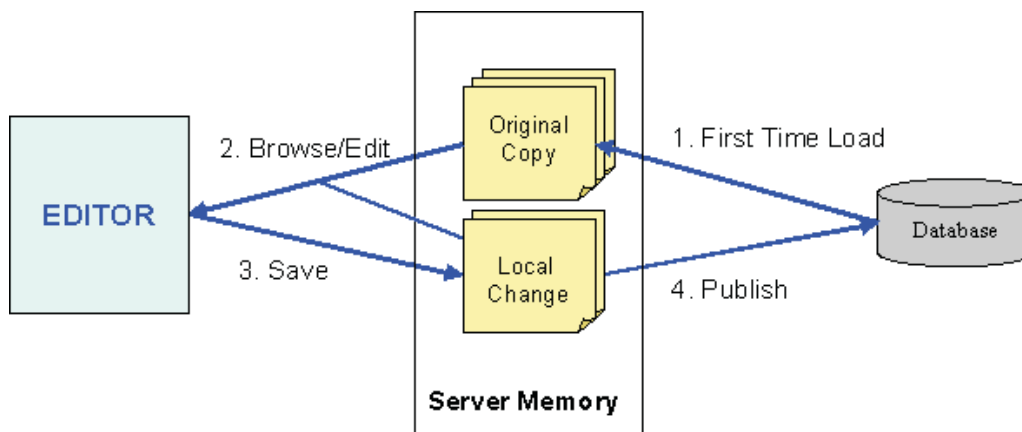
Figure 3. Business rules manager sequence of events

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 27. 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.

Table 27. Function buttons (continued)

Button Name	Function
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.
is like	Indicates that the query should look for business rule groups where the value of a business rule group name, target name space, or property is like the specified string. The string can contain wildcard characters. Use the percent character ('%') to specify a wildcard for any number of characters and use the underscore character ('_') to specify a single character wildcard. These wildcard characters must follow SQL syntax.
is not equal to	Indicates that the value of the business rule group name, target name space, or property must not match the specified string.
is not like	Indicates that the query should look for business rule groups where the value of a business rule group name, target name space, or property is not like the specified string. The string can contain wildcard characters as defined in the "like" operator.

Value Provides a text area for entering the property value. The value can be empty and is taken into the Search context.

> **Example:** If the value of property PayMethod is left empty and its query operator is set to "is not equal to," the Search will find all the business rule groups whose PayMethod property has the value set to a non-empty string.

Action
> Enables moving a property up or down inside the property table and deleting custom properties.

Search Results section

This section contains the following elements:

Rule Groups
> Lists the names of the business rule groups that the search query returned.

Status Shows the status of the business rule group returned from the runtime as a search result. The status can be one of the following four kinds of status.

Tip: Clicking on a result business rule group opens its business rule group page.

Status	Description
Same as Local	Indicates that a copy of the result business rule group already exists in the business rule manager and that its content and the content of the result business rule group are exactly the same. Thus, no further action is taken after the search.
Modified from Runtime	Indicates that a copy of the result business rule group already exists in the business rules manager. However, another user session modified the master copy, and so the contents of the local and result business rule groups are different. The business rules manager will automatically update the local copy to get new modifications from the runtime.
Modified in Local	Indicates that a copy of the result business rule group already exists in the business rules manager. However, it has been modified by the current user. The business rules manager will use the local copy for any further actions by the user.
New to Local	Indicates that a copy of the result business rule group does not exist in the business rules manager. In this case, the business rules manager will create a local copy of the result business rule group and will also display it in the navigation pane.

Description

Provides additional information for the business rule group.

Adding, deleting, and modifying business rule group properties

You can use custom properties on business rule groups for searches in order to retrieve subsets of business rule groups that you want to view and modify. You add new custom properties, delete or modify existing properties through the editing pages of business rule groups. The number of custom properties on a business rule group is unlimited.

Before you begin

You need to be in the edit mode for the business rule group.

Restriction: Properties support on business rule groups is available on 6.1 business rule groups and later.

About this task

To add, delete, or modify business rule group properties, perform the following steps.

Procedure

1. Select from the following options.

Option	Description
Option	Steps
Add a property to the rule	1. Click **Add Property**. 2. Specify a unique **Name**. The name cannot be empty. 3. Specify a unique **Value**. Each property can only be defined once in a business rule group and must have a defined value. The value can be an empty string or zero in length, but not null. Setting a property to null is the same as deleting the property.
Delete a property	In the **Action** field of the selected property, click **Delete**.
Modify a property	Enter the new name and value in the corresponding field.
Sort properties	Click **Sort** to sort the properties on the business rule groups by the property names in alphabetical ascending order.

2. Click **Save**.

Results

The business rules manager will validate the rules before sending the properties to the server.

Searching business rule groups

You can perform a search query on a business rule group to locate or narrow a specified set of business rule groups that you want to work with. You create a search query based on the name, target name space, custom properties, or any combination of these.

Before you begin

You need to be on the Search for Business Rule Groups page, which you can open by clicking **Search** in the business rules manager toolbar.

About this task

To create a search query, perform the following steps.

Procedure

1. In the **Name** field enter the name of the business rule group to search for. You can leave this value empty; however, it will not be included in the search context. The value you enter is used as both a name and a display name. Consequently, the search will look for business rule groups with either the names or the display names that match the entered name value. If you want to specifically search by either name or display name, but not both, you need to indicate such a search through property names.

Example: If you enter IBMSystemName for the name of a property and VIPGroup for the value of the property, the business rules manager will search for business rule groups with the names, but not display names, matching VIPGroup.

2. In the **Target Namespace** field enter the URL of the business rule group. You can leave this value empty; however, it will not be included in the search context.

3. For each **Search Data** field select one of the following four query operators.

Option	Description
Query Operator	Description
is equal to	Indicates that the value of a business rule group name, target name space, or property must match the specified string exactly.
is like	Indicates that the query should look for business rule groups where the value of a business rule group name, target name space, or property is like the specified string. The string can contain wildcard characters. Use the percent character ('%') to specify a wildcard for any number of characters and use the underscore character ('_') to specify a single character wildcard. These wildcard characters must follow SQL syntax. **Examples:** 1. If you enter "is like" "Discount" for the business rule group name and "http://calculateDiscounts" as the target name space, the search will return all the business rule groups containing that string and with that URL. 2. If you enter "is like" "%Discount%" for the business rule group name, the search will return all the business rule groups with names such as AirlineTicketDiscount and MovieTicketDiscountRules.
is not equal to	Indicates that the value of the business rule group name, target name space, or property must not match the specified string.
is not like	Indicates that the query should look for business rule groups where the value of a business rule group name, target name space, or property is not like the specified string. The string can contain wildcard characters as defined in the "like" operator.

4. **Optional:** Click **Add Property** to add as many properties as needed for the search context.

   a. Specify the **Name**. It must be unique inside the Properties table of the search context and must not be empty.

   b. Specify the **Query Operator**.

   c. Specify the **Value**. The value can be empty and is taken into the search context.

Example: If the value of property PayMethod is left empty and its query operator is set to "is not equal to," the search will find all the business rule groups whose PayMethod property has the value set to a non-empty string.

   d. Click the up and down arrows in the **Action** field to order the properties.

Tip: You can combine the properties in the **Logical Operator** field using "And", "Or", or "Not" to create a search query containing multiple properties.

Example: To search for all the business rule groups in target namespace "http://calculateDiscounts" and with the DiscountedItem property containing string "men T-Shirts" and with the Ship Handling property set to value "Free", you would use the logical property "And".

Note: Adding, deleting, or modifying the properties on the Search for Business Rule Groups page only applies within the search context. It does not affect the properties of any rule object inside the business rules manager.

5. Click **Search**.

Results

The business rule groups that match the search query display in the **Search Results** section on the Search for Business Rule Groups page. The status of the business rule group returned from the runtime as a search result may be one of the following four kinds of status.

Status	Description
Same as Local	Indicates that a copy of the result business rule group already exists in the business rule manager and that its content and the content of the result business rule group are exactly similar. Consequently, no further action is taken after the search.
Modified from Runtime	Indicates that a copy of the result business rule group already exists in the business rules manager. However, another user session modified the master copy, and so the contents of the local and result business rule groups are different. The business rules manager will automatically update the local copy to get new modifications from the runtime.
Modified in Local	Indicates that a copy of the result business rule group already exists in the business rules manager. However, it has been modified by the current user. The business rules manager will use the local copy for any further actions by the user.
New to Local	Indicates that a copy of the result business rule group does not exist in the business rules manager. In this case, the business rules manager will create a local copy of the result business rule group and will also display it in the navigation pane, too.

Note: The synchronization of changes of the business rule groups occurs at the same time as the search results returned and is applied in the business rules manager context. This means that the next operation on an affected business rule group will work with the latest updates of the business rule group.

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. 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 207
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. 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 208.

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 207
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 207
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 208.

Related tasks

"Deleting scheduled rule logic entries" on page 207
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 208

Related tasks

"Deleting scheduled rule logic entries" on page 207
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 208.

Related tasks

"Deleting scheduled rule logic entries" on page 207
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 208.

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 208.

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 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.

Business Rules

With Business Rules you can view and change business rules within WebSphere Process Server to influence the performance of your business.

The Business Rules widget displays business rules within a hierarchy. At the top is the business rule group. Each business rule group contains business rule sets, which in turn contain if-then rules that determine the business logic. Some if-then rules have parameters that you can change to influence the performance of your business. If you move the cursor over a parameter, it changes shape and the parameter displays as a field that you can edit or change.

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 make up 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 > Server Types** to display the different server types.
2. Click **WebSphere 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 make up 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 > Server Types > WebSphere 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 make up 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 do 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.

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 → **Server Types** → **WebSphere application servers** → *server_name* →
   Business Integration → **Selectors** → *selector_name*.

2. Click one of the selector tables in the display to view the target components
   that make up 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 before 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 before 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 do 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.

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, do the following:

Procedure

1. Display the selector table as described in "Displaying selector tables". Click **Servers → Server Types → WebSphere application servers → *server_name* → Business Integration → Selectors → *selector_name*.**
2. Click one of the selector tables in the display to view the target components that make up 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 do 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.

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 > Server Types > WebSphere 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 > Server Types > WebSphere 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.

Working with adapters

WebSphere Process Server supports two types of adapters: WebSphere Adapters and WebSphere Business Integration Adapters. Adapters enable business applications to act as services by connecting them to diverse enterprise information systems (EISs), such as databases, enterprise resource planning systems, file systems, and e-mail systems.

With the help of an adapter, the application and EIS can communicate, sending and retrieving 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 5 on page 218 and Figure 6 on page 219 provide details of the communication between the server and the EIS for the two types of adapters.
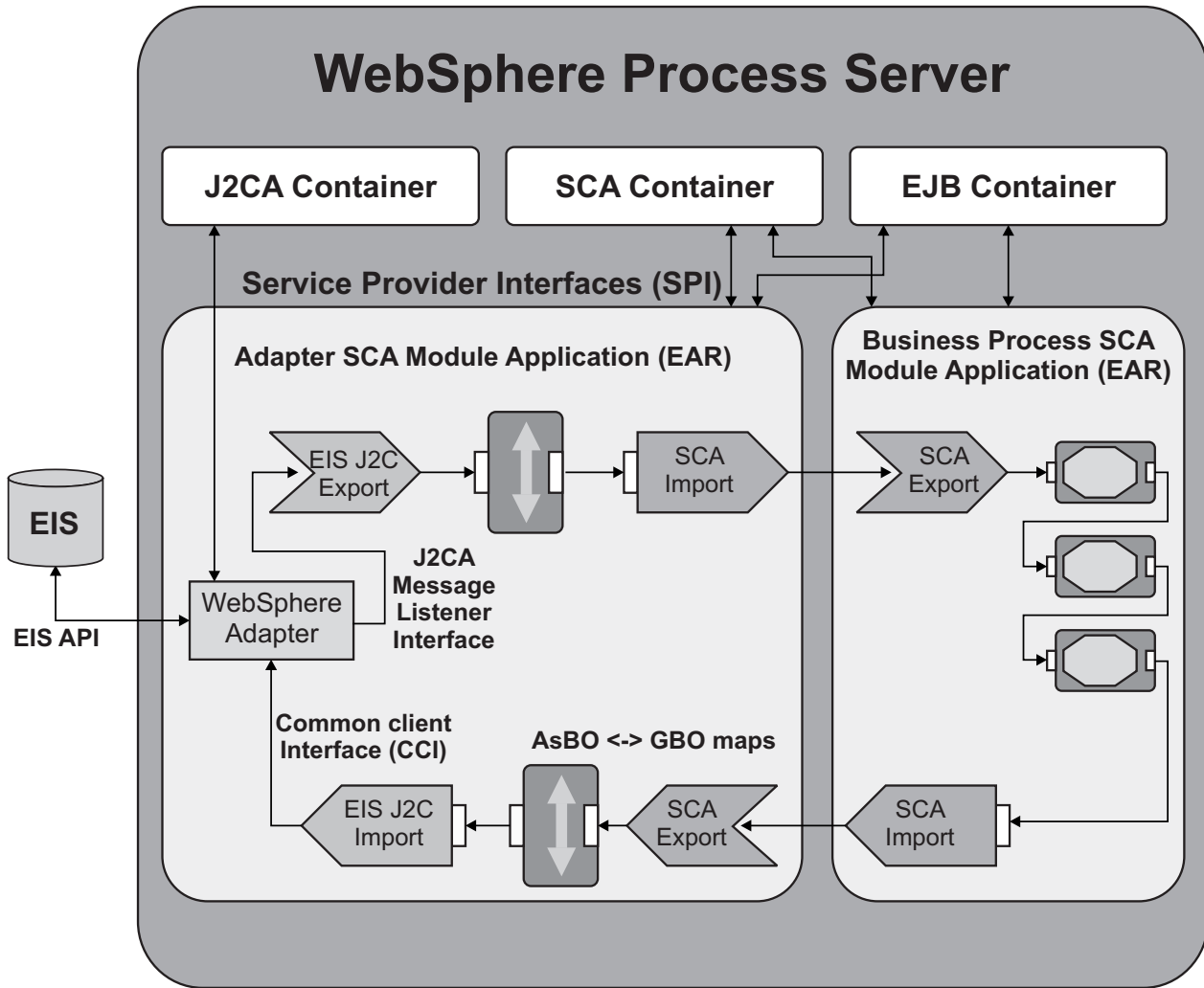
Figure 5. Detailed schematic of a WebSphere Adapter

Figure 5 depicts a WebSphere Adapter managing the connectivity between a Java EE component supported by the server and the EIS. The WebSphere Adapter resides inside the server.
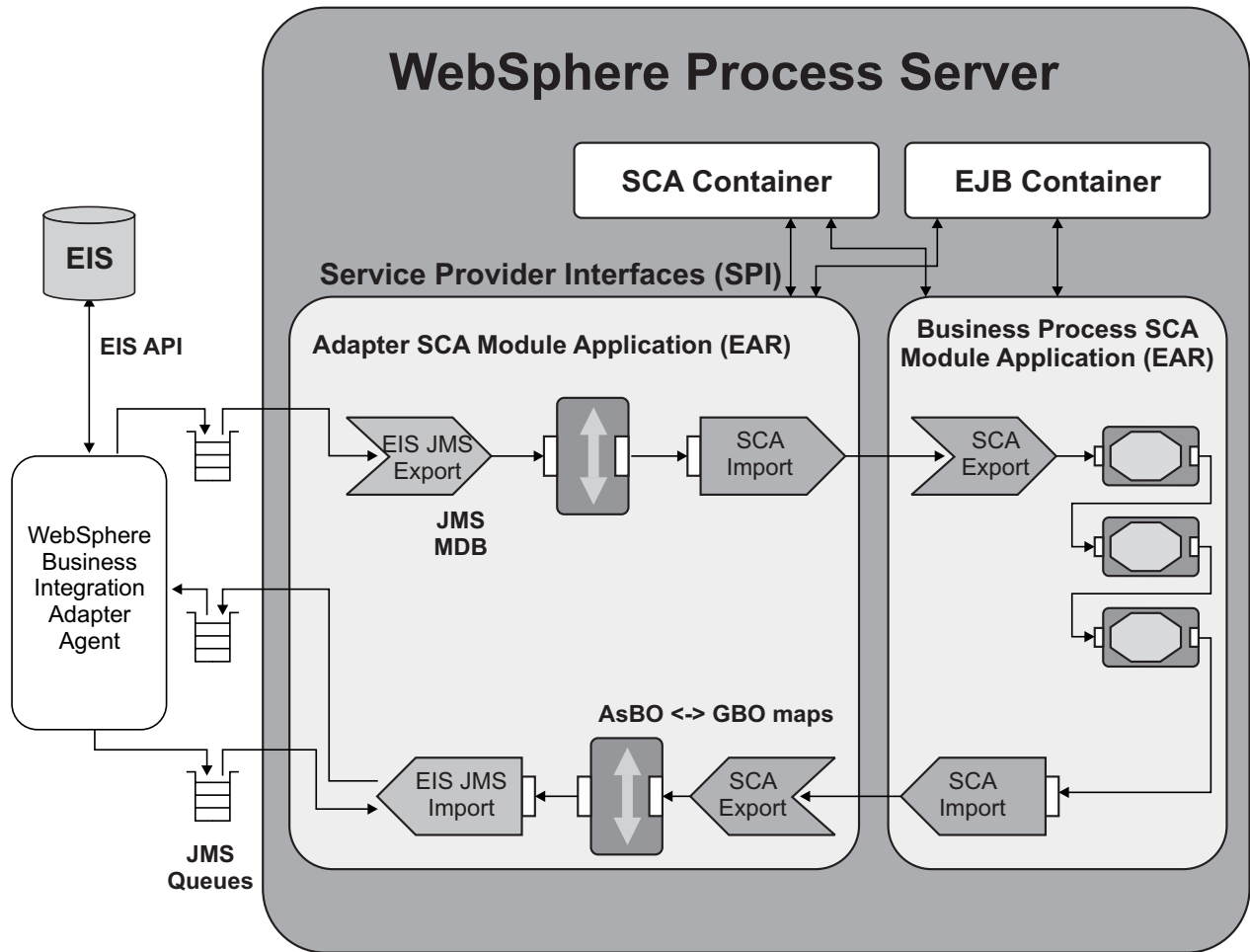
Figure 6. Detailed schematic of a WebSphere Business Integration Adapter.

Figure 6 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 28 shows the differences between the two types of adapter.

Table 28. 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 28. 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 Java EE 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.

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. Expand **Server types**.
   c. Select **WebSphere application servers**.
   d. 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.
   e. From the **Business Integration** list on the Configuration tab, select **WebSphere Business Integration Adapter Service**.
   f. Select **Manage the WebSphere Business Integration Adapter resources**.
   g. From the list of resources, 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.

Working with events

Events are requests or responses sent from one component to another. You can process events in a specific sequence. When events fail, you can use the failed event manager to view, discard, modify, or resubmit the events.You can also use the Store and forward feature to prevent subsequent failures from occurring when a component calls asynchronously to a service that is unavailable.

Processing events in sequence

Event sequencing enables WebSphere Process Server and WebSphere Enterprise Service Bus components to 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 consists of one or more attributes of the business objects associated with an invocation. All events that share the same key are grouped together and processed in sequence. Events that do not have an event sequencing key continue to be processed as normal, in parallel with the sequenced events.

A sequenced event acquires a lock before being sent to the target component for processing. As soon as the business logic for the event has executed, the lock is released and given to the next event with the same event sequencing key. If the

event cannot acquire the necessary lock, the execution of the invocation is suspended until the lock is acquired.

Related concepts

"Example: Event sequencing"
To understand how event sequencing works, consider a situation in which a source application (Component A) asynchronously invokes a target application (Component B) to create new orders, and then updates those orders with revised data.

"Considerations for implementing event sequencing" on page 226
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" on page 223
Event sequencing enables WebSphere Process Server and WebSphere Enterprise Service Bus components to process events from asynchronous invocations in the order in which they are delivered. Event order is maintained throughout the entire business integration scenario.

Considerations for implementing event sequencing

Use the information in these topics to help you plan for, implement, and troubleshoot event sequencing in your business integration environment.

Related concepts

"Processing events in sequence" on page 223
Event sequencing enables WebSphere Process Server and WebSphere Enterprise Service Bus components to 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 check box 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
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.

Enabling event sequencing for a JMS Export binding
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.

Enabling event sequencing for a WebSphere MQ JMS Export bindings
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.

Event sequencing with store-and-forward

Setting an event sequencing qualifier and a store-and-forward qualifier on the
same interface is supported. When enabled simultaneously, event sequencing is
stopped when the store is started by a qualifying runtime exception. The
sequenced processing of events resumes when the store is stopped and forwarding
is started.

When store-and-forward is in store mode, the "Process requests when error
encountered" attribute of the event sequencing qualifier is ignored.

Important: For failed events that are sequenced, you must change the
store-and-forward state to forward in order to resubmit events. Events cannot be
resubmitted when store-and-forward is in store mode.

For network deployment environments where an application cluster has two or
more members and both event sequencing and store-and-forward qualifiers are
specified, the store-and-forward service is enabled for all members of the cluster.
When an event triggers the store, event-sequencing processing stops.

For example, consider the following situation. Event sequencing and
store-and-forward qualifiers are declared for failed events being processed by a
cluster with two members, called member A and member B. The event-sequencing
service processes events on member A of the cluster. When the store is triggered by
a qualifying runtime exception, event sequencing processing stops . If member A
becomes unavailable, event sequencing is activated on member B. On member B,
event sequencing processing does not start because the store-and-forward function
is active.

For information on store-and-forward processing, see How the store-and-forward
feature works.

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 29. 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 lock up.

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 a Service Component Architecture (SCA) 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 30. 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, failed WebSphere MQ events, and failed operations within long-running business processes.

Note: For service runtime exceptions that are generated when a requested service is unavailable, you can use the Store and Forward feature to prevent further failures. You specify a store-and-forward qualifier when you configure a component that will be invoked asynchronously. When a runtime error is generated by that component, subsequent events (in this case, asynchronous requests) are prevented from reaching the component. See "Preventing failures when a service is unavailable" for more information.

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 consists 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 (for example, 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 WebSphere MQ events

A WebSphere MQ event can fail when there is a problem (such as a data-handling exception) in the WebSphere MQ binding export or import used by an SCA module.

WebSphere Integration Developer provides a recovery binding property that allows you to enable or disable recovery for each WebSphere MQ 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 WebSphere MQ bindings is enabled by default. When it is enabled, WebSphere MQ 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 WebSphere MQ

In addition, a failed SCA event is created when the `ServiceRuntimeException` exception is thrown in a WebSphere MQ binding target component after a single retry in WebSphere MQ.

These failures can occur during inbound or outbound communication. During outbound communication, MQImport sends a request message and receives the response message; a failed event is generated if the WebSphere MQ import binding detects a problem while processing the service response. During inbound communication, the sequence of events is as follows:

1. MQExport receives the request message.
2. MQExport invokes the SCA component.
3. The SCA component returns a response to MQExport.
4. MQExport sends a response message.

A failed event is generated if the WebSphere MQ export binding detects a problem while processing the service request.

The Recovery service captures the WebSphere MQ message and stores it in the failed event database. It also captures and stores the module name, component name, operation name, failure time, exception detail, and WebSphere MQ properties of the failed event. You can use the failed event manager or a custom program to manage failed WebSphere MQ events, including resubmitting or deleting the event.

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.

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.

WebSphere Integration Developer provides a recovery binding property that allows you to enable or disable 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 failed SCA 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.

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 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
   - WebSphere MQ
   - 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 31. 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 WebSphere MQ 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.

Table 31. Search criteria (continued)

Search criteria	Field or fields to use	Supported event types	Usage notes
The time period during which the event failed	**From date** **To date**	SCA JMS WebSphere MQ 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 qualifier**	SCA	None
Whether the event caused the store to be started	**Store and forward qualifier**	SCA Business Process Choreographer	None
Whether the event was caused because a failure response could not be sent to Business Process Choreographer	**Process response qualifier**	SCA	None

Table 31. 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

All failed events have the following data:

* The event ID, type, and status
* The time the event failed
* The deployment target associated with the event

In addition, SCA, JMS, WebSphere MQ, Business Process Choreographer, and Business Flow Manager hold queue events have data specific to the event type:

* SCA events
    – The session ID
    – The type of service invocation used 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
    – 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 type of service invocation used
    – The names of the destination module, component and method for 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:
  - The message type and priority
  - The JMS destination
  - The delivery mode
  - Redelivery data, including the redelivered count and redelivered indicator (`true` or `false`)
  - The destination replies are sent to for request-response or two-way interactions
- WebSphere MQ events:
  - The type of service invocation used
  - The names of the destination module, component and method for 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 WebSphere MQ-specific properties associated with the failed event:
    - The message type, format, and priority
    - The WebSphere MQ destination
    - The delivery mode
    - Redelivery data, including the redelivered count and redelivered indicator (`true` or `false`)
    - The reply-to queue and queue manager
- Business Process Choreographer events:
  - The names of the destination module and component for 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 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 Service Component Architecture (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 241

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 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 Choreographer 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 Service Component Architecture (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 that has 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 the data associated with the failed event.
4. Navigate through the business data hierarchy by clicking the name of each parameter (these appear as links in the Parameter Name column). When you have located the parameter with values 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 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 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 Service Component Architecture (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.

Resubmitting failed Business Process Choreographer responses

When a failure response cannot be delivered to the requesting business process due to an infrastructure problem, an event is stored in the failed event database. These type of events will have a process response qualifier specified on them. You can resubmit these failed events to either the request queue or the response queue using the failed event manager.

About this task

To resubmit a failed SCA event, perform the following steps.

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** or **Resubmit with trace**.
4. If the Process Response event qualifier is defined for the failed event, a resubmit page will appear. Select **Resubmit requests to the destination** or **Resubmit the exception response to the source**. Selecting **Resubmit the exception response to the source** allows the event to be sent to the response queue without having to be reprocessed.

Results

Depending on whether you selected **Resubmit requests to the destination** or **Resubmit the exception response to the source**, the event will be resubmitted to the appropriate queue.

Managing failed SCA events

When problems processing a Service Component Architecture (SCA) request or response message create a failed SCA 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

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. Because runtime exceptions are not declared as part of the interface, component developers should attempt to resolve the exception and thus prevent a runtime exception from inadvertently being propagated to the client if the client is a user interface.

To manage a failed SCA event, perform the following steps.

Procedure

1. Use the failed event manager to locate information about the failed SCA event, taking note of the exception type.
2. Locate the exception type in Table 32 to determine the location and possible causes of the error, as well as suggested actions for managing the failed event.

Table 32. Failed SCA events

Exception type	Possible cause of error	Suggested action
ServiceBusinessException	A business exception occurred during the execution of a business operation.	Look at the exception text to determine the exact cause, and then take appropriate action.

Table 32. Failed SCA events (continued)

Exception type	Possible cause of error	Suggested action
ServiceExpirationRuntimeException	A SCA asynchronous message has expired .	Set the expiration time using the RequestExpiration qualifier on the service reference. Investigate why the service is not responding fast enough.
ServiceRuntimeException	A runtime exception occurred during the invocation or execution of a service.	Look at the exception text to determine the exact cause, and then take appropriate action.
ServiceTimeoutRuntimeException	Response to an asynchronous request was not received within the configured period of time.	Set the expiration time using the RequestExpiration qualifier on the service reference. Investigate why the service is not responding fast enough.
ServiceUnavailableException	This exception is used to indicate that there was an exception thrown while invoking an external service via an import.	Look at the exception text to determine the exact cause, and then take appropriate action.
ServiceUnwiredReferenceRuntimeException	A SCA reference used to invoke a service is not wired correctly.	Look at the exception text to determine the exact cause, and then take appropriate action to correctly wire the SCA reference.

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 33 on page 252 to determine the location and possible causes of the error, as well as suggested actions for managing the failed event.

Table 33. 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 33. 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 failed WebSphere MQ events

When problems processing a WebSphere MQ request or response message create a failed WebSphere MQ 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 WebSphere MQ event, perform the following steps.

Procedure

1. Use the failed event manager to locate information about the failed event, taking note of the exception type.
2. Locate the exception type in Table 34 to determine the location and possible causes of the error, as well as suggested actions for managing the failed event.

Table 34. Failed WebSphere MQ events

Exception type	Location of error	Possible cause of error	Suggested action
FaultServiceException	Fault handler or fault selector	There is malformed data in the WebSphere MQ message.	1. Inspect the 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.

Table 34. Failed WebSphere MQ events (continued)

Exception type	Location of error	Possible cause of error	Suggested action
ServiceRuntimeException	Fault handler	The fault selector and runtime exception handler are configured to interpret the WebSphere MQ 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 WebSphere MQ message.	1. Inspect the 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.
SelectorException	Function selector	There is malformed data in the WebSphere MQ message.	1. Inspect the 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 241 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 241 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"
I am having trouble deleting expired events	"Using the Delete Expired Events function appears to suspend the failed event manager"
I am having trouble with failed events not being created	"Failed events are not being created" on page 258

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 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 z/OS, 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 logs directory of the profile path.

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.

Table 35. 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 *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. • – 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. • **Managed node 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 SystemErr.log and SystemOut.log are found in deployment manager job logs unless customized to write to a file. The start and stop server logs are found in *profile_root*/logs/dmgr if started in USS.	You will work primarily with two 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.
Node agent logs SystemErr.log and SystemOut.log are found in node agent job logs unless customized to write to a file. The start and stop server logs are found in *profile_root*/logs/nodeagent if started in USS.	You will work primarily with two 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.

Table 35. Profile-specific log files updated during runtime (continued)

Log	Contents
Server logs SystemErr.log and SystemOut.log are found in server job logs unless customized to write to a file. The start and stop server logs are found in *profile_root*/logs/<servername> if started in USS.	You will work primarily with two 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.
Node federation log files are found in the *profile_root*/logs/ directory.	Two log files are generated when you attempt to federate a node 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):The location of the Integrated Solutions Console application deployment log file is in the *<profilepath>*/logs directory.	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 log files for a profile creation is the *install_root*/logs/ directory.	These files contain the output from profile creation. 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 257
I am having trouble deleting expired events	"Using the Delete Expired Events function appears to suspend the failed event manager" on page 257
I am having trouble with failed events not being created	"Failed events are not being created" on page 258

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 store-and-forward processing

This topic discusses problems that you can encounter with store-and-forward processing.

Select the problem you are experiencing from the table below:

Problem	Refer to the following
I am having problems setting the store-and-forward qualifier	"Store-and-forward qualifier processing only works on asynchronous interfaces"
Qualifying runtime exceptions are occurring, but events are not getting stored	"Store is not activated by qualifying runtime exceptions"
Messages are still being processed even though the Store and Forward widget shows the state is set to Store (Network deployment environment)	"In a network deployment environment, messages are being processed even though the store-and-forward state is set to Store" on page 264
The Store and Forward widget shows the state is set to Forward, but messages are not being processed by all members of the cluster. (Network deployment environment)	"In a network deployment environment, messages are not getting processed by all members of the cluster even though the store-and-forward state is set to Forward" on page 264

Store-and-forward qualifier processing only works on asynchronous interfaces

The store-and-forward qualifier must be specified on an asynchronous interface. The store cannot be activated if the interface is called synchronously.

Here are some guidelines (with respect to components) to help you determine if the interface is being called synchronously or asynchronously.

- Examine your short-running business process and what import it invokes. For example, JMS is an asynchronous import. Therefore, it is called asynchronously by a short-running process. HTTP is a synchronous import. Therefore, it is called synchronously.
- Long-running processes invoke imports based on the preferred interaction style set on the import's interface. Look at the interaction style set on the import's interface to see whether it is synchronous or asynchronous.

    Note: You can find this setting on the interface's detail tab.
- POJO components invoke components based on the code that is written in the component. Look at the code written in the component to see whether it is synchronous or asynchronous.

Also, consider these restrictions:
- Store-and-forward qualifier cannot be set on long-running processes.
- Store-and-forward cannot be set on exports (except SCA export).

Store is not activated by qualifying runtime exceptions

If the store is not being activated by qualifying runtime exceptions, check the following.

- The exception specification in the store-and-forward qualifier matches the exception that occurs at runtime. If the exception specification does not match, storing does not activate.

- The user code in the path is not catching the exception and wrapping it. Or, it is converting it into a different exception. The exception received by the store-and-forward function can be viewed in the exception details for the failed event.
- The destination component for a failed event has a store-and-forward qualifier set on it. Storing is activated once a failed event is generated. If a failed event is generated for a component that is upstream from the component that has a store-and-forward qualifier set on it, then the store-and-forward component is being invoked synchronously and not asynchronously. If a failed event is generated for a component that is downstream from the store-and-forward qualifier component, rather than the component with the store-and-forward qualifier set on it, then there is an asynchronous invocation closer to the failure and the store-and-forward qualifier should be moved to that component.

In a network deployment environment, messages are being processed even though the store-and-forward state is set to Store

Messages might continue to be processed by some members of a cluster, despite the state being set to Store, if the state is not set to Store for each member of the cluster. To fix this problem, confirm that the state is set to Store for each member of the cluster in the Store and Forward widget. If any members of the cluster are set to Forward, change them to Store.

This might also happen if one of the members of the cluster is forced to restart. Since the Store state is not persistent, it reverts to the Forward state at restart. To fix this problem, change the state to Store for the module in the Store and Forward widget.

Note: When the service becomes available again, you should not set the state to Store immediately if you want new events to be processed. If you set the state to Store before new events have the chance to be processed, they will be stored in the queue.

In a network deployment environment, messages are not getting processed by all members of the cluster even though the store-and-forward state is set to Forward

Messages might continue to be stored by some members of a cluster, despite the state being set to Forward, if the store-and-forward state is not set to Forward for each member of the cluster. To fix this problem, confirm that the state is set to Forward for the module in the Store and Forward widget. If any members of the cluster are set to Store, change them to Forward.

Note:

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.

Cross-component trace

Cross-component trace identifies whether a Service Component Architecture operation completed successfully.

Cross-Component Trace overview

Cross-Component Trace identifies whether a Service Component Architecture (SCA) operation completed successfully. It provides information about what modules were used at run time and what service calls were made. Additionally, you can configure Cross-Component Trace with *data snapshot* to capture data sent in and passed between SCA components.

When you enable Cross-Component Trace, invocation records are generated during SCA processing of modules and components. The invocation records help to identify the resulting call chains, and to correlate log messages, first-failure data capture (FFDC) and directly logged exceptions to these call chains. The separation of these messages along call chains serves two purposes as follows:

1. Problem determination using WebSphere Integration Developer
2. A better understanding of the run time behavior of the system, including an understanding of how the SCA service calls move through the system.

Note: The information in this topic focuses on using Cross-Component Trace for problem determination.

When you configure Cross-Component Trace with data snapshot, the generated invocation records contain the invocation input and output data that was passed between the components during processing. The log records associated with the WebSphere Process Server applications hold information about errors or events that occurred during processing and can be used for problem determination using WebSphere Integration Developer.

Cross-component trace uses the WebSphere Application Server logging service to capture information and to provide context for the other entries in the logs. The information pertaining to SCA component processing (including the sequential flow of the event) is captured by Cross-Component Trace and written to the `systemout.log` or `trace.log` file according to well defined rules. You can set Cross-Component trace logging parameters to write the records to either the `trace.log` or `systemout.log`

Note: The extra data about what was passed between modules can be large and is kept in separate files and not in the trace.log or systemout.log.

You can enable and disable Cross-Component Trace from either the administrative console or from the Server Log view in WebSphere Integration Developer.

You can use the Server Log view in WebSphere Integration Developer to display invocation records that can contain the invocation data that passed between components. Using the Server Log view you can filter records, display invocation records in hierarchical format, and load invocation records directly into the integration test client. For more information about the Server Log view and Cross-Component Trace from the development environment perspective, see the WebSphere Integration Developer information center.

Cross-Component Trace processing runs on the server. When enabled, the processing associated with Cross-Component Trace is applied to all modules in the server. In cases where there are many modules and solutions deployed in the server, Cross-Component Trace can affect system performance. Cross-Component Trace adds additional data in the `systemout.log` and `trace.log` files and the data snapshot option further increases disk space usage and system performance. In development and test systems, the effect on performance should not present a problem. However, in production, such an affect on performance might not be acceptable. Much of the extra data captured might be for modules that do not need Cross-Component Trace enabled. For information about the effect Cross-Component Trace has on system performance, see Cross-component trace and system performance.

In situations where you have many modules deployed in the server, but only one on which you want to apply Cross-Component Trace, consider enabling application-specific Cross-Component Trace. For information about application-specific Cross-Component Trace, see Application-specific Cross-Component Trace.

Enabling Cross-Component Trace for BPEL long-running instances

Take into consideration the following information when enabling Cross-Component Trace for BPEL long-running process instances:

- To create a Cross-Component Trace for long running BPEL process instances, you must select **Enable Cross-Component Trace** and **Trace all**, or enable Cross-Component Trace for the desired SCA module before the BPEL process instance is created. If a BPEL process instance is created when Cross-Component Trace is not enabled, then future changes to the Cross-Component Trace configuration will not result in Cross-Component Trace data being captured for work done by the process instance.
- Server-level Cross-Component Trace settings may change multiple times over duration of a long running BPEL process instance. This can result in *gaps* in the Cross-Component call chains for the instance.

  Note: Deleting old logs can also result in the loss of call chain data.

In summary, Cross-Component Trace for long-running BPEL process instances works best at the server-level rather than for specific modules. Cross-Component Trace for specific modules for BPEL long-running process instances lasts for the duration of the BPEL process instance. Consider carefully using application-specific Cross-Component Trace for long running BPEL processes in a production environment. Enabling Cross-Component Trace, without setting the **Trace all** parameter or without listing any SCA modules for trace, will affect performance of any server that has BPEL, because each internal step of BPEL checks to see if Cross-Component Trace is on for that instance.

Cross-Component Trace in the development environment

You can turn on Cross-Component Trace from the Server Log view in WebSphere Integration Developer. Typically, developers would enable Cross-Component Trace from the Server Log view (rather than the administrative console) as part a development testing protocol. If you are running an application against your unit test environment (UTE), the Cross-Component Trace affect on performance resulting from hundreds of applications installed would not apply (a unit test environment would most likely only have one application installed).

Related concepts

▶ Process instances
A process instance is the instantiation of a process template.

Related tasks

"Configuring logging for Cross-Component Trace using the administrative console"
on page 285
You can use the administrative console to configure Cross-Component Trace
logging.

"Enabling Cross-Component Trace using the administrative console" on page 279
You can use the administrative console to prepare the server for Cross-Component
Trace operations.

"Enabling Cross-Component Trace with data snapshot using the administrative
console" on page 281
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.

"Enabling Cross-Component Trace levels using the command line" on page 283
You can set Cross-Component Trace levels and determine the current trace level
settings using the wsadmin command. Setting Cross-Component Trace level
involves specifying parameters that determine whether to enable Cross-Component
Trace, and if enabled, what type of trace operations are to be performed.

Related information

▶ Using the Server Logs view for problem determination

▶ Enabling or disabling Cross-Component tracing from the Server Log view

▶ WebSphere Application Server: Tracing and logging configuration

Application-specific Cross-Component Trace
Application-specific Cross-Component Trace provides the same functionality as
Cross-Component Trace, but applies the functionality to a single Service
Component Architecture (SCA) module.

When you apply Cross-Component Trace to a single SCA module (rather than to
an entire server) you reduce the performance and resource affect to the server.

When you enable application-specific Cross-Component Trace, the trace activities
persist for the duration of the call, from invocation to completion, including any
"fan-out calls" to other modules, even when those other modules reside on other
servers. The call chain produced by Cross-Component Trace and
application-specific Cross-Component Trace indicates the flow of an event,
including a sequential list of all SCA components in the module that were involved
with processing the event. For example, when a customer applies for a home loan
over the internet, the request goes into the HomeLoan module and is processed
using the various components in the module. At some point, an external call is
made to determine the credit rating of the customer that is applying for the home
loan. If application-specific Cross-Component Trace is turned on for the home loan
module, it shows the flow of the request in the module, and it also shows the
external callout for the customer credit rating, even when the credit rating process
resides on another server. Using the log files (`trace.log` or `systemout.log`) from all
the servers, you can view the activity associated with the home loan request.

Note: Cross-Component Trace logging is configurable.

Note: Application-specific Cross-Component Trace flows over SCA bindings, including across Java Virtual Machines (JVM).

You can enable application-specific Cross-Component Trace in the same manner as Cross-Component Trace:

- Enable Cross-Component Trace

  Enabling Cross-Component Trace on the server or for a specific SCA module provides trace data related to SCA processing.

- Enable with data capture

  Enabling Cross-Component Trace with data capture on the server or for a specific module means that the generated invocation records contain input and output data that was passed between SCA components during processing.

Through settings on the administrative console, you can choose specific SCA modules on which to enable Cross-Component Trace and you can indicate on the selected modules whether or not you want the data snapshot feature enabled. For more information see *Enabling Cross-Component Trace for selected SCA modules.*

Using application-specific Cross-Component Trace and server Cross-Component Trace together

In some scenarios, enabling Cross-Component Trace for specific SCA modules does not provide benefits above what is provided by server Cross-Component Trace. However, there might be situations where enabling both server and application-specific Cross-Component Trace makes sense.

The following table describes the implications of implementing both Cross-Component Trace (on the server) and Cross-Component Trace for specific SCA modules.

Table 36. Cross-Component Trace and application-specific Cross-Component Trace

Type of trace	Data snapshot Y/N?	Implications
Cross-Component Trace (server)	Yes (for all servers)	In this scenario, turning Cross-Component Trace on for a specific SCA module would not provide any additional benefits.
Cross-Component Trace (server)	Yes (for some servers)	In this scenario, if you also enable Cross-Component Trace for a specific SCA module, the call chains for that module that go to other Java virtual machines (JVM) (where server Cross-Component Trace is not turned on) are enabled with Cross-Component Trace.

If Cross-Component Trace is enabled on the server, then Cross-Component Trace for a specific module can be used for one, or both of the following purposes:

- To enable the data snapshot feature for a specific SCA module
- To provide call chain support to other servers for which Cross-Component Trace is enabled, but **Trace all** is not selected.

Enabling application-specific Cross-Component Trace for BPEL long-running instances

Take into consideration the following information when enabling application-specific Cross-Component Trace for BPEL long-running process instances:

- To create a Cross-Component Trace for long running BPEL process instances, you must select **Enable Cross-Component Trace** and **Trace all**, or enable Cross-Component Trace for the desired SCA module before the BPEL process instance is created. If a BPEL process instance is created when Cross-Component Trace is not enabled, then future changes to the Cross-Component Trace configuration will not result in Cross-Component Trace data being captured for work done by the process instance.
- If the server-level Cross-Component Trace is turned off (for example, if **Enable Cross-Component Trace** is not selected), call chains are not created for the BPEL work.
- Server-level Cross-Component Trace settings may change multiple times over the life of a long-running BPEL process instance. This can result in *gaps* in the Cross-Component call chains for the instance.

    Note: Deleting old logs can also result in the loss of call chain data.
- If server-level Cross-Component Trace is enabled, but not set to **Trace all**, there will be some BPEL applications in which there are *gaps* in the call chains. This can happen when the BPEL includes a *Receive* task.

In summary, Cross-Component Trace for long running BPEL process instances works best at the server-level rather than for specific modules. Application-specific Cross-Component Trace for BPEL long running process instances lasts for the duration of the BPEL process instance. Consider carefully using application-specific Cross-Component Trace for long running BPEL processes in a production environment. Enabling Cross-Component Trace, without setting **Trace all** or without listing any SCA modules for trace, will affect performance of any server that has BPEL, because each internal step of BPEL checks to see if Cross-Component Trace is on for that instance.

Flow of events for using application-specific Cross-Component Trace

The description that follows portrays a scenario where one might implement Cross-Component Trace on a specific SCA module:

1. Although there are many applications running on **Appserver1**, you notice a problem with the HomeLoan module.

    Problems with specific modules can display in various ways. A problem might display in the server logs, or a developer might be alerted to a problem when the data resulting from processing does not seem to be correct. Additionally, you might be curious about the outcome of a call made to the module.

2. You decide to turn on Cross-Component Trace for the HomeLoan module that is running on **Appserver1**.

    The decision to enable Cross-Component Trace for the HomeLoan application means that *call chains* between the SCA components are captured for the HomeLoan module.

    Although the Cross-Component Trace is set on the HomeLoan module, when the processing by that module requires calls to other modules, the call chains

track that call processing as well, even if the call is made to a module that resides on different server. Call chains continue until the call execution is completed.

No call chains are created or made available for the work/events processed by any of the other applications on **Appserver1** that are separate from and not referenced by the HomeLoan module. Even if an application is referenced by the HomeLoan module, only work done in that application that are based on calls from the HomeLoan module will have call chains created

3. You analyze the data captured from the Cross-Component Trace of the HomeLoan module.

   The correlated records from logs and additional SCA context data captured for a particular point in the SCA calling chain can be viewed using a Server Log view in WebSphere Integration Developer.

Related concepts

Process instances
A process instance is the instantiation of a process template.

"Enabling Cross-Component Trace for selected SCA modules" on page 288
You can enable Cross-Component Trace on specific Service Component Architecture (SCA) modules on a server. Use Cross-Component Trace to identify trace.log and systemout.log data that is associated with WebSphere Process Server and WebSphere Enterprise Service Bus modules and components.

Related tasks

"Enabling application-specific Cross-Component Trace using the administrative console" on page 289
You can use the administrative console to choose specific SCA modules on which to run Cross-Component Trace.

Cross-Component Trace and system performance
The runtime processing activities associated with Cross-Component Trace can affect system performance.

Even a slight affect on performance might not acceptable, especially in a production environment. If you do not want to incur any affect on performance, you can turn off Cross-Component Trace.

Because of the multiple configuration options associated with implementing Cross-Component Trace (Cross-Component Trace on the server or Cross-Component Trace on specific modules, Cross-Component Trace with or without data capture), it is important that you understand the effect that Cross-Component Trace configuration decisions have on system behavior. This understanding is relevant regarding turning off the Cross-Component Trace feature. The following table lists and describes the effect of turning Cross-Component Trace on or off.

Table 37. Affect on system performance of turning Cross-Component Trace on or off

Setting on administrative console	wsadmin Parameter	Description of Effect
Enable Cross-Component Trace=cleared	setLevel=disable	**Effect on setting parameters** If you do not select **Enable Cross-Component Trace**, or if you use wsadmin to set Cross-Component Trace to **setLevel=disable**, the system behaves as if Cross-Component Trace does not exist at all. There is no system checking associated with trace activities and no affect on performance. If you disable Cross-Component Trace from the administrative console, the system prevents you from setting any other Cross-Component Trace configuration parameters. If you use the command-line equivalent (wsadmin) to set Cross-Component Trace configuration parameters and **Enable Cross-Component Trace** is not selected on the administrative console, you can still set parameters using wsadmin (the system does not throw an exception). The configuration parameters that you set using the command line are preserved for when you turn on Cross-Component Trace. **Effect on runtime processing** In a multi-server environment, the manner in which you configure Cross-Component Trace effects the processing associated with trace activities. For example, if you have turned off Cross-Component Trace on server A, and you have enabled application-specific Cross-Component Trace on server B, and the application running on server B calls server A, call chain information does not result.

Table 37. Affect on system performance of turning Cross-Component Trace on or off (continued)

Setting on administrative console	wsadmin Parameter	Description of Effect
Enable Cross-Component Trace=checked	When **Enable Cross-Component Trace** is selected, the wsadmin setting is in one of three different states as follows: • Enable • Enable with data snapshot • Ready It depends on the other selections in the console as to what the WSADMIN state is.	**Effect on setting parameters** When you turn Cross-Component Trace on by selecting the **Enable Cross-Component Trace** check box, the system allows you to set all other configuration parameters associated with Cross-Component Trace, either by setting the parameters on the administrative console, or by setting the parameters using the command-line equivalent (wsadmin). **Effect on runtime processing** When Cross-Component Trace is turned on, expect a slight affect on performance. Turning on Cross-Component Trace results in the server support for application-specific trace coming from other servers, as well as those originating on the current server. When Cross-Component Trace is enabled, there is an adverse affect on the performance for long running BPEL processes, *even if you have not selected Trace all* and there are no modules listed in the table for **Enable tracing for the selected Service Component Architecture (SCA) modules**.
Enable with data snapshot = checked	setLeve l= enable with data snapshot	Implementing the data snapshot feature of Cross-Component Trace affects performance and disk space usage. You should keep this in mind when using Cross-Component Trace on production systems. In situations where you want to run Cross-Component Trace on most or on all modules, but require the data snapshot feature for only a few, consider setting the **Trace all** parameter and then using the **Enable tracing for the selected Service Component Architecture (SCA) modules** table to enable data snapshot on only those modules that require it.

Table 37. Affect on system performance of turning Cross-Component Trace on or off (continued)

Setting on administrative console	wsadmin Parameter	Description of Effect
Save cross-component trace output to = trace or system	Cross-Component Trace uses the WebSphere Application Server logging service to capture information and to provide context for the other entries in the logs. For more information, see *Tracing and logging configuration* in the WebSphere Application Server information Center.	The logging option that you choose for Cross-Component Trace can have an affect on system performance. You can save Cross Component Trace output to the `trace.log` file or to the `systemout.log` file. Choosing to save Cross Component Trace output to `systemout.log` file takes the system more time to write out the log. For more information on the processing associated with logging and Cross-Component Trace, see *Configuring logging for Cross-Component Trace*.

Related concepts

"Troubleshooting Service Component Architecture processing and call chains" Cross-Component Trace identifies whether a Service Component Architecture (SCA) operation completed successfully. It 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 log records associated with the WebSphere® ESB applications hold information about errors or events that occurred during processing and can be used for problem determination using WebSphere Integration Developer.

Related tasks

"Configuring logging for Cross-Component Trace using the administrative console" on page 285
You can use the administrative console to configure Cross-Component Trace logging.

"Enabling Cross-Component Trace levels using the command line" on page 283
You can set Cross-Component Trace levels and determine the current trace level settings using the wsadmin command. Setting Cross-Component Trace level involves specifying parameters that determine whether to enable Cross-Component Trace, and if enabled, what type of trace operations are to be performed.

Related information

WebSphere Application Server: Tracing and logging configuration

Troubleshooting Service Component Architecture processing and call chains

Cross-Component Trace identifies whether a Service Component Architecture (SCA) operation completed successfully. It 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 log records associated with the WebSphere® ESB applications hold information about errors or events that occurred during processing and can be 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**.

Handling and deleting collected data

Consider the following with regard to handling and deleting data collected by Cross-Component Trace:

- SCA call chain information is added to the systemout.log and trace.log files and is purged as those files are purged.
- Data snapshots capture the input and output data of call chains.

  The input and output data is captured as files in the `logs\XCT` directory. To view this data using WebSphere Integration Developer, WebSphere Integration Developer needs access to the systemout.log files and the `logs\XCT` directory. If WebSphere Integration Developer is not available on the server, copying the logs directory and placing it on a machine (so that it can be accessed by WebSphere Integration Developer) preserves the file structure so that WebSphere Integration Developer can make use of the log files and the data snapshot files.

  Note: WebSphere Integration Developer can use the data snapshot files where they are (without moving them) if it can access the files in the logs directory. If you need to move files, it is safest to move the entire logs directory. By moving the entire logs directory you get the XCT, first failure data capture (FFDC) files, and the systemout.log and trace.log files.

  Data snapshot files are written to server-specific subdirectories using the following directory structure:

  ```
  logs\
       server
       ffdc
       xct\
           server-specific_dir\
                            2009-0-25-11
                            2009-0-26-12
                            2009-0-26-14
  ```

  Where `server-specific_dir` name is derived from the name of the server. For example, **server1** is the default server name for a stand-alone installation.

- Data snapshot files in `logs\XCT\server` are referenced from the systemout.log and trace.log files that were created at the same time by the server. When WebSphere Application Server deletes the old systemout.log and trace.log files, the associated Cross-Component Trace data snapshot files in `logs\XCT\server` can also be deleted.

  You can use the timestamps in the systemout.log and trace.log files to identify and determine what data snapshot files to delete. It is safe to delete all the data snapshot files for a server that are older than the oldest date in the systemout.log and trace.log files. Preferably, you should use the **Delete data snapshot files** function from the administrative console when data snapshot files are no longer needed. For detailed information on the ways that you can delete data snapshot files, see *Deleting data snapshot files*.

- Do not save or add files to the `logs\XCT` directory. Do not copy or create new directories into the `logs\XCT` directory.

WebSphere Process Server manages the content of the `logs\XCT` directory and deletes items that are no longer needed. WebSphere Process Server might consider unrecognized files or directories as unnecessary and delete them. If you want to save a copy of the data snapshot files, copy the data to another directory outside of the `logs\XCT` directory.

Cross-Component Trace settings and call chain processing

The information in this section describes the affect that Cross-Component Trace configuration settings have on call-chain processing.

The information in this section includes a description of various Cross-Component Trace configurations and explains the call chain events that result from the configurations.

General rules on call chain processing and Cross-Component Trace configuration decisions

- If Cross-Component Trace is turned off for a server, then no Cross-Component Trace records are written to that server's logs.
- Cross-Component Trace configuration settings for a particular server, *affect that server only*.

  For example, if Server A has **Trace all = Yes** and Server B has **Trace all = No**, Cross-Component call chains are in the logs for Server A only. Similarly, this rule applies to setting the data snapshot feature. If **Enable data snapshot = Yes** on Server A and **Enable data snapshot = No** on Server B, then only Server A will have data snapshot files in its logs directory.
- Application-specific Cross-Component Trace data flows between servers that have the **Enable Cross-Component Trace = Yes**.

  For example, if both Server A and Server B have **Enable Cross-Component Trace = Yes** and Server A has enabled Cross-Component Trace for a specific SCA module, the calls made from the Cross-Component Trace-enabled module on Server A (to applications or services on Server B), will result in Server A having call chains for all of activity related to the Cross-Component Trace-enabled module. Server B would also have call chains, but only for those calls that came from the Cross-Component Trace-enabled module on Server A. The logs of the two servers can be combined to reveal the entire call chain activity.
- To create a Cross-Component Trace for long running BPEL process instances, you must select **Enable Cross-Component Trace** and **Trace all**, or enable Cross-Component Trace for the desired SCA module before the BPEL process instance is created.

  For more information see *Enabling Cross-Component Trace for BPEL long-running instances* in the Cross-Component Trace overview.

The following illustration is of two servers (Server A and Server B), both with Cross-Component Trace enabled. Server A has the **Trace all** value set to Yes, while Server B has **Trace all** set to No.
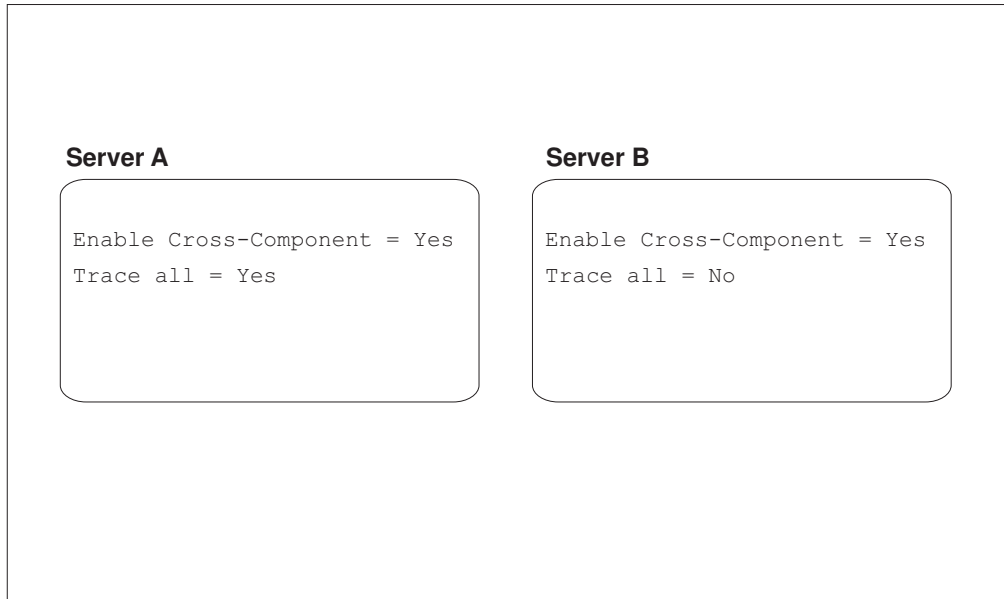
```
┌─────────────────────────────────────────────────────────────────────┐
│                                                                     │
│   Server A                            Server B                      │
│   ╭─────────────────────────────╮    ╭─────────────────────────────╮│
│   │                             │    │                             ││
│   │ Enable Cross-Component = Yes│    │ Enable Cross-Component = Yes ││
│   │ Trace all = Yes             │    │ Trace all = No              ││
│   │                             │    │                             ││
│   │                             │    │                             ││
│   │                             │    │                             ││
│   ╰─────────────────────────────╯    ╰─────────────────────────────╯│
│                                                                     │
│                                                                     │
│                                                                     │
│                                                                     │
│                                                                     │
└─────────────────────────────────────────────────────────────────────┘
```

Result: For the Cross-Component Trace configuration scenario illustrated above, call chain events would result on Server A, but not on Server B.

The following illustration is of two servers (Server A and Server B), both with Cross-Component Trace enabled. Server A has the **Trace all** value set to No and it includes Module A as a module on which to enable Cross-Component Trace. Server B has **Trace all** set to No and has no SCA modules selected for Cross-Component Trace.
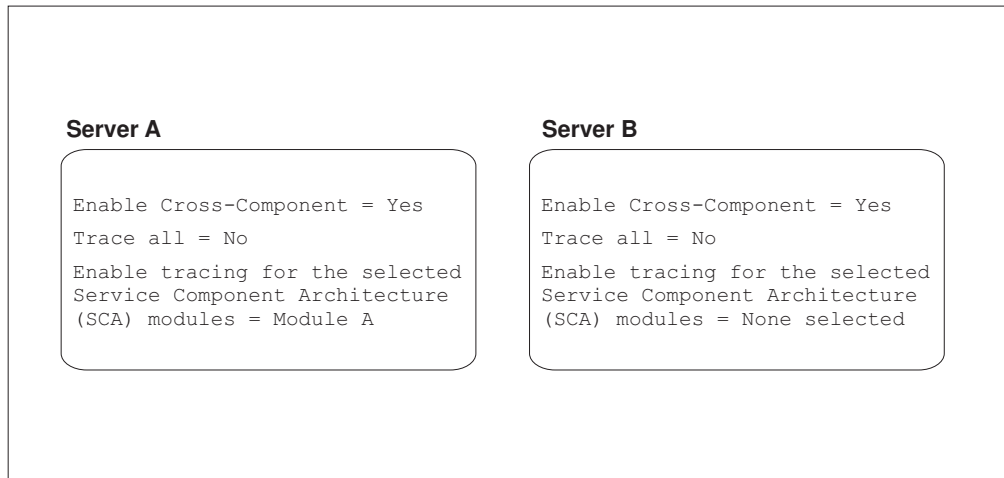
```
┌─────────────────────────────────────────────────────────────────────┐
│                                                                     │
│                                                                     │
│                                                                     │
│   Server A                            Server B                      │
│   ╭─────────────────────────────╮    ╭─────────────────────────────╮│
│   │ Enable Cross-Component = Yes│    │ Enable Cross-Component = Yes ││
│   │ Trace all = No              │    │ Trace all = No              ││
│   │ Enable tracing for the selected│ │ Enable tracing for the selected│
│   │ Service Component Architecture│  │ Service Component Architecture ││
│   │ (SCA) modules = Module A    │    │ (SCA) modules = None selected││
│   ╰─────────────────────────────╯    ╰─────────────────────────────╯│
│                                                                     │
│                                                                     │
└─────────────────────────────────────────────────────────────────────┘
```

Result: For the Cross-Component Trace configuration scenario illustrated above, call chain events would result on Server A. Trace activity for all Module A operations are written to the log on Server A. Any calls made from Module A to applications or services on Server B, results in call chains. The call chains on Server B would only pertain to those calls that came from Module A (because that module is configured for Cross-Component Trace).

The runtime processing activities associated with Cross-Component Trace can affect system performance.
You can delete the data snapshot files from the `logs\XCT` directory to free up disk space. 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. When the data from the systemout.log and trace.log is deleted, the corresponding data snapshot files remain on the disk. You can delete these files manually or you can set a parameter on the administrative console which results in data snapshot files being deleted by the system periodically.

Related information

➦ Loading server console and log files into the Server Logs view

Enabling Cross-Component Trace for the server

You can enable cross-component trace on a server. Use cross-component trace to identify `trace.log` and `systemout.log` data that is associated with WebSphere Process Server and WebSphere Enterprise Service Bus modules and components.

Enabling Cross-Component Trace using the administrative console

You can use the administrative console to prepare the server for Cross-Component Trace operations.

Before you begin

You must be logged in as administrator to perform this task.

About this task

This task describes how to enable Cross-Component Trace using the administrative console. The parameters that you set using this procedure control the various aspects of Cross-Component Trace behavior, including the level at which trace operations occur. For information on what Cross-Component Trace does and how it works, see *Cross-Component Trace overview*.

To enable Cross-Component Trace on a server, use the following procedure.

Procedure

1. Ensure that the administrative console is running, and then click **Troubleshooting** → **Cross-Component Trace** to display the list of servers. In a network deployment environment, observe the operating status of the selected servers in the **Status** column: **running** or **not running**. If the status of the server is **not running**, runtime tracing for that server is disabled and you can specify only parameters on the **Configuration** tab of the Cross-Component Trace configuration page. In this case, the trace parameters that you set take effect only when the server starts or restarts.

   Note: The **Status** column does not display for stand-alone servers.
2. From the list of servers, select a server on which to enable Cross-Component Trace.

Clicking the link in the **Server** column takes you to one of two possible Cross-Component Trace configuration pages, depending on the version of the server. Version information for the server displays in the table column labeled **Version**.

If the **Version** information displays version 7.0, proceed to 3 for instructions.

If the **Version** information displays version 6.2.0.x, proceed to 4 for instructions

3. Enable Cross-Component Trace for version 7.0 server You can enable Cross-Component Trace on either the **Configuration** tab or the **Runtime** tab.

   If you enable Cross-Component Trace on the **Configuration** tab, the setting is applied by the system whenever the server is started or restarted.

   If you enable Cross-Component Trace on the **Runtime** tab, the setting is applied by the system immediately, as long as the server is running.

   Save all runtime changes to the server configuration file
   This field displays on the **Runtime** tab only. Select this option if you want to apply changes you make on the Runtime tab (which are applied by the system immediately) to the configuration.

   If you select **Save all runtime changes to the server configuration file**, the changes you make are applied by the system when the server starts or restarts as well as to the current runtime.

   Enable Cross-Component Trace
   Selecting **Enable Cross-Component Trace** prepares the server for the following:
   - Cross-Component Trace for inbound application-specific call chains
   - Enabling Cross-Component Trace on any module that is listed for **Enable tracing for the selected Service Component Architecture (SCA) modules**.

   Enable Cross-Component Trace in the Configuration collects data when the server starts or restarts.

   Trace all
   Select this option to turn on Cross-Component Trace for the creation of call chain information *for all* SCA modules in the server. Even with **Trace All** selected, you can add additional SCA modules to the table of modules under **Enable tracing for the selected Service Component Architecture (SCA) modules**.

   When you select **Trace all**, the server honors Cross-Component Trace call chains for modules coming from (inbound) other servers. When **Trace all** is selected, the server also checks if Cross-Component Trace is on for any modules in the server and honors those settings so that calls to those modules result in application-specific Cross-Component call chains.

4. Enable Cross-Component Trace for version 6.2.0.x server You can enable Cross-Component Trace on either the **Configuration** tab or the **Runtime** tab.

   If you enable Cross-Component Trace on the **Configuration** tab, the setting is applied by the system whenever the server is started or restarted.

   If you enable Cross-Component Trace on the **Runtime** tab, the setting is applied by the system immediately, as long as the server is running.

   Save all runtime changes to the server configuration file
   This field displays on the **Runtime** tab only. Select this option if you want to apply changes you make on the **Runtime** tab (which are applied by the system immediately) to the configuration.

If you select **Save all runtime changes to the server configuration file** the changes you make are applied by the system when the server starts or restarts.

Enable Cross-Component Trace

Selecting **Enable Cross-Component Trace** turns Cross-Component Trace on for the server.

Enable Cross-Component Trace in the Configuration collects data when the server starts or restarts.

Enable with data snapshot

Selecting **Enable with data snapshot** enables the data snapshot feature of Cross-Component Trace.

When you configure Cross-Component Trace with data snapshot, the generated invocation records contain the invocation input and output data that was passed between the components during processing. The log records associated with the WebSphere Process Server applications hold information about errors or events that occurred during processing and can be used for problem determination using WebSphere Integration Developer.

5. After you have set the parameters to enable Cross-Component Trace, click **OK** to save the settings.

Results

For the server selected, collected data is added to the trace.log file or systemout.log and is purged as those files are purged. See "Troubleshooting Service Component Architecture processing and call chains" on page 275 for more information.

Related concepts

"Cross-Component Trace overview" on page 266
Cross-Component Trace identifies whether a Service Component Architecture (SCA) operation completed successfully. It provides information about what modules were used at run time and what service calls were made. Additionally, you can configure Cross-Component Trace with *data snapshot* to capture data sent in and passed between SCA components.

Related tasks

"Configuring logging for Cross-Component Trace using the administrative console" on page 285
You can use the administrative console to configure Cross-Component Trace logging.

Related information

 Enabling or disabling Cross-Component tracing from within WebSphere Integration Developer

 WebSphere Application Server log levels

Enabling Cross-Component Trace with data snapshot using the administrative console

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.

For an understanding of the data snapshot feature of Cross-Component Trace, see *Cross-Component Trace overview*.

About this task

To enable cross-component tracing, use the following procedure.

Procedure

1. Ensure that the administrative console is running, and then click **Troubleshooting** ▸ **Cross-Component Trace** to display the list of servers. In a network deployment environment, observe the operating status of the selected servers in the **Status** column: **running** or **not running**. If the status of the server is **not running**, runtime tracing for that server is disabled and you can specify only parameters on the **Configuration** tab of the Cross-Component Trace configuration page. In this case, the trace parameters that you set take effect only when the server starts or restarts.

   Note: The **Status** column does not display for stand-alone servers.

2. From the list of servers, select a server on which to enable Cross-Component Trace with data snapshot.

   Clicking the link in the **Server** column takes you to one of two possible Cross-Component Trace configuration pages, depending on the version of the server. Version information for the server displays in the table column labeled **Version**.

3. Enable Cross-Component Trace with data snapshot

   To enable Cross-Component Trace with data snapshot for the server, select the check box for the following fields:

   • **Enable Cross-Component Trace**
   • **Trace all**
   • **Enable data snapshot on this server**

   Note: Clearing the **Enable Cross-Component Trace** option turns off Cross-Component Trace. However, you can still implement a Cross-Component Trace strategy that works with application specific Cross-Component Trace, by building a list of SCA modules to trace (as specified in the **Enable tracing for the selected Service Component Architecture (SCA) modules** table) . Because Enable Cross-Component Trace is cleared, there is no affect on performance. When you are ready to enable Cross-Component Trace, the trace operations will include the modules that you have specified.

   If you enable Cross-Component Trace with data snapshot on the **Configuration** tab, the setting is applied by the system whenever the server is started or restarted.

   If you enable Cross-Component Trace with data snapshot on the **Runtime** tab, the setting is applied by the system immediately, as long as the server is running. If you want to apply changes you make on the **Runtime** tab to the configuration, select the check box for **Save all runtime changes to the server configuration file**.

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 data snapshot 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.

You can delete data snapshot files manually or you can set a parameter on the administrative console to enable a periodic deletion of data snapshot files based on a disk space usage threshold.

See Deleting data snapshot files collected by Cross-component trace and "Troubleshooting Service Component Architecture processing and call chains" on page 275 for more information

Related concepts

"Cross-Component Trace overview" on page 266
Cross-Component Trace identifies whether a Service Component Architecture (SCA) operation completed successfully. It provides information about what modules were used at run time and what service calls were made. Additionally, you can configure Cross-Component Trace with *data snapshot* to capture data sent in and passed between SCA components.

Enabling Cross-Component Trace levels using the command line

You can set Cross-Component Trace levels and determine the current trace level settings using the wsadmin command. Setting Cross-Component Trace level involves specifying parameters that determine whether to enable Cross-Component Trace, and if enabled, what type of trace operations are to be performed.

Before you begin

For a description of Cross-Component Trace, see Cross-component trace overview.

About this task

There are several options that you can choose when deciding how to implement Cross-Component Trace.

This task describes how to use the wsadmin command to determine (getLevel) current trace settings and how to set (setLevel) Cross-Component Trace options.

Procedure

1. **To determine the current trace settings, perform the following steps**
   a. Open a command window.
      The wsadmin command can be found at the `<WPS>/profiles/<dmgr profile>/bin` directory or the `<WPS>>/bin` directory.
   b. At the command prompt, enter the wsadmin command to enter the wsadmin environment.
   c. Query the Cross-component trace MBean:
      ```
      set xctBean [$AdminControl queryNames *:*,type=XCTMBean]
      ```
   d. Get the current Cross-Component Trace level:
      ```
      $AdminControl invoke $xctBean getLevel
      ```

2. **To set cross component trace values, perform the following steps:**
   a. Open a command window.

      The wsadmin command can be found at the <WPS>/profiles/<dmgr profile>/bin directory or the <WPS>>/bin directory.
   b. At the command prompt, enter the wsadmin command to enter the wsadmin environment.
   c. Query the Cross-component trace MBean:

      ```
      set xctBean [$AdminControl queryNames *:*,type=XCTMBean]
      ```
   d. Set the level of Cross-Component Trace using one of the following values:
      - disable

        Enter this value to turn off cross component trace. When you disable Cross-Component Trace, the system behaves as if Cross-Component Trace does not exist at all. There is no system checking associated with trace activities and no affect on performance.
      - READY

        Enter this value to turn on Cross-Component Trace with no other values set, or with only trace settings for specific modules. When in the READY state, the server honors application-specific Cross-Component Trace call chains coming from other servers. In the READY state, the server also checks if application-specific trace is on for any modules in the server and honors those settings so that calls to those modules result in application-specific cross-component call chains starting.

        If you were enabling Cross-Component Trace levels from the administrative console, the READY state would be configured as follows:
        - **Enable Cross-Component Trace** = checked
        - **Trace all** = not checked
        - **Enable data snapshot on this server** = not checked

        The **Enable tracing for the selected Service Component Architecture (SCA) modules** table on the administrative console may or may not have values in it for the READY state.
      - enable

        Enter this value to turn on Cross-Component Trace.
      - "enable with data snapshot"

        Enter this value to turn on Cross-Component Trace with data snapshot capture functionality.

      See the sample commands in the **Example** section.

Example

The following command disables Cross-Component Trace:

```
wsadmin -connType SOAP -host myDmgr -port 8879 -user dmgradmin -password dmgrpass
> set xctBean [$AdminControl queryNames *:*,type=XCTMBean]
> $AdminControl invoke $xctBean setLevel disable
```

The following command enables Cross-Component Trace:

```
wsadmin -connType SOAP -host myDmgr -port 8879 -user dmgradmin -password dmgrpass
> set xctBean [$AdminControl queryNames *:*,type=XCTMBean]
> $AdminControl invoke $xctBean setLevel enable
```

The following command enables Cross-Component Trace with data snapshot:

```
wsadmin -connType SOAP -host myDmgr -port 8879 -user dmgradmin -password dmgrpass
> set xctBean [$AdminControl queryNames *:*,type=XCTMBean]
> $AdminControl invoke $xctBean setLevel {"enable with data snapshot"}
```

The following command sets Cross-Component Trace level to "READY":

```
wsadmin -connType SOAP -host myDmgr -port 8879 -user dmgradmin -password dmgrpass
> set xctBean [$AdminControl queryNames *:*,type=XCTMBean]
> $AdminControl invoke $xctBean setLevel READY
```

Related concepts

"Cross-Component Trace overview" on page 266
Cross-Component Trace identifies whether a Service Component Architecture
(SCA) operation completed successfully. It provides information about what
modules were used at run time and what service calls were made. Additionally,
you can configure Cross-Component Trace with *data snapshot* to capture data sent
in and passed between SCA components.

"Cross-Component Trace and system performance" on page 272
The runtime processing activities associated with Cross-Component Trace can affect
system performance.

Configuring logging for Cross-Component Trace using the administrative console

You can use the administrative console to configure Cross-Component Trace
logging.

Before you begin

You must be logged in as administrator to perform this task.

About this task

Cross-Component trace uses the WebSphere Application Server logging service to
capture information and to provide context for the other entries in the logs. The
information pertaining to SCA component processing (including the sequential
flow of the event) is captured by Cross-Component Trace and written to either the
`systemout.log` or `trace.log` file according to well defined rules. For detailed
information on logging rules, see *Tracing and logging configuration* in the WebSphere
Application Server information Center.

You can select which log file Cross-Component Trace records are written to (either
`trace.log` or `systemout.log`).

This task describes how to use the administrative console to configure
Cross-Component Trace logging.

For information about how to view data captured by the Cross-Component Trace
feature, see *Server Logs view* in the WebSphere Integration Developer information
center.

Procedure

1. Ensure that the administrative console is running, and then click
   Troubleshooting ⏵ **Cross-Component Trace** to display the list of servers. In a
   network deployment environment, observe the operating status of the selected
   servers in the **Status** column: **running** or **not running**. If the status of the
   server is **not running**, runtime tracing for that server is disabled and you can
   specify only parameters on the **Configuration** tab of the Cross-Component

Trace configuration page. In this case, the trace parameters that you set take effect only when the server starts or restarts.

Note: The **Status** column does not display for stand-alone servers.

2. From the list of servers, select a server on which to configure the logging for Cross-Component Trace.

   Clicking the **Server Name** takes you to the Cross-Component Trace configuration page.

   Note: Configuring the logging for Cross-Component Trace is supported on version 7.0 servers. You cannot configure logging for Cross-Component Trace on version 6.2.0.x or earlier versions of the server.

3. Select the logging option in **Save cross-component trace output to** Available options include:

   - **trace**

     The default and recommended option is **trace**, which will result in Cross-Component Trace data being written to the `trace.log` file. This maps to the WebSphere Application Server logging level of *Fine*. Selecting **trace** provides the best performance for collecting Cross-Component Trace data.

     Anything written to the SystemOut.log is also written to the trace.log. So, if other tracing is on, there is a lot more data to look at.

   - **system**

     This option maps to the WebSphere Application Server logging level of *Info*, which will result in Cross-Component Trace data being written to the `systemout.log` file, and potentially `trace.log` if other trace settings are enabled.

     Choosing **system** takes the system more time to write out the log.

     The SystemOut.log has less data in it, so it may be easier to look through.

4. Click **OK** to apply Cross-Component Trace logging configuration changes to the server.

Related concepts

"Cross-Component Trace overview" on page 266
Cross-Component Trace identifies whether a Service Component Architecture (SCA) operation completed successfully. It provides information about what modules were used at run time and what service calls were made. Additionally, you can configure Cross-Component Trace with *data snapshot* to capture data sent in and passed between SCA components.

"Cross-Component Trace and system performance" on page 272
The runtime processing activities associated with Cross-Component Trace can affect system performance.

Related tasks

"Enabling Cross-Component Trace using the administrative console" on page 279
You can use the administrative console to prepare the server for Cross-Component Trace operations.

Related information

WebSphere Application Server: Tracing and logging configuration

Configuring logging for Cross-Component Trace using the command line

You can configure Cross-Component Trace logging using the wsadmin command.

Before you begin

You must be logged in as administrator to perform this task.

About this task

This task describes how to use wsadmin to configure Cross-Component Trace logging.

Note: Configuring the logging for Cross-Component Trace is supported on version 7.0 servers. You cannot configure logging for Cross-Component Trace on version 6.2.0.x or earlier versions of the server.

The logging levels that you set determine how Cross-Component Trace records are written.

Cross-Component Trace uses the WebSphere Application Server logging service to capture information and to provide context for the other entries in the logs. The information pertaining to SCA component processing (including the sequential flow of the event) is captured by Cross-Component Trace and written to either the `SystemOut.log` or `trace.log` file according to well defined rules. For detailed information on logging rules, see *Tracing and logging configuration* in the WebSphere Application Server information Center.

For information about how to view data captured by the Cross-Component Trace feature, see *Server Logs view* in the WebSphere Integration Developer information center.

Procedure
1. **To determine the current Cross-Component Trace logging settings, perform the following steps**
   a. Open a command window.
      The wsadmin command can be found at the `<WPS>/profiles/<dmgr profile>/bin` directory or the `<WPS>>/bin` directory.
   b. At the command prompt, enter the wsadmin command to enter the wsadmin environment.
   c. Query the Cross-component trace MBean:
      `set xctBean [$AdminControl queryNames *:*,type=XCTMBean]`
   d. Get the current Cross-Component Trace logging settings:
      `$AdminControl invoke $xctBean getConfigTraceLevel`
2. **To set Cross-Component Trace logging values, perform the following steps:**
   a. Open a command window.
      The wsadmin command can be found at the `<WPS>/profiles/<dmgr profile>/bin` directory or the `<WPS>>/bin` directory.
   b. At the command prompt, enter the wsadmin command to enter the wsadmin environment.
   c. Query the Cross-component trace MBean:
      `set xctBean [$AdminControl queryNames *:*,type=XCTMBean]`
   d. Call the `setConfigTraceLevel` command on the MBean:
      `$AdminControl invoke $xctBean setConfigTraceLevel` *level*
      Where *level* is set to either `FINE` or `INFO`.
      The default logging level for production environments is FINE.

Setting the `setConfigTraceLevel` to **FINE**, results in Cross-Component Trace data being written to the `trace.log` file. Setting the `setConfigTraceLevel` to **FINE** provides the best performance for collecting Cross-Component Trace data.

Setting the `setConfigTraceLevel` to **INFO** results in Cross-Component Trace data being written to the `SystemOut.log` file, and potentially `trace.log` if other trace settings are enabled.

Setting `setConfigTraceLevel` to **INFO** takes the system more time to write out the log.

The `SystemOut.log` has less data in it, so it may be easier to look through.

Note: Any value above FINE, (such as ALL, FINER or FINEST) will redirect to `trace.log`.

Note: WebSphere Integration Developer resets the logging level in the unit-test environment (UTE) to `SystemOut.log`. WebSphere Integration Developer will also set the logging level to `SystemOut.log` for any remote server that it accesses as WebSphere Integration Developer can only access `SystemOut.log` files, not `trace.log` files from a remote machine.

See the sample commands in the **Example** section.

Example

The following command is an example of setting the Cross-Component logging level to write to the `trace.log` file:

```
wsadmin -connType SOAP -host myDmgr -port 8879 -user dmgradmin -password dmgrpass
> set xctBean [$AdminControl queryNames *:*,type=XCTResourcesMBean]
> $AdminControl invoke $xctBean setConfigTraceLevel FINE
```

The following command is an example of setting the Cross-Component logging level to write to the `SystemOut.log` file:

```
wsadmin -connType SOAP -host myDmgr -port 8879 -user dmgradmin -password dmgrpass
> set xctBean [$AdminControl queryNames *:*,type=XCTResourcesMBean]
> $AdminControl invoke $xctBean setConfigTraceLevel INFO
```

Related information

WebSphere Application Server: Tracing and logging configuration

Enabling Cross-Component Trace for selected SCA modules

You can enable Cross-Component Trace on specific Service Component Architecture (SCA) modules on a server. Use Cross-Component Trace to identify `trace.log` and `systemout.log` data that is associated with WebSphere Process Server and WebSphere Enterprise Service Bus modules and components.

Choosing to run Cross-Component Trace on SCA modules is known as *application-specific Cross-Component Trace*.

Related concepts

"Application-specific Cross-Component Trace" on page 269
Application-specific Cross-Component Trace provides the same functionality as
Cross-Component Trace, but applies the functionality to a single Service
Component Architecture (SCA) module.

Enabling application-specific Cross-Component Trace using the administrative console

You can use the administrative console to choose specific SCA modules on which
to run Cross-Component Trace.

Before you begin

You must be logged in as administrator to perform this task.

About this task

This task describes how to use the administrative console to enable
application-specific Cross-Component Trace by adding selected service component
architecture (SCA) modules to the table of SCA modules.

Application-specific Cross-Component Trace is best used when there are a small
number of SCA modules on which you want to enable Cross-Component Trace.

Attention: Try to keep the list of SCA modules in **Enable tracing for the selected
Service Component Architecture (SCA) modules** to a small number. If the number
of modules listed in the table begins to grow, consider selecting **Trace all** instead.
There is a small affect on performance for each SCA module added to the list.

Procedure

1. Ensure that the administrative console is running, and then click
   Troubleshooting ▸ Cross-Component Trace to display the list of servers. In a
   network deployment environment, observe the operating status of the selected
   servers in the **Status** column: **running** or **not running**. If the status of the
   server is **not running**, runtime tracing for that server is disabled and you can
   specify only parameters on the **Configuration** tab of the Cross-Component
   Trace configuration page. In this case, the trace parameters that you set take
   effect only when the server starts or restarts.

   Note: The **Status** column does not display for stand-alone servers.
2. From the list of servers, select the server hosting the modules on which you
   want to enable Cross-Component Trace.

   Note: Application-specific Cross-Component Trace is supported on version 7.0.
   You cannot enable Application-specific Cross-Component Trace on version
   6.2.0.x or earlier versions of the server.

   Clicking the **Server Name** takes you to the server's Cross-Component Trace
   configuration page.

   The parameters that you set on the **Configuration** tab are applied by the
   system when the server is started or restarted. The parameters that you set on
   the **Runtime** tab are applied by the system immediately.
3. Go to **Enable tracing for the selected Service Component Architecture (SCA)
   modules** for a list of modules for which Cross-Component Trace has been
   enabled.

4. Click **Add** to go to a list of all the SCA modules on the server.

   From the list of SCA modules, choose the SCA module on which you want to enable Cross-Component Trace.

5. Click **OK** to return to the server's Cross-Component Trace configuration page.

Results

The module that you selected should now display in the list of SCA modules.

Related concepts

"Application-specific Cross-Component Trace" on page 269
Application-specific Cross-Component Trace provides the same functionality as Cross-Component Trace, but applies the functionality to a single Service Component Architecture (SCA) module.

Enabling application-specific Cross-Component Trace with data snapshot using the administrative console

You can use the administrative console to enable application-specific Cross-Component Trace with data snapshot.

Before you begin

You must be logged in as administrator to perform this task.

About this task

This task describes how to use the administrative console to enable application-specific Cross-Component Trace with data snapshot.

Application-specific Cross-Component Trace with data snapshot is best used when there are a small number of SCA modules on which you want to enable Cross-Component Trace with data snapshot.

In situations where you want to run Cross-Component Trace on most or on all modules, but require the data snapshot feature for only a few, consider setting the **Trace all** parameter and then using the **Enable tracing for the selected Service Component Architecture (SCA) modules** table to enable data snapshot on only those modules that require it.

Attention: Try to keep the list of SCA modules in **Enable tracing for the selected Service Component Architecture (SCA) modules** to a small number. If the number of modules listed in the table begins to grow, consider selecting **Trace all** instead. There is a small affect on performance for each SCA module added to the list.

Attention: Implementing the data snapshot feature of Cross-Component Trace affects performance and disk space usage. You should keep this in mind when using Cross-Component Trace on production systems.

Procedure

1. Ensure that the administrative console is running, and then click **Troubleshooting** ▸ **Cross-Component Trace** to display the list of servers. In a network deployment environment, observe the operating status of the selected servers in the **Status** column: **running** or **not running**. If the status of the server is **not running**, runtime tracing for that server is disabled and you can specify only parameters on the **Configuration** tab of the Cross-Component

Trace configuration page. In this case, the trace parameters that you set take effect only when the server starts or restarts.

Note: The **Status** column does not display for stand-alone servers.

2. From the list of servers, select the server hosting the modules on which you want to enable Cross-Component Trace.

Note: Application-specific Cross-Component Trace is supported on version 7.0. You cannot enable Application-specific Cross-Component Trace on version 6.2.0.x or earlier versions of the server.

Clicking the **Server Name** takes you to the server's Cross-Component Trace configuration page.

The parameters that you set on the **Configuration** tab are applied by the system when the server is started or restarted. The parameters that you set on the **Runtime** tab are applied by the system immediately.

3. Go to **Enable tracing for the selected Service Component Architecture (SCA) modules** for a list of modules for which Cross-Component Trace has been enabled.

4. From the list of modules, set **Enable with data snapshot** for the modules on which you want to collect data snapshot files and click **OK**. You are returned to the configuration page.

If the module on which you want to enable Cross-Component Trace is not in the table, select **Add** to add the module and then select **Enable with data snapshot**.

Note: You can also disable Cross-Component Trace on SCA modules by removing the module from the table.

5. Click **Apply** .

Results

The selected modules are marked for Cross-Component Trace with data snapshot.

Related tasks

"Disabling Cross-Component Trace on SCA modules" on page 295
You can use the administrative console to disable Cross-Component Trace on Service Component Architecture (SCA) modules.

Enabling application-specific cross-component trace using the command line

You can set application-specific cross-component trace levels and determine the current trace level settings using the wsadmin command. Setting application-specific cross-component trace level involves specifying parameters that determine if application-specific cross-component trace is enabled, and if so, what type of trace operations are to be performed.

Before you begin

For a description of application-specific cross-component trace, see Application-specific cross-component trace.

About this task

Application-specific cross-component trace provides the same functionality as cross-component trace, but applies the functionality to a single application. When

you apply cross-component trace to a single application (rather than to an entire server) you reduced performance and resource impact to the server and the data capture is targeted to a single application, making the data analysis easier and less time consuming.

This task describes how to use the wsadmin command to enable application-specific cross-component trace.

Procedure

1. **To get a list of the names of modules that have trace levels set and to determine the trace level for a particular module, perform the following steps:**

   Note: The getModules command returns only the name of modules for which the trace level has already been set.

   a. Open a command window.

   The wsadmin command can be found at the <WPS>/profiles/<dmgr profile>/bin directory or the <WPS>>/bin directory.

   b. At the command prompt, enter the wsadmin command to enter the wsadmin environment.

   c. Query the Cross-component trace MBean:

   `set xctBean [$AdminControl queryNames *:*,type=XCTMBean]`

   d. Get the modules for which application-specific trace levels have already been set:

   `$AdminControl invoke $xctBean getModules`

   The result is a list of module names.

   e. To determine the current application-specific trace level for a particular module, perform the following steps:

   1) Open a command window.

   The wsadmin command can be found at 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) Query the Cross-component trace MBean:

   `set xctBean [$AdminControl queryNames *:*,type=XCTMBean]`

   4) Get the current application-specific trace level for a particular module:

   Note: You need to know the name of the module before invoking this command.

   `$AdminControl invoke $xctBean getModuleLevel moduleName`

   where moduleName is the name of the module.

2. **To set the current application-specific trace level for a particular module, perform the following steps:**

   Note: It is advisable to always query by running a getModuleLevel to make sure that the value was set correctly. An invalid value sets the level to *disable*.

   a. Open a command window.

   The wsadmin command can be found at the <WPS>/profiles/<dmgr profile>/bin directory or the <WPS>>/bin directory.

b. At the command prompt, enter the wsadmin command to enter the wsadmin environment.

c. Query the Cross-component trace MBean:

```
set xctBean [$AdminControl queryNames *:*,type=XCTMBean]
```

d. Set the current application-specific trace level for a particular module:

```
set args [list level moduleName]
$AdminControl invoke $xctBean setModuleLevel $args
```

where level is one of the levels specify the trace level as follows:

- disable

  Enter this value to turn off cross component trace. When you disable cross-component trace, the system behaves as if cross-component trace does not exist at all. There is no system checking associated with trace activities and no affect on performance.

- enable

  Enter this value to turn on cross-component trace.

- "enable with data snapshot"

  Enter this value to turn on cross-component trace with data snapshot capture functionality.

  Note: The value READY is not valid for modules. READY can only be set at the root level, where moduleName is the name of the module. Refer to the procedure on setting cross component trace values in Enabling cross-component trace levels using the command line for information about how root-level cross-component trace settings affect application-specific trace settings behavior.

3. **To remove a module from inclusion in application-specific tracing:**

   a. Open a command window.

   The wsadmin command can be found at the <WPS>/profiles/<dmgr profile>/bin directory or the <WPS>>/bin directory.

   b. At the command prompt, enter the wsadmin command to enter the wsadmin environment.

   c. Query the Cross-component trace MBean:

   ```
   set xctBean [$AdminControl queryNames *:*,type=XCTMBean]
   ```

   d. Remove a module from application-specific tracing:

   Note: You need to know the name of the module before invoking this command.

   ```
   $AdminControl invoke $xctBean removeModule moduleName
   ```

   where moduleName is the name of the module.

Example

The following command disables cross-component trace on the SCA module named HomeLoanApp:

```
wsadmin -connType SOAP -host myDmgr -port 8879 -user dmgradmin -password dmgrpass
> set xctBean [$AdminControl queryNames *:*,type=XCTMBean]
> set args [list disable HomeLoanApp]
> $AdminControl invoke $xctBean setModuleLevel $args
```

The following command enables cross-component trace on the SCA module named HomeLoanApp:

```
wsadmin -connType SOAP -host myDmgr -port 8879 -user dmgradmin -password dmgrpass
> set xctBean [$AdminControl queryNames *:*,type=XCTMBean]
> set args [list enable HomeLoanApp]
> $AdminControl invoke $xctBean setModuleLevel $args
```

The following command enables cross-component trace with data snapshot on the SCA module named HomeLoanApp:

```
wsadmin -connType SOAP -host myDmgr -port 8879 -user dmgradmin -password dmgrpass
> set xctBean [$AdminControl queryNames *:*,type=XCTMBean]
> set args [list {enable with data snapshot} HomeLoanApp]
> $AdminControl invoke $xctBean setModuleLevel $args
```

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.

You can also disable Cross-Component Trace using wsadmin. For information on using wsadmin to disable Cross-Component Trace, see information on setting level of Cross-Component Trace in *Enabling Cross-Component Trace levels using the command line*.

About this task

To disable Cross-Component tracing, use the following procedure.

Procedure

1. Ensure that the administrative console is running, and then click **Troubleshooting** → **Cross-Component Trace** to display the list of servers. Observe the operating status of the selected servers in the **Status** column: **running** or **not running**. If the status of the server is **not running**, runtime tracing for that server is disabled and you can specify only parameters on the **Configuration** tab of the Cross-Component Trace configuration page. In this case, the trace parameters that you set take effect only when the server starts or restarts.

2. From the list of servers, select the server on which to disable Cross-Component Trace.

   Clicking the **Server Name** takes you to one of two possible Cross-Component Trace configuration pages, depending on the version of the server. Version information for the server displays in the table column labeled **Version**.

3. Disable Cross-Component Trace

   From the Cross-Component Trace configuration page, you can choose to disable Cross-Component Trace from the **Configuration** tab or the **Runtime** tab. Settings made to the **Configuration** tab are applied by the system whenever the server is started or restarted. The parameters that you set on the **Runtime** tab are applied by the system immediately, as long as the server is running. To disable Cross-Component Trace clear the **Enable Cross-Component Trace** check box.

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.

Disabling Cross-Component Trace on SCA modules

You can use the administrative console to disable Cross-Component Trace on Service Component Architecture (SCA) modules.

Before you begin

You must be logged in as administrator to perform this task.

About this task

This task describes how to disable Cross-Component Trace on SCA modules using the administrative console.

Keeping the list of SCA modules to a minimum may help system performance.

Procedure

1. Ensure that the administrative console is running, and then click **Troubleshooting** → **Cross-Component Trace** to display the list of servers. In a network deployment environment, observe the operating status of the selected servers in the **Status** column: **running** or **not running**. If the status of the server is **not running**, runtime tracing for that server is disabled and you can specify only parameters on the **Configuration** tab of the Cross-Component Trace configuration page. In this case, the trace parameters that you set take effect only when the server starts or restarts.

   Note: The **Status** column does not display for stand-alone servers.

2. From the list of servers, select the server hosting the modules on which you want to disable Cross-Component Trace.

   Note: Application-specific Cross-Component Trace is supported on version 7.0 servers only.

   Clicking the **Server Name** takes you to the server's Cross-Component Trace configuration page.

   The parameters that you set on the **Configuration** tab are applied by the system when the server is started or restarted. The parameters that you set on the **Runtime** tab are applied by the system immediately.

3. Go to **Enable tracing for the selected Service Component Architecture (SCA) modules** for a list of modules for which Cross-Component Trace has been enabled.

4. Choose the **Select** check box of the module that you want to remove.

5. Click **Remove**

Results

The SCA module is removed from the list and Cross-Component Trace on that module has been disabled.

Related tasks

"Enabling application-specific Cross-Component Trace with data snapshot using the administrative console" on page 290
You can use the administrative console to enable application-specific Cross-Component Trace with data snapshot.

Deleting data snapshot files

You can delete the data snapshot files from the `logs\XCT` directory to free up disk space. 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. When the data from the systemout.log and trace.log is deleted, the corresponding data snapshot files remain on the disk. You can delete these files manually or you can set a parameter on the administrative console which results in data snapshot files being deleted by the system periodically.

Related concepts

"Troubleshooting Service Component Architecture processing and call chains" on page 275
Cross-Component Trace identifies whether a Service Component Architecture (SCA) operation completed successfully. It 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 log records associated with the WebSphere® ESB applications hold information about errors or events that occurred during processing and can be used for problem determination using WebSphere Integration Developer.

Setting disk space capacity for data snapshot file deletion using the administrative console

You can use the administrative console to set a disk space capacity for data snapshot files. The capacity that you set is used by the system to delete data snapshot files automatically.

Before you begin

Before setting the disk space capacity, check to see how much space is available on the disk.

About this task

This task describes the procedure for setting a disk capacity to be used as a mechanism for automatically deleting data snapshot files.

When the disk use capacity is reached, the system deletes data snapshot files from the disk automatically.

Procedure

1. From the administrative console, navigate to the Cross-Component Trace page. **Troubleshooting** → **Component Trace** → *server_name* → **Cross-component Trace disk use**.

2. Set **Cross-Component Trace disk use** parameters. Field information is as follows:

   - **In use**

     This field lists the current disk space consumed by Cross-Component trace snapshot files

   - **Total disk space allowed for this server**

Enter the disk space use allowed in megabytes. The value you enter represents a threshold. When the threshold is exceeded, the system automatically deletes data snapshot files from the disk. The files are deleted in a sequential fashion, from oldest to most recent, until the system achieves the disk space required to write the next data snapshot, while remaining below the disk use threshold.

There is a minimum level of 50 MB. Settings below this minimum are converted to the minimum. Setting of 0 or less than 0 (-1, -2...) results in 0 and turns off the automatic delete feature.

Setting disk space capacity for data snapshot files using the command line

You can use wsadmin to query and set a disk space capacity to be used as a trigger for automatically deleting data snapshot files that are captured as a result of Cross-Component Trace operations.

Before you begin

For a description of Cross-Component Trace, see Cross-component trace overview.

About this task

This task describes how to use the wsadmin command to set a disk space capacity as a threshold for initiating an automated clean-up operation of the disk holding the data snapshot files that are captured as a result of Cross-Component Trace operations.

The system deletes the data snapshot files periodically based on a disk space use parameter.

Procedure

1. **To determine the maximum allowed disk space capacity, perform the following steps:**
   a. Open a command window.

   The wsadmin command can be found at the <WPS>/profiles/<dmgr profile>/bin directory or the <WPS>>/bin directory.
   b. At the command prompt, enter the wsadmin command to enter the wsadmin environment.
   c. Query the Cross-component trace Resources MBean:

   `set xctBean [$AdminControl queryNames *:*,type=XCTResourcesMBean]`
   d. Call the `getSnapshotCapacity` command on the MBean

   `$AdminControl invoke $xctBean getSnapshotCapacity`

2. **To set disk space capacity, perform the following steps:**
   a. At the command prompt, enter the wsadmin command to enter the wsadmin environment.

   The wsadmin command can be found at the <WPS>/profiles/<dmgr profile>/bin directory or the <WPS>>/bin directory.
   b. At the command prompt, enter the wsadmin command to enter the wsadmin environment.
   c. Query the Cross-component trace Resources MBean:

   `set xctBean [$AdminControl queryNames *:*,type=XCTResourcesMBean]`
   d. Call the `setSnapshotCapacity` command on the MBean:

```
$AdminControl invoke $xctBean setSnapshotCapacity limit
```
Where *limit* is the disk space use allowed in megabytes.

Note: There is a minimum level of 50 MB. Settings below this minimum are converted to the minimum. Setting of 0 or less than 0 (-1, -2...) results in 0 and turns off the automatic delete feature.

The value you enter represents a threshold. When the threshold is exceeded, the system automatically deletes data snapshot files from the disk. The files are deleted in a sequential fashion, from oldest to most recent, until the system achieves the disk space required to write the next data snapshot, while remaining below the disk use threshold.

Example

The following command is an example of setting the disk capacity for data snapshot files at 500 MB:

```
wsadmin -connType SOAP -host myDmgr -port 8879 -user dmgradmin -password dmgrpass
> set xctBean [$AdminControl queryNames *:*,type=XCTResourcesMBean]
> $AdminControl invoke $xctBean setSnapshotCapacity 500
```

Deleting data snapshot files manually

You can delete data snapshot files manually from the `logs\XCT` directory.

Before you begin

You must have read and write access to the logs directories of each server.

In addition to being able to delete data snapshot files manually, you can also delete data snapshot files using the following methods:

1. Letting the system delete the data snapshot files automatically (**recommended method**).

   You can configure an automated data snapshot file cleanup from the administrative console. With this method, the system deletes the data snapshot files based on a user-defined disk space usage.

   The value you enter represents a threshold. When the threshold is exceeded, the system automatically deletes data snapshot files from the disk. The files are deleted in a sequential fashion, from oldest to most recent, until the system achieves the disk space required to write the next data snapshot, while remaining below the disk use threshold.

   If you have enabled the automatic data snapshot file delete process from the administrative console, there is no need to delete data snapshot files manually.

2. Selecting **Delete data snapshot files** from the administrative console or running the equivalent wsadmin command to remove the data snapshot files.

If you choose to delete data snapshot files manually (as described in this topic), make sure you have disabled the automatic data snapshot file cleanup from the administrative console, **or** that you have shut down the server. If you have disabled data snapshot file cleanup from the administrative console, it is OK to delete the files manually and you do not need to shut down the server.

It is acceptable to turn off the automated data snapshot file cleanup from the administrative console, delete some data snapshot files manually, and then go back into the console and turn on automated data snapshot file cleanup. You might also decide to lower the threshold on disk space usage limit, deleting even more files automatically.

Note: Deleting data snapshot files manually in combination with the automatic data snapshot file deletion can result in inconsistency in the cross-component trace resource manager (such as disk space calculation or cached files for deletion).

About this task

The following steps describe how to manually delete from the disk data snapshot files collected by cross-component tracing.

In a network deployment environment with multiple nodes and servers defined on one machine, data snapshot files are written to a *server-specific* subdirectory. This allows you to identify, and delete the data snapshot files associated with each server. Cross-Component Trace data snapshot files are written to the following directory structure:

```
logs\
     server
     ffdc
     xct\
       server-specific_dir\
                           2009-0-25-11
                           2009-0-26-12
                           2009-0-26-14
```

Where `server-specific_dir` name is derived from the name of the server. For example, **server1** is the default server name for a stand-alone installation.

Procedure

1. Go to the `logs\XCT\` *server-specific* 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` *server-specific* directory.

Results

The captured data is deleted.

Delete data snapshot files using the administrative console

You can delete data snapshot files using the administrative console.

Before you begin

In addition to being able to delete data snapshot files by using the administrative console, you can also delete data snapshot files by using the following methods:

1. Letting the system delete the data snapshot files automatically (**recommended method**).

   You can configure an automated data snapshot file cleanup from the administrative console. With this method, the system deletes the data snapshot files based on a user-defined disk space usage.

   The value you enter represents a threshold. When the threshold is exceeded, the system automatically deletes data snapshot files from the disk. The files are deleted in a sequential fashion, from oldest to most recent, until the system achieves the disk space required to write the next data snapshot, while remaining below the disk use threshold.

If you have configured Cross-Component Trace to delete data snapshot files automatically from the administrative console, there is no need to delete data snapshot files manually.

Note: Although automated data snapshot file cleanup is the recommended method for deleting data snapshot files, using the **Delete data snapshot files** (as described in this topic) is also a good practice, even when you have configured the system to delete data snapshot files automatically. The automated data snapshot file cleanup is initiated only when disk use threshold is reached. If the data snapshot feature is turned off for a period of time, existing data snapshot files remain in the system, taking up disk space. So, using the **Delete data snapshot files** periodically makes sense.

2. Deleting data snapshot files manually from the from the `logs\XCT` directory or by running the `clearSnapshot` wsadmin command.

About this task

This task describes the procedure for deleting data snapshot files using the administrative console.

Cross-Component Trace data snapshot files are written to the following directory structure:

```
logs\
    server
    ffdc
    xct\
       server-specific_dir\
                         2009-0-25-11
                         2009-0-26-12
                         2009-0-26-14
```

Procedure

1. From the administrative console, navigate to the cross-component trace page. **Troubleshooting** → **Component Trace** → *server_name* → **Cross-component Trace disk use**.
2. Click **Delete data snapshot files** to remove the data snapshot files from the disk.

Deleting data snapshot files using the command line

You can determine the current disk space capacity and delete data snapshot files using the wsadmin command.

Before you begin

For a description of application-specific cross-component trace, see Cross-component trace overview.

About this task

The data files generated as a result of enabling cross-component trace with data snapshot, have a one-to-one relationship with the data added to the `systemout.log` and `trace.log` files. When WebSphere Application Server deletes old systemout.log and trace.log files, the result is orphaned data snapshot files that need to be deleted.

This task describes how to use the wsadmin command to retrieve the current disk usage and to delete data snapshot files from the `logs\XCT` directory. There is administrative console-equivalent to this command.

Deleting data snapshot files using the command line, or from the administrative console is a more preferable method than deleting the data snapshot files manually. However, the recommended method for deleting data snapshot shot files is to configure an automated data snapshot file cleanup from the administrative console.

Deleting data snapshot files using the command line deletes the XCT logs for a server. If there are multiple servers, run this command for each of the servers.

Note: In a network deployment environment with multiple nodes and servers defined on one machine, data snapshot files are written to a *server-specific* subdirectory. This allows you to identify, and delete the data snapshot files associated with each server.

Procedure

1. **To determine the current disk space usage, perform the following steps**
   a. Open a command window.

      The wsadmin command can be found at the `<WPS>/profiles/<dmgr profile>/bin` directory or the `<WPS>>/bin` directory.
   b. At the command prompt, enter the wsadmin command to enter the wsadmin environment.
   c. Query the Cross-component trace Resources MBean

      `set xctBean [$AdminControl queryNames *:*,type=XCTResourcesMBean]`
   d. Call the `getSnapshotSize` command on the MBean

      `$AdminControl invoke $xctBean getSnapshotSize`
2. **To clear disk space usage, perform the following steps:**
   a. Open a command window.

      The wsadmin command can be found at the `<WPS>/profiles/<dmgr profile>/bin` directory or the `<WPS>>/bin` directory.
   b. At the command prompt, enter the wsadmin command to enter the wsadmin environment.
   c. Query the Cross-component trace Resources MBean

      `set xctBean [$AdminControl queryNames *:*,type=XCTResourcesMBean]`
   d. Invoke the `clearSnapshot` command on the MBean:

      `$AdminControl invoke $xctBean clearSnapshot`

Example

This example illustrates how to use a wsadmin command to delete data snapshot files from the `logs\XCT` directory.

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
> set xctBean [$AdminControl queryNames *:*,type=XCTResourcesMBean]
> $AdminControl invoke $xctBean clearSnapshot
```

This example illustrates how to use a wsadmin command to determine the current disk space usage:

```
wsadmin -connType SOAP -host myDmgr -port 8879 -user dmgradmin -password dmgrpass
> set xctBean [$AdminControl queryNames *:*,type=XCTResourcesMBean]
> $AdminControl invoke $xctBean getSnapshotSize
```

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 `AMQERRxx.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 message listener 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 is 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.

- Service timeout runtime exception caused by late response when using the temporary dynamic response destination correlation scheme.

  The JMS import will timeout after a period of time determined by the SCA response expiration qualifier, or if this is not set it will default to 60 seconds.

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 binding.

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.

Generic JMS-based SCA messages not appearing in the failed event manager

If SCA messages originated through a generic 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 generic JMS bindings.

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. 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 Java EE 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 Java EE and non-Java EE 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.

MQ-based SCA messages not appearing in the failed event manager

If SCA messages originated because of a WebSphere MQ interaction failure, you would expect to find these messages in the failed event manager. If these messages are not showing in the failed event manager, check that the underlying WebSphere MQ destination has a maximum failed deliveries value greater than 1. Setting this value to 2 or more allows interaction with the failed event manager during SCA invocations for the WebSphere MQ bindings.

MQ failed events are replayed to the wrong queue manager

When a predefined connection factory is to be used for outbound connections, the connection properties must match those defined in the activation specification used for inbound connections.

The predefined connection factory is used to create a connection when replaying a failed event and must therefore be configured to use the same queue manager from which the message was originally received.

Troubleshooting a failed deployment

Use the information in this group of topics to identify and resolve errors in your deployment environment.

Deleting JCA activation specifications

The system builds JCA 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 JCA 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 JCA 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 JCA 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 JCA activation specifications related to the module that created these destinations.

IBM ®

Printed in USA