Enhanced Connector Applications SupportPac VP01 for IBM WebSphere Business Events 3.0.0

Third edition (May 2012).

Note:

Before using this information and the product it supports, read the information in “Notices” on page 80.

Third edition (May 2012).

Introduction to the SupportPac

The Enhanced Connector Applications SupportPac contains enhanced implementations for the File System, HTTP, JDBC (RDBMS), JMS (Message Queue), and SOAP connectors provided with WebSphere Business Events V7.0.1. These connectors offer improved performance and scalability.

Overview

The connectors provided with WebSphere Business Events V7.0.1 run as a stand-alone application, outside of the application server environment as shown in Figure 1:

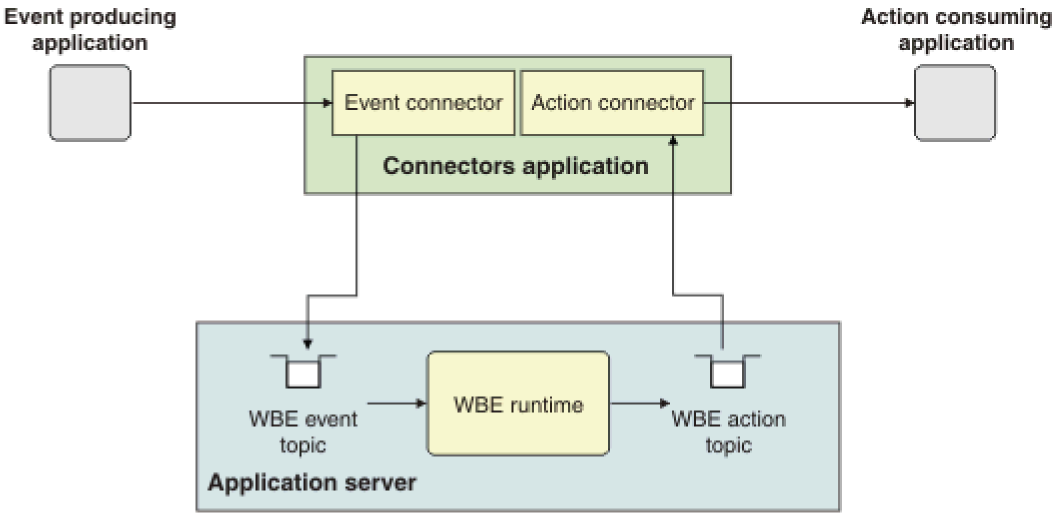


Figure 1: Base WebSphere Business Events connectors topology.

The connectors application is managed outside of the application server and only a single instance of the application can be running.

The Enhanced Connector Applications SupportPac provides alternative implementations of the File System, HTTP, JDBC, JMS, and SOAP connectors that run as applications within the application server environment, as shown in Figure 2:

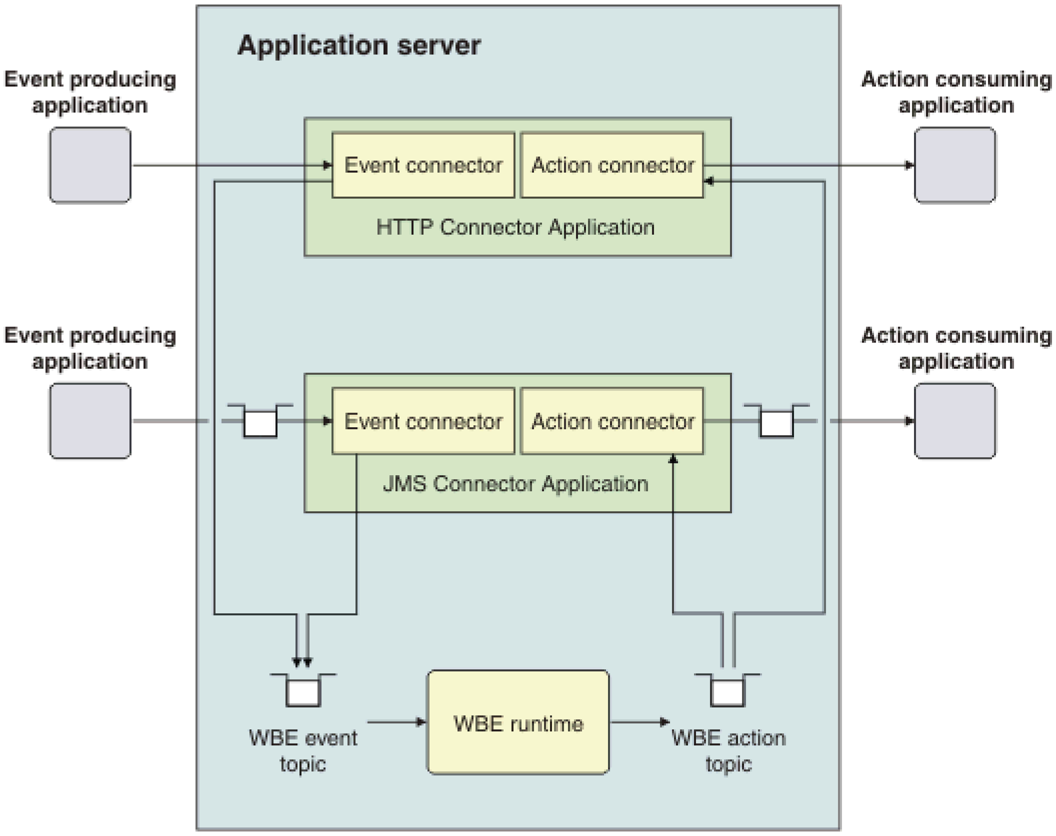

Application server
Event producing application
Action consuming application
Event connector
Action connector
HTTP Connector Application
Event producing application
Action consuming application
Event connector
Action connector
JMS Connector Application
WBE event topic
WBE runtime
WBE action topic

Figure 2: Enhanced connector applications running in the application server, with the HTTP and JMS (Message Queue) connectors as examples.

By running inside the application server environment, connector applications can be deployed to a cluster, providing greater scalability and high availability, as shown in Figure 3:

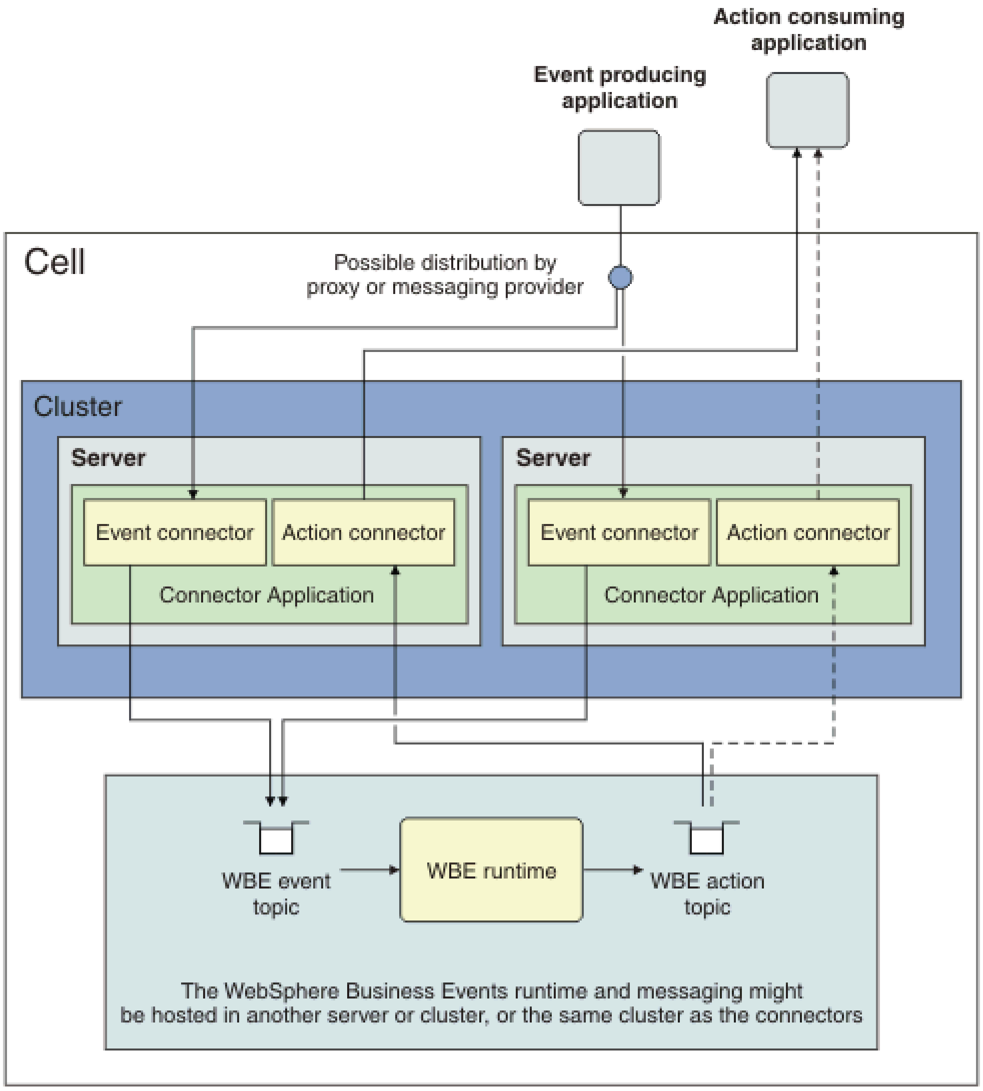

Action consuming application
Event producing application
Cell
Possible distribution by proxy or messaging provider
Cluster
Server
Event connector
Action connector
Connector Application
Server
Event connector
Action connector
Connector Application
WBE event topic
WBE runtime
WBE action topic
The WebSphere Business Events runtime and messaging might be hosted in another server or cluster, or the same cluster as the connectors

Figure 3: Enhanced connector applications in a cluster.

The connector applications are independent of the WebSphere Business Events Runtime. The connector applications can be installed on the same server or cluster as the WebSphere Business Events Runtime, or on a separate server or cluster in the same cell.

By deploying the connector applications to a cluster, the connectors can continue to process events and actions in the event of a failure of a cluster member, as shown in Figure 4:

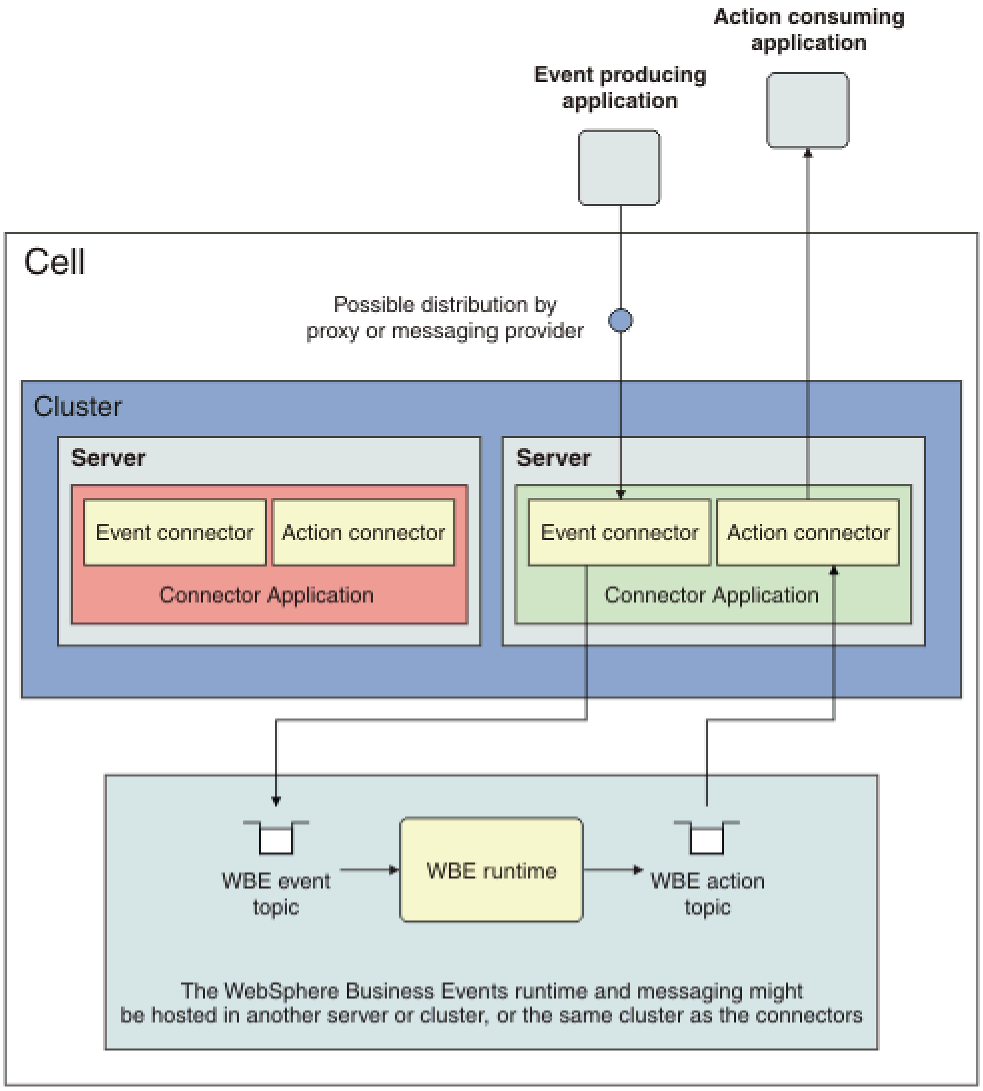

Action consuming application
Event producing application
Cell
Possible distribution by proxy or messaging provider
Cluster
Server
Event connector
Action connector
Connector Application
Server
Event connector
Action connector
Connector Application
WBE event topic
WBE runtime
WBE action topic
The WebSphere Business Events runtime and messaging might be hosted in another server or cluster, or the same cluster as the connectors

Figure 4: Enhanced connector applications in a cluster, showing high availability in the event of a server failure.

Installing the SupportPac

Before you begin

The Enhanced Connector Applications SupportPac must be installed on a machine hosting WebSphere Business Events V7.0.1 with Fix Pack 1.

If you are using WebSphere MQ as your Java Messaging Service (JMS) provider, you must upgrade WebSphere Application Server to V7.0.0.13, or later, before installing the SupportPac.

If WebSphere Business Events was installed with an installation type other than Typical, you must complete the following steps before installing the SupportPac:

- Locate the following file:

  Windows:
  ```
  <wbe_install_dir>\config\was\setenv.bat
  ```

  Linux and UNIX:
  ```
  <wbe_install_dir>/config/was/setenv.sh
  ```

  where <*wbe_install_dir*> is the installation location of WebSphere Business Events.

  Open the file in a text editor. Verify and where necessary update the properties to match your WebSphere Business Events environment.

- If the WebSphere Business Events Runtime is installed on a cluster, ensure that the following WebSphere Business Events properties are set to the correct values:

  ```
  as.director.connectors.soapRepositoryHost
  as.director.connectors.soapRepositoryPort
  ```

  If you use a proxy server to access the cluster members via HTTP, the property values must specify the hostname and port of the proxy. Otherwise, specify the hostname and port for one of the cluster members. When any changes are made to the property values, you must perform the following additional steps:

  1. Synchronize nodes
  2. Stop the cluster
  3. Stop and start the deployment manager
  4. Start the cluster

For further information about WebSphere Business Events properties and settings, see the *WebSphere Business Events 7.0.1.1 information center*.

About this task

The Enhanced Connector Applications SupportPac can be used with both stand-alone and clustered instances of WebSphere Business Events. For a stand-alone environment, you must install the connector SupportPac on the WebSphere Business Events machine. For a clustered environment, you must install the connector SupportPac on the machine that hosts the deployment manager for your environment.

Procedure

1. Obtain the connectors installation file. The file name is:

   Windows:
   `vp01.zip`

   Linux and UNIX:
   `vp01.tar.gz`

2. Copy the installation file to the machine hosting WebSphere Business Events. Extract the installation file to the directory where WebSphere Business Events is installed, *`wbe_install_dir`*. The default values for *`wbe_install_dir`* are:

   Windows:
   `C:\Program Files\IBM\WBE701`

   Linux and UNIX:
   `/opt/IBM/WBE701`

Results

The WebSphere Business Events Enhanced Connector Applications SupportPac is installed in the `csp` directory that is created in the *`wbe_install_dir`* directory.

Working with the SupportPac

Disabling the existing File System, HTTP, JDBC, JMS, and SOAP connectors

If you deploy connectors from the Enhanced Connector Applications SupportPac, you cannot continue to use the corresponding existing technology connectors. Before you run the new connectors, you must disable the existing connectors. All existing File System, HTTP, JDBC, JMS, and SOAP connectors must be disabled. Otherwise, actions can be duplicated, events and actions are processed less efficiently, and performance is degraded.

For example, if you want to use the new connectors to handle HTTP events and actions you must first disable the existing HTTP event and action connectors.

Before you begin

The application server or cluster hosting the WebSphere Business Events Runtime must be running. In a cluster environment, you must ensure that all cluster members are running along with the deployment manager.

If you are using WebSphere MQ as the JMS provider, the WebSphere MQ client jar files must be available on the machine where the **`disableTechnologyConnectors`** script is run. If the WebSphere MQ product is not installed on the machine, the jar files must be obtained from the WebSphere MQ product and copied to the machine. The following environment variable must also be set before running the script:

```
MQ_JAVA_LIB_PATH=<mq_java_lib_dir>
```

where *`<mq_java_lib_dir>`* is the directory containing the WebSphere MQ client jars. If the WebSphere MQ product is installed on the machine and the variable has not already been set, the variable should point to the directory *`<WebSphere_MQ_Install_Dir>`*`/java/lib`, where *`<WebSphere_MQ_Install_Dir>`* is the location of the directory where you installed WebSphere MQ.

Note: The WebSphere Business Events Runtime is stopped and started during the disabling process.

Procedure

To disable the existing technology connectors, complete the following steps:

1. Log onto the computer hosting WebSphere Business Events.
2. Stop the existing WebSphere Business Events connectors application (`connectors.bat` or `connectors.sh`) if it is running.
3. Navigate to the `<wbe_install_dir>/csp/bin` directory.
4. Run the following command:

   Windows:
   `disableTechnologyConnectors.bat <parameters>`

Linux and UNIX:

```
disableTechnologyConnectors.sh <parameters>
```

5. If necessary, restart the WebSphere Business Events connectors application.

Optional parameters for disableTechnologyConnectors

`-type`

A comma separated list of the technology connectors types to disable. Valid values are `file`, `http`, `jms`, and `soap`. If not specified, all of the File System, HTTP, JMS, and SOAP technology connectors are disabled.

`-messaging`

The type of messaging provider used by the WebSphere Business Events Runtime. Valid values are SIB and MQ.

`-host`

The hostname of the application server to connect to. For a clustered environment, you must specify the hostname of the deployment manager.

`-port`

The bootstrap port of the application server to connect to. For a clustered environment, you must specify the bootstrap port of the deployment manager.

`-providerURL`

Only applies when using WebSphere MQ as the messaging provider for queues and topics used by the WebSphere Business Events Runtime. For a single server environment, the parameter value specifies the provider URL for the server running the WebSphere Business Events Runtime and if not specified defaults to values based on the host and port arguments or their defaults. For a clustered environment, this parameter requires a corbaloc URL that specifies the hostname and port information for all application servers running the WebSphere Business Events Runtime, for example, `corbaloc:iiop:host1.sample.com:9810,iiop:host2.sample.com:9810,iiop:host3.sample.com:9810`

`-serverName`

For a single server environment only. The name of the application server where the WebSphere Business Events Runtime is installed. If not specified, the name of the server defined within the profile is used.

-clusterName

For a clustered environment only. The name of the application server cluster where the WebSphere Business Events Runtime is installed. If this parameter is specified, and messaging is MQ, the **-providerURL** parameter is also required.

-profileName

In a single server environment, the name of the WebSphere Application Server profile that contains the WebSphere Business Events Runtime. In a clustered environment, the name of the WebSphere Application Server profile that contains the deployment manager for the cell containing WebSphere Business Events. If this parameter is not specified, the default profile is used. In a cluster environment, the default might not match the required profile.

-user
-password

When the application server security is enabled, you must specify the administration user ID and password. There are three ways to enter the details:

1. You can enter the user ID and password each time the deploy process requires authentication. You are prompted if you do not specify the user ID and password on the command line or in a properties file.

2. You can specify the **–user** and **–password** parameters in the script. Although you are not prompted for the details during the deploy process, the user ID and password might be visible in the shell history and process listings.

3. You can specify the details in the `sas.client.props` file, which is located at:

   ```
   <path to WAS profile>/properties/sas.client.props
   ```

   To specify the credentials in this file, configure the following properties:

   ```
   com.ibm.CORBA.loginSource=properties
   com.ibm.CORBA.loginUserid=user ID
   com.ibm.CORBA.loginPassword=password
   ```

Examples

disableTechnologyConnectors.bat

The no argument form of the command disables the existing File System, HTTP, JDBC, JMS, and SOAP connectors. The process connects to the default host and port using the default profile and assumes a single server environment using SIB messaging.

disableTechnologyConnectors.bat –type=http

This command disables the existing HTTP connector only. The process connects to the default host and port using the default profile and assumes a single server environment using SIB messaging.

```
disableTechnologyConnectors.bat -messaging=MQ
-providerURL= corbaloc:iiop:localhost:2809
```

This command disables the existing File System, HTTP, JDBC, JMS, and SOAP connectors. WebSphere MQ is assumed to be the messaging provider. The default profile is used during the process. The WebSphere Business Events Runtime must be running on localhost using bootstrap port 2809.

```
disableTechnologyConnectors.bat -messaging=MQ -profileName=MyDmgr -
clusterName=MyCluster -port=2810 -providerURL=
corbaloc:iiop:host1.sample.com:9810,iiop:host2.sample.com:9810
```

This command disables the existing File System, HTTP, JDBC, JMS, and SOAP connectors. WebSphere MQ is assumed to be the messaging provider. The MyDmgr profile is used during the process and the bootstrap port of 2810 is used for connections to the dmgr. The WebSphere Business Events Runtime is installed on the MyCluster cluster and connections to it will use the providerURL information.

Deploying the new SupportPac connectors

After you have successfully installed the SupportPac, the connectors must be deployed before they can be used to consume events and produce actions. The deployment process encapsulates the following in a single command:

- Generating connector applications.
- Creating the necessary activation specifications required for the connectors to communicate with the WebSphere Business Events Runtime.
- Verifying that activation specifications required for consuming JMS events are defined, if required.
- Installing the connector applications to a server or cluster.

Before you begin

Before proceeding to deploy connectors, consider the following important points:

- The deployment process interrogates the WebSphere Business Events Runtime to determine which events and actions will use the connectors. Any events or actions which you want to consume or produce via the connectors must be published to the WebSphere Business Events Runtime before you deploy the connectors.
- The WebSphere Business Events Runtime must be running before you can deploy the connectors.
- If you publish additional event or action definitions to the WebSphere Business Events Runtime after deploying the connectors, the connectors must be redeployed before they are able to process these additional events or actions.
- If you publish changes to connector properties in event or action definitions, after deploying the connectors, the connectors must be redeployed before they recognise the changes.
- Publishing changes made solely to interaction sets, filters, or constants does not require the connectors to be redeployed.
- To consume events from message queues or topics you must create the appropriate activation specifications. These activation specifications must be created before attempting to deploy the connectors. For more details, see Using activation specifications when consuming JMS events.
- To consume events or produce actions by using the JDBC connector, you must specify the JNDI name of a WebSphere Application Server data source for the WebSphere Business Events datasource. For more details, see Configuring the WebSphere Business Events datasource.
- If a specific connector has no events or actions to process, that connector is not deployed.
- If the **`-update`** parameter is specified and a previously deployed connector is no longer required, it is removed.

Procedure

The connectors are deployed using the **`deployConnectors`** command supplied by the SupportPac. The **`deployConnectors`** command can be used to deploy, update, and create File System, HTTP, JDBC, JMS, and SOAP connectors. To execute the command complete the following steps:

1. Log onto the computer hosting WebSphere Business Events.
2. Navigate to the `<wbe_install_dir>/csp/bin` directory.
3. Run the following command:

Windows:

```
deployConnectors.bat <parameters>
```

Linux and UNIX:

```
deployConnectors.sh <parameters>
```

Required parameters for deployConnectors

`-outputDir`

The directory where the connector applications are created. If multiple connector types are generated, this directory will contain an application for each connector type. The log file for the deployment process is also created within the specified directory and is named `deployConnectors_<date>.log`.

Optional parameters for deployConnectors

`-serverName`

The name of the application server to which the connector applications are to be installed. If neither the **`-serverName`** or **`-clusterName`** parameters are specified, a single server is assumed to exist in the profile and the name of that server is used. If the server does not exist, deployment will fail.

`-clusterName`

Required for a cluster environment only. The name of the application server cluster to which the connector applications are to be installed.

`-profileName`

In a single server environment, the name of the WebSphere Application Server profile that contains the WebSphere Business Events Runtime. In a clustered environment, the name of the WebSphere Application Server profile that contains the deployment manager for the cell containing WebSphere Business Events. If this parameter is not specified, the default profile is used. In a cluster environment, the default might not match the required profile.

`-type`

The connectors to be deployed, updated, or created. The connector types must be specified as a comma separated list. The valid connector types are `file`, `http`, `jms`, and `soap`. If this parameter is not specified, the connector types are automatically detected based on the projects deployed to the WebSphere Business Events Runtime. Only connectors which are used by currently deployed projects are created.

`-messagingServerName`

The name of the application server where the messaging engine for the WebSphere Business Events messaging resides. Defaults to the **`-serverName`** value if specified, otherwise the name of the server defined in the profile.

`-messagingClusterName`

The name of the cluster where the messaging engine for the WebSphere Business Events messaging resides. Defaults to the **`-clusterName`** value if specified, otherwise the name of the cluster defined in the profile.

`-host`

The hostname of the application server to connect to when connectors are deployed, updated, or created. For a clustered environment, you must specify the hostname of the deployment manager.

`-port`

The bootstrap port of the application server to connect to when connectors are deployed, updated, or created. For a clustered environment, you must specify the bootstrap port of the deployment manager.

`-providerURL`

Provides the hostname and port information in the form of a corbaloc URL, for all application servers running the WebSphere Business Events Runtime. If the WebSphere Business Events Runtime is installed to a single server, this parameter may be excluded, in which case the values for the host and port parameters will be used. If the WebSphere Business Events Runtime is installed to a cluster, this parameter must be specified and should list all servers in the cluster, for example,

```
corbaloc:iiop:host1.sample.com:9810,iiop:host2.sample.com:9810,iiop:host3.sample.com:9810
```

`-update`

Updates the connector applications after they have been deployed. When the connector applications are updated, any connectors that are no longer required are uninstalled. For example, an existing JMS connector application will be uninstalled if the connector is not used by any of the projects.

`-create`

Creates and saves the connector applications to the directory specified by the **`-outputDir`** parameter. The connector applications can then be deployed manually as required.

`-user`
`-password`

When the application server security is enabled, you must specify the administration user ID and password. There are three ways to enter the details:

1. You can enter the user ID and password each time the deploy process requires authentication. You are prompted if you do not specify the user ID and password on the command line or in a properties file.

2. You can specify the **-user** and **-password** parameters in the script. Although you are not prompted for the details during the deploy process, the user ID and password might be visible in the shell history and process listings.

3. You can specify the details in the `sas.client.props` file, which is located at:

   ```
   <path to WAS profile>/properties/sas.client.props
   ```

   To specify the credentials in this file, configure the following properties:

   ```
   com.ibm.CORBA.loginSource=properties
   com.ibm.CORBA.loginUserid=user ID
   com.ibm.CORBA.loginPassword=password
   ```

-help

The help information is displayed. If other parameters are specified they are ignored.

Parameters applicable only when using WebSphere MQ as your JMS provider

-mqTransportType

The type of transport to use to connect to the queue manager. This parameter is required if using WebSphere MQ. Valid values are: `CLIENT`, `BINDINGS`, or `BINDINGS_THEN_CLIENT`.

-mqQMHost

The hostname of the machine on which the WebSphere MQ queue manager is running. This parameter is optional for the transport type of `BINDINGS`, but is required for all other transport types.

-mqQMPort

The port on which the WebSphere MQ queue manager listener is accepting connections.

-mqQMName

The name of the WebSphere MQ queue manager to connect to. This parameter is required when using WebSphere MQ as your JMS provider.

-mqChannelName

The name of the channel used for the MQ connection.

Examples

```
deployConnectors.bat -outputDir=C:\temp\connectors
```

This command generates the File System, HTTP, JDBC, JMS, and SOAP connectors, if required. The connector applications are output to the directory C:\temp\connectors. If the applications are not currently installed, they are installed to the local application server using the default profile, host, and port values. If the applications are already installed, there is no update.

```
deployConnectors.bat -outputDir=C:\temp\connectors -update
```

This command generates the File System, HTTP, JDBC, JMS, and SOAP connectors, if required. The connector applications are output to the directory `C:\temp\connectors`. The applications are then installed to the local application server using the default profile, host, and port values. If the applications are already installed on the server, they are updated or uninstalled if no longer required.

```
deployConnectors.bat -outputDir=C:\temp\connectors -type=http -update
```

This command generates only the HTTP connector, if required. The connector application is output to the directory `C:\temp\connectors`. The connector application is then installed to the local application server using the default profile, host, and port values. If the application is already installed on the server, it is updated or uninstalled if no longer required.

```
deployConnectors.bat -outputDir=C:\temp\connectors -profileName=MyDmgr
-clusterName=MyCluster -port=2810
-providerURL=
corbaloc:iiop:host1.sample.com:9810,iiop:host2.sample.com:9810 -update
```

This command generates the File System, HTTP, JDBC, JMS, and SOAP connectors, if required. The connector applications are output to the directory `C:\temp\connectors`. The connector applications are then installed to the cluster named MyCluster. The MyDmgr profile is used during the deployment and the bootstrap port of 2810 is used for connections to the dmgr. Connections to the WebSphere Business Events Runtime will use the providerURL information. If the applications are already installed on the cluster, they are updated or uninstalled if no longer required.

```
deployConnectors.bat -outputDir=C:\temp\connectors -
mqQMHost=host1.sample.com
-mqQMPort= 1418 -mqQMName=myqm -mqTransportType= BINDINGS_THEN_CLIENT
-update
```

This command generates the File System, HTTP, JDBC, JMS, and SOAP connectors, if required. The connector applications are output to the directory `C:\temp\connectors`. The connector applications are then installed to the local server. The default profile is used during the deployment along with default bootstrap host and port values. WebSphere MQ messaging is assumed to be the messaging provider and activation specifications will use the MQ properties specified. If the applications are already installed on the cluster, they are updated or uninstalled if no longer required.

Re-enabling the existing File System, HTTP, JDBC, JMS, and SOAP connectors

Before you begin

You might want to re-enable the existing connectors if do not want to use the new connectors, but you do not want to uninstall the SupportPac. If you do want to uninstall the SupportPac (which will remove the new connectors), see Uninstalling the SupportPac.

The application server or cluster hosting the WebSphere Business Events Runtime must be running. In a cluster environment, you must ensure that all cluster members are running along with the deployment manager.

Note: The WebSphere Business Events Runtime is stopped and started during the enabling process.

Procedure

If required, the existing technology connectors can be re-enabled by completing the following steps:

1. Log onto the computer hosting WebSphere Business Events.
2. Stop the existing WebSphere Business Events connectors application (`connectors.bat` or `connectors.sh`) if it is running.
3. Navigate to the `<wbe_install_dir>/csp/bin` directory.
4. Run the following command:

   Windows:
   `enableTechnologyConnectors.bat <parameters>`

   Linux and UNIX:
   `enableTechnologyConnectors.sh <parameters>`

5. If necessary, restart the WebSphere Business Events connectors application.

Optional parameters for enableTechnologyConnectors

`-type`

> A comma separated list of the technology connector types to enable. Valid values are `file`, `http`, `jms`, and `soap`. If not specified, all of the File System, HTTP, JMS, and SOAP technology connectors are enabled.

`-host`

> The hostname of the application server to connect to. For a clustered environment, you must specify the hostname of the deployment manager.

`-port`

> The bootstrap port of the application server to connect to. For a clustered environment, you must specify the bootstrap port of the deployment manager.

-serverName

For a single server environment only. The name of the application server where the WebSphere Business Events Runtime is installed. If not specified, the name of the server defined in the profile is used.

-clusterName

For a clustered environment only. The name of the application server cluster where the WebSphere Business Events Runtime is installed.

-profileName

In a single server environment, this parameter specifies the name of the WebSphere Application Server profile that contains the WebSphere Business Events Runtime. In a clustered environment, this parameter specifies the name of the WebSphere Application Server profile that contains the deployment manager for the cell containing WebSphere Business Events. If this parameter is not specified, the default profile is used. In a cluster environment, the default may not match the required profile.

-user
-password

When the application server security is enabled, you must specify the administration user ID and password. There are three ways to enter the details:

1. You can enter the user ID and password each time the deploy process requires authentication. You are prompted if you do not specify the user ID and password on the command line or in a properties file.

2. You can specify the **–user** and **–password** parameters in the script. Although you are not prompted for the details during the deploy process, the user ID and password might be visible in the shell history and process listings.

3. You can specify the details in the `sas.client.props` file, which is located at:

   ```
   <path to WAS profile>/properties/sas.client.props
   ```

   To specify the credentials in this file, configure the following properties:

   ```
   com.ibm.CORBA.loginSource=properties
   com.ibm.CORBA.loginUserid=user ID
   com.ibm.CORBA.loginPassword=password
   ```

Examples

```
enableTechnologyConnectors.bat
```

The no argument form of the command enables all of the existing File System, HTTP, JDBC, JMS, and SOAP connectors. The process connects to the default host and port using the default profile and assumes a single server environment using SIB messaging.

```
enableTechnologyConnectors.bat –type=http
```

This command enables the existing HTTP connector only. If the existing JMS or SOAP connectors have been disabled, they remain disabled. The process connects to the default host and port using the default profile and assumes a single server environment using SIB messaging.

The File System connector

You can use the File System connector to read events from files, and to write actions to files.

Why you might want to use the File System connector

You might want to use the new File System connector if:

- You want to generate events from delimited files such as comma-separated (.csv) files.
- You want to process files in a highly available manner, so if one server in a cluster hosting an instance of the connector fails, that processing continues from the point of failure on another server.

Scenario

An IT support specialist sends the daily transactions for a local branch to the company head office once a day. All the transactions for the day, for that branch, are recorded in a single text file, with one transaction per line in a comma-separated delimited format. The file is copied to the head office servers in a specified directory at the end of the day. The file containing the daily transactions is always named `<`*`BRANCHNAME`*`>_`*`DATE`*`_transactionsLog.txt`, where <*BRANCHNAME*> is the name of the local branch, and *DATE* is the date of the day on which the file was created.

At head office, the company has WebSphere Business Events installed in a clustered environment to ensure that event processing can continue in the event of partial system failure. Each file might contain millions of transactions: it is important in the event of partial system failure of the WebSphere Business Events cluster that no transactions are lost, and that processing of the file can continue from the point of failure on another cluster member.

The company wants each transaction to generate a single event within WebSphere Business Events, and it wants each file to be processed within an hour of being received on its servers.

The scenario is shown in Figure 6:

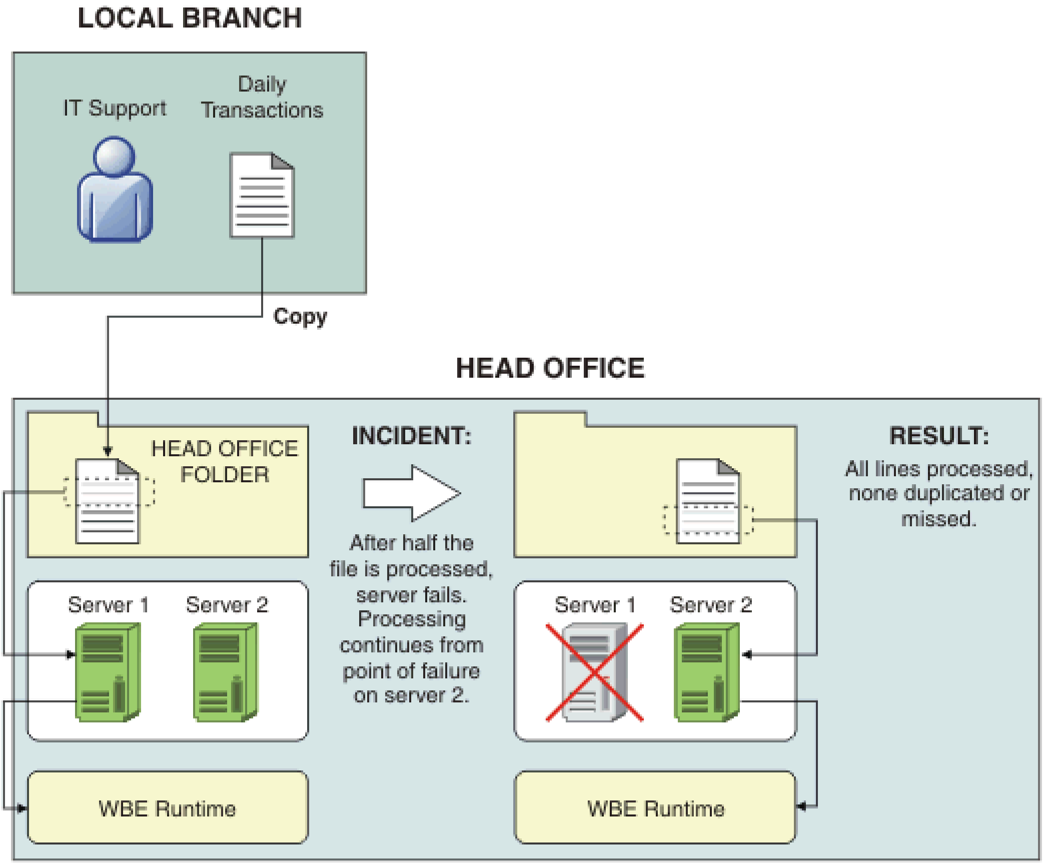

LOCAL BRANCH
IT Support
Daily Transactions
Copy
HEAD OFFICE
HEAD OFFICE FOLDER
INCIDENT:
After half the file is processed, server fails. Processing continues from point of failure on server 2.
RESULT:
All lines processed, none duplicated or missed.
Server 1
Server 2
Server 1
Server 2
WBE Runtime
WBE Runtime

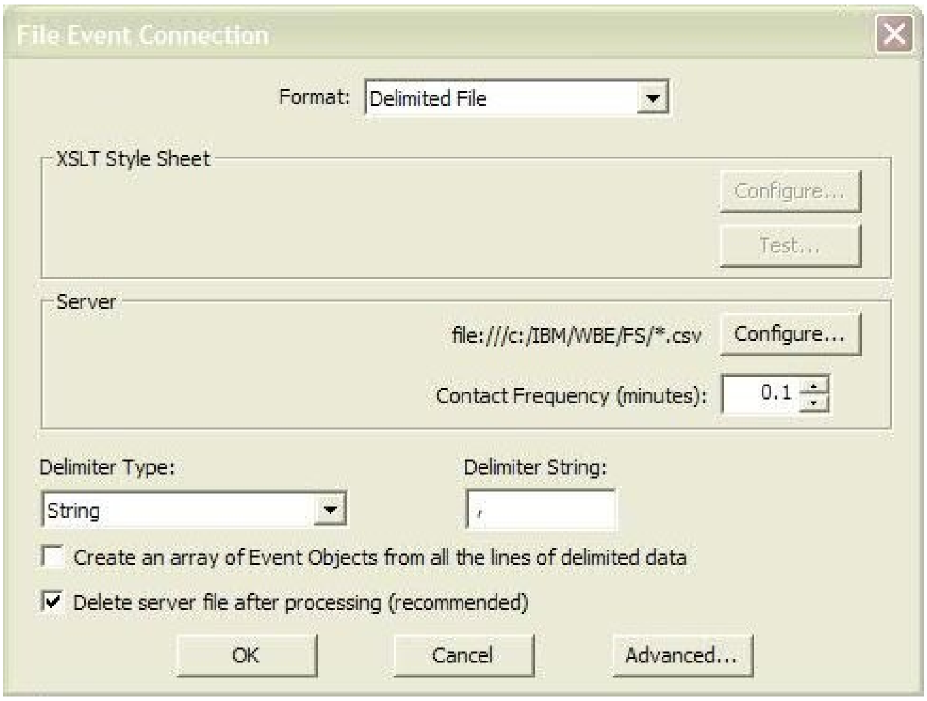
Figure 6: Scenario for using the File System connector with WebSphere Business Events installed in a clustered environment.

Consuming events by using the File System connector

The File System connector supports a number of formats for events consumed from files. The format is specified in the connection settings for the event by using the Design Data tool.

Configuring an event to use the File System connector

In Design Data, right-click the event and select **Event Properties**. Click the **Connection** tab then **File System Connection** and then click **Configure**. The following screen capture shows the File Event Connection window:



File Event Connection

Format: Delimited File

XSLT Style Sheet

Server

file:///c:/IBM/WBE/FS/*.csv

Contact Frequency (minutes): 0.1

Delimiter Type: String

Delimiter String: ,

□ Create an array of Event Objects from all the lines of delimited data

☑ Delete server file after processing (recommended)

You can configure the following settings:

Format

The format of the event data that is expected by the connector. The File System connector expects the event to be received in the format specified. There are four formats:

Connector Packet
The format must match the format of the WebSphere Business Events event packet schema.

Custom XML (XSL input)
The event data can be any custom xml document. The specification of an XSL style sheet is required. The style sheet must transform the custom xml to the WebSphere Business Events event packet format.

Delimited File
The event data is in delimited format. The data is separated by the delimiter specified in the **Delimiter Type** field.

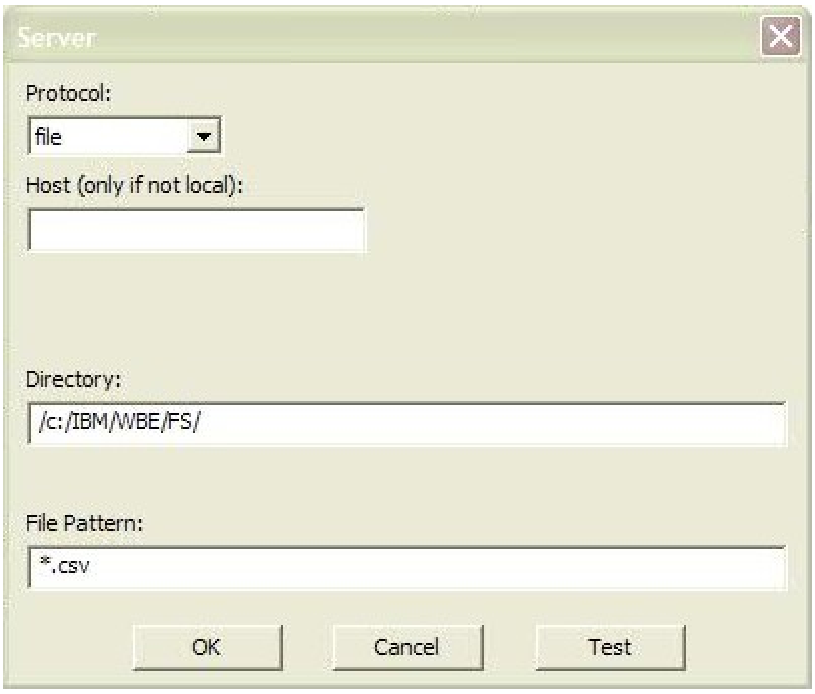
File Info Only
The event data contains only information about the file, not its contents. This information includes, for example, the file name and directory, the file size, and the date and time of last modification.

XSLT Style Sheet

Required if Custom XML is specified as the format. Specifies the XSL style sheet used to transform the WebSphere Business Events action packet format into the required custom xml.

Server

The source location of the delimited file. Click **Configure**. The following screen capture shows the Server window:



Protocol: file

Host (only if not local):

Directory: /c:/IBM/WBE/FS/

File Pattern: *.csv

You can select the following options:

Protocol
The way in which the event is handled. You can select only file.

Host
The DNS name or IP address of the computer from which the event is retrieved. If the file is stored on the same computer as WebSphere Business Events Runtime, leave this field blank. The system on which the File System connector is installed must have the appropriate authentication to access the remote file system.

Directory
The fully-qualified path of the directory where files are retrieved. If you are accessing a directory on another (non-Windows) computer, that directory must be mounted (sharable).

File Pattern
The pattern that the file name must match to initiate the event. Use an asterisk (*) as a wildcard to match any number of characters, and use a question mark (?) to match a single character. For example, *.xml matches any file with an extension of .xml, while my????.* matches a file with a 6-character file name beginning with "my" and any file extension. A pattern with a value of myfile.xml requires a file name with an exact match. A value of * (the default) means that any file matches.

Contact Frequency

The polling interval in minutes for the File System connector to check for any file modifications. The default value is 10 minutes.

Delimiter Type

The way that fields are separated in the file. There are three types:

New Line

Each field is on its own line. Each file contains one event.

String

Each field is delimited by a string defined in the Delimiter String setting, for example, "**,**" would be used for a comma-separated file. Each line of the file represents one event.

Tab

Each field is delimited by a tab. Each line of the file represents one event.

Create an array of Event Objects from all the lines of delimited data

The way that data is mapped to events and event objects.

Clear this option if you want each line of delimited data mapped to a single event containing a single event object.

Select this option if you want each line of delimited data mapped to an event object and each file is mapped to a single event containing an array of event objects representing the lines of delimited data.

Note: selecting this option results in degraded performance.

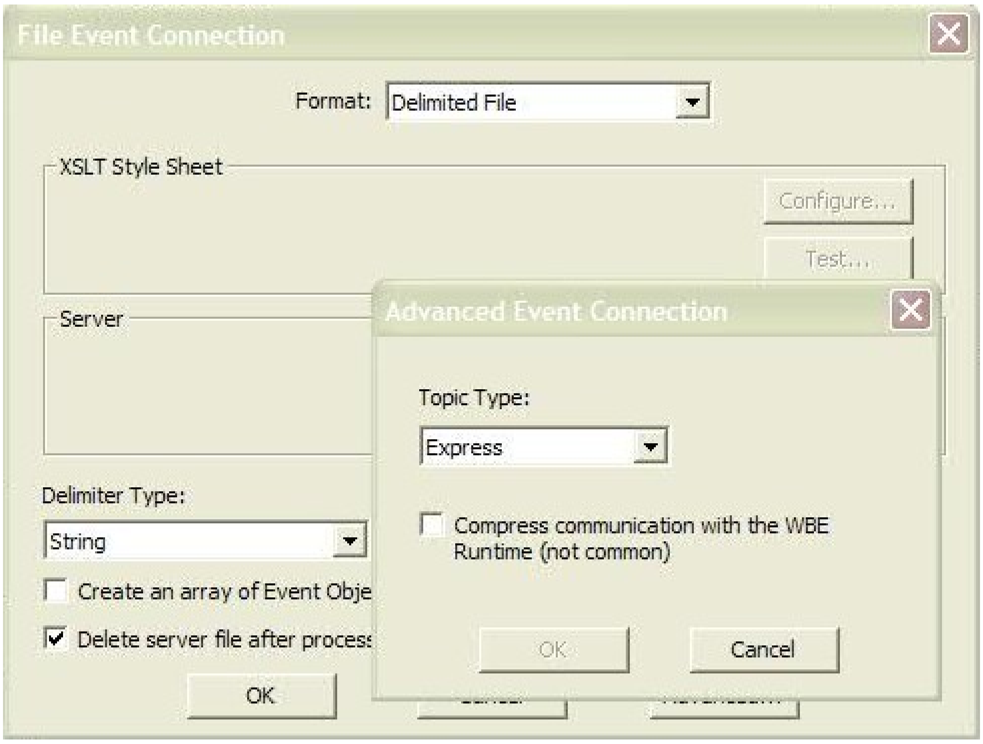
Delete server file after processing

The option to clean up files after processing.

Select this option so that the file containing the event data is deleted after it has been processed and all events have been sent.

You can configure the following advanced settings, by clicking **Advanced**:

The following screen capture shows the Advanced Event Connection window:



Format: Delimited File

Delimiter Type: String

□ Create an array of Event Obje

☑ Delete server file after process

Topic Type: Express

□ Compress communication with the WBE Runtime (not common)

Topic Type

Specifies how messages are transmitted between the connector and the WebSphere Business Events Runtime:

Express
Indicates that messages are non-persistent and that the WebSphere Business Events Runtime does not use durable subscriptions when consuming the messages.

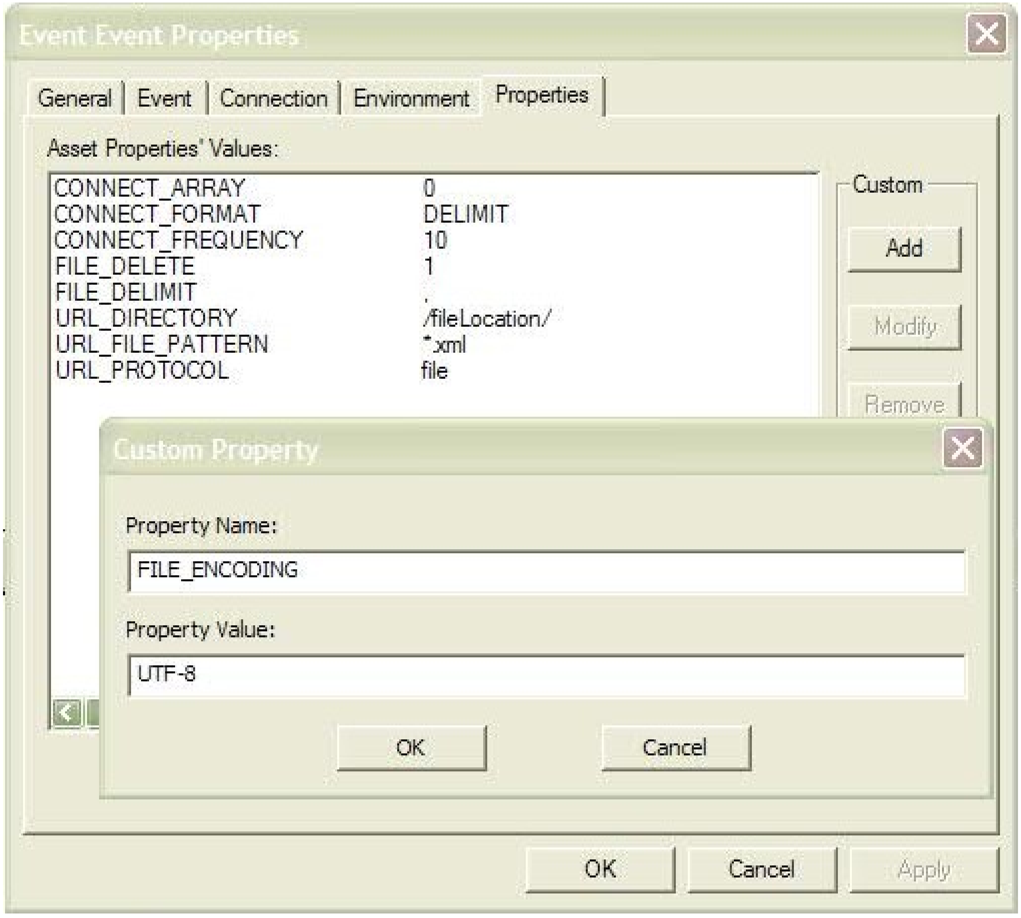
Reliable
Indicates that messages are persistent and that the WebSphere Business Events Runtime uses durable subscriptions when consuming the messages. Use Reliable if it is essential that no messages are lost in the event of server failure.

Compress communication with the WBE Runtime (not common)

Specifies whether the data in the event messages sent by the connector to the WebSphere Business Events Runtime is compressed.

Configuring custom properties for the File System connector

In Design Data, right-click the event and select **Event Properties**. Click the **Properties** tab and then click **Add**. The following screen capture shows the Event Properties window, with an example of a custom property for an event named Event:



Asset Properties' Values:

CONNECT_ARRAY	0
CONNECT_FORMAT	DELIMIT
CONNECT_FREQUENCY	10
FILE_DELETE	1
FILE_DELIMIT	,
URL_DIRECTORY	/fileLocation/
URL_FILE_PATTERN	*.xml
URL_PROTOCOL	file

Property Name: FILE_ENCODING

Property Value: UTF-8

FILE_ENCODING
Specifies the encoding of the files used by the File System connector. The default value if not set is UTF-8. Set this value if the files you want to process are not UTF-8 encoded.

MAX_BATCH_SIZE
Specifies the number of lines to process at a time from a file. The default value is 250. If you set a value for this property and the option **Create an array of Event Objects from all the lines of delimited data** is selected, the batch size corresponds to the maximum number of event objects expected in an event. For more information, see Transaction handling.

TRIGGER_FILE_EXT
Specifies the file extension for trigger files. If not set, trigger files are not used and processing of files begins as soon as they are detected within the specified directory. For more information, see File locking and trigger files.

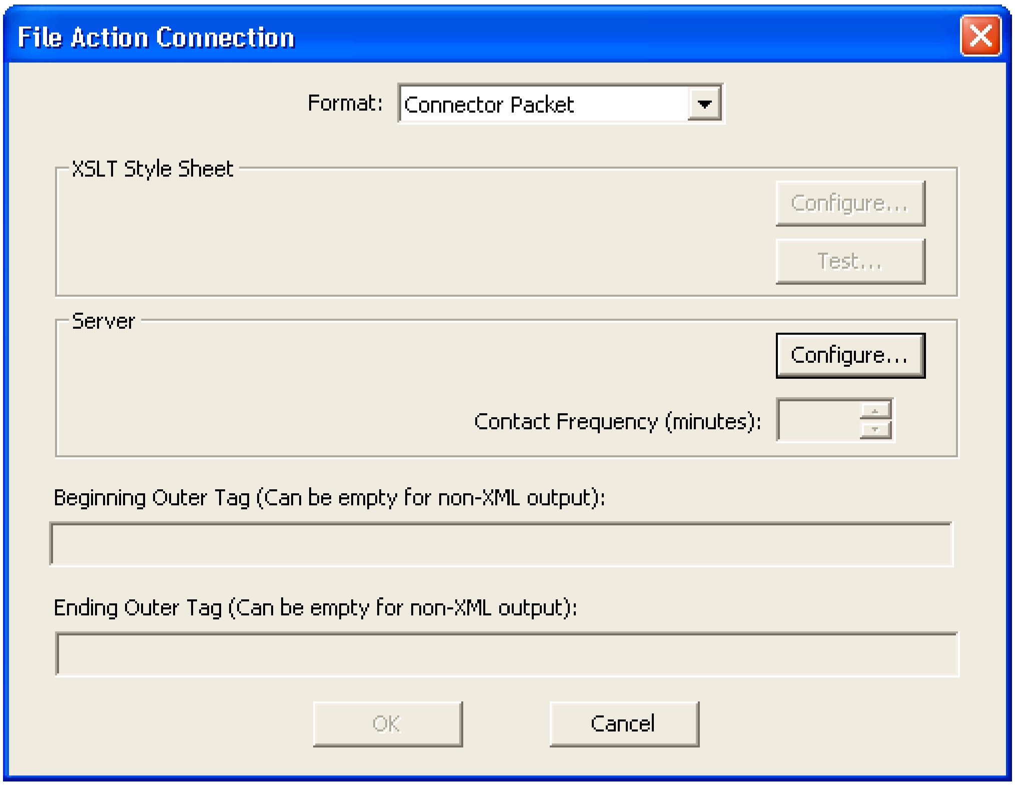
XA_LOG_DIR
Specifies the log directory for transaction logs for the File System connector. If not set, logs are stored in a subdirectory named fcxa in the directories from which the events are retrieved. For more information, see Transaction handling.

Producing actions by using the File System connector

The File System connector supports two formats for actions written to files. The format is specified in the connection settings for the action by using the Design Data tool.

Configuring an action to use the File System connector

In Design Data, right-click the action and select **Action Properties**. Select the **Connection** tab then **File System Connection** and then click **Configure**. The following screen capture shows the File Action Connection window:


File Action Connection
Format: Connector Packet
XSLT Style Sheet
Configure...
Test...
Server
Configure...
Contact Frequency (minutes):
Beginning Outer Tag (Can be empty for non-XML output):
Ending Outer Tag (Can be empty for non-XML output):
OK
Cancel

You can configure the following settings:

Format
The format of the action data that is sent by the File System connector. Choose between:

Connector Packet
The action data is formatted in the WebSphere Business Events action packet format.

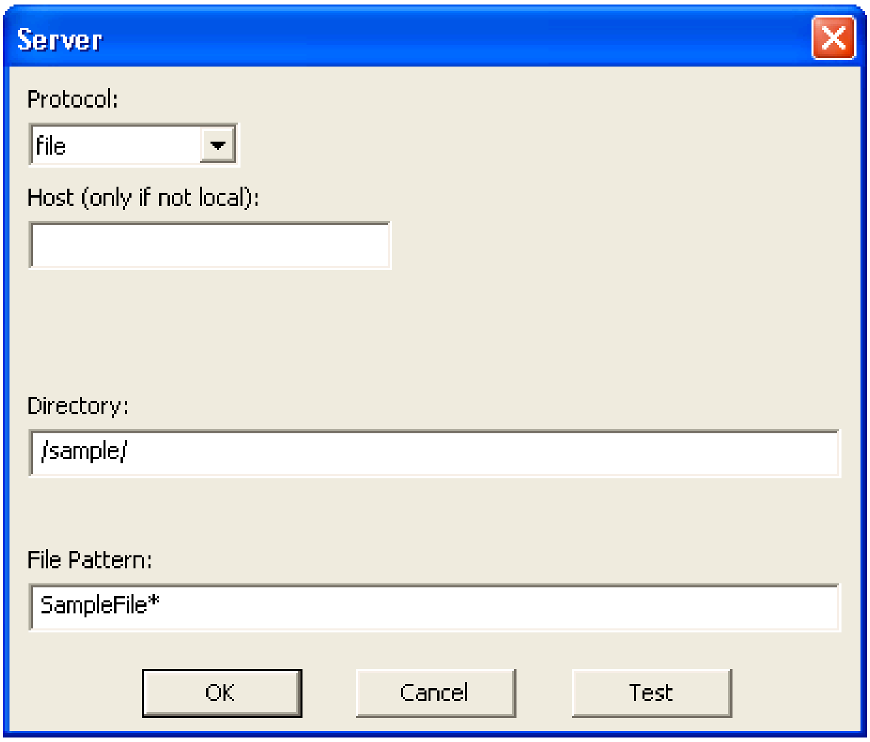
Custom XML (XSL output)
The action packet is transformed using an XSL style sheet into a custom XML format. The specification of the XSL style sheet is required.

XSLT Style Sheet
Required if Custom XML is specified as the format. Specifies the XSL style sheet used

to transform the WebSphere Business Events action packet format into the required custom XML.

Server

To specify the URL to which the connector sends the action data and how it is sent. Click **Configure**. The following screen capture shows the Server window:



You can configure the following Server settings:

Protocol
The way in which the action is handled. You can select only **file**.

Host
The DNS name or IP address of the computer to which the action is written. If the file is stored on the same computer as WebSphere Business Events Runtime, leave this field blank. The system on which the File System connector is installed must have the appropriate authentication to access the remote file system.

Directory
The fully-qualified path of the directory where files are written. If you are accessing a directory on another (non-Windows) computer, that directory must be mounted (sharable).

File Pattern
The pattern of the file or files to be created. Enter a file pattern that includes an asterisk (*) and the connector creates a matching file for each action. The asterisk (*) in the pattern is replaced with an UID. The default is *, which means the file is created without an extension (file type). If an asterisk (*) is not used

and the specified file already exists, the file will not be overwritten.

Beginning Outer Tag
This value is not used by the Enhanced Connector Applications SupportPac File System connector.

Ending Outer Tag
This value is not used by the Enhanced Connector Applications SupportPac File System connector.

Ensuring high availability of file processing

When the File System connector is deployed to a cluster, it is enabled for high availability. The connector is active on one member of the cluster at any one time. If the server with the active connector fails, the connector automatically becomes active on another server within the same cluster. If failover occurs, and no “in doubt” transactions exist, the connector continues processing existing files from the line that had previously been reached. If “in doubt” transactions exist, these transactions are resolved before any further processing of the file. Whether transactional recovery occurs on the server on which the connector has become active is determined by whether peer recovery is enabled. For more information, see Transactional high availability.

Configuring retry values for sending JMS messages

When the File System connector sends events to the WebSphere Business Events Runtime, JMS connections are established. If the connections cannot be established, the connectors will retry for two minutes. The period of time spent in retrying to establish a connection can be overridden by setting one the following WebSphere Business Events properties:

```
wbe.csp.jms.connection.max.retry.time
```

The period of time, in seconds, that the connectors will retry to establish a connection.

```
wbe.csp.jms.connection.max.retry.attempts
```

The maximum number of attempts that the connectors will make to establish a connection. Setting this value will override the retry time value.

After changing these properties, if the connector applications have been deployed, the connector applications must be updated by running the deployConnectors command specifying the –update argument in addition to any other arguments required for your topology.

Transaction handling

Batch size

The MAX_BATCH_SIZE property determines the maximum number of lines to read from a file in a single batch. Each batch is read and sent to the WebSphere Business Events Runtime in a single transaction. If any errors occur during the processing of a batch, the full batch is discarded and reprocessed.

xalog files

The File System connector uses its own files to keep a record of transactional information. For each event file that is processed, a corresponding xalog file is generated. The name of the file is `<event_file_name>.xalog` where *<event_file_name>* is the name of the event file being processed. By default, these files are generated in a subdirectory named fcxa that is created within the directory containing the event file. If a different name for the subdirectory is required, it can be specified by adding a custom property to the connector settings for the event named XA_LOG_DIR.

Note: Do not delete the xalog file unless the corresponding event file has been fully processed. If you select the **Delete server file after processing** option for the event, the xalog file will automatically be deleted after the event file has been deleted.

Transactional high availability

By default, if a server failure occurs in a cluster, transaction recovery for that server occurs when the server is restarted. Recovery might be delayed if the server fails to restart. WebSphere Application Server can be configured for peer recovery, so that transaction recovery for one server within a cluster occurs on another server in the same cluster. You must enable peer recovery to provide transactional high availability for the File System connector. For more information about enabling peer recovery, see Transactional high availability in the *WebSphere Application Server information center*.

Status checking

If you selected the **Delete server file after processing** option for events, any files processed in relation to the events are deleted after processing is complete. Corresponding xalog or trigger files are also deleted. If you cleared the option, files remain after processing is complete. In this situation, it might be unclear if a file has been fully processed. To establish the status of file processing, complete the following steps:

Checking status when the connector is running

If the File System connector is running, you can retrieve the processing status by enabling the following diagnostics trace string for the server or servers on which the connector is running:

```
com.ibm.wbe.csp.XALogTrace=all
```

When the trace setting is enabled, the following information is recorded in the server trace file for each file processed:

```
    --------------------------------------------------------------
    File processing state for C:\temp\eventfiles\example.txt
    Complete: false
    Lines processed: 250
    Last processed: 12/3/11 15:49
    --------------------------------------------------------------
```

The information is recorded each time that the file is examined before possible processing, then every two minutes while being processed.

Checking status when the connector is stopped

If the File System connector is stopped (for example, if the server or cluster on which it is running is stopped) a script is provided that generates a report, in the form of a csv file, that lists files and their related status. The script is named `fcLogReport.bat(.sh)` and can be executed by following these steps:

1. Log in to the computer hosting WebSphere Business Events
2. Navigate to the `<wbe_install_dir>/csp/bin` directory
3. Run the following command:

   Windows:
   ```
   fcLogReport.bat <parameters>
   ```

   Linux and UNIX:
   ```
   fcLogReport.sh <parameters>
   ```

Required parameters for fcLogReport

`-eventDir`

The directory containing the files that are processed by the File System connector.

`-reportFile`

The fully-qualified name of the report file to generate.

Optional parameters for fcLogReport

`-logSubDir`

The name of the sub directory beneath `eventDir` that contains the xalog files. The default is `fcxa`

Example

The following script:

```
fcLogReport.bat -eventDir=C:\temp\eventfiles
-reportFile=C:\reports\myReport.csv
```

might generate the following report:

```
File name, Complete, Lines processed, Last processed
C:\temp\eventfiles\example.txt,false,250,12/3/11 15:49
C:\temp\eventfiles\example2.txt,true,1234,12/3/11 15:21
```

File locking and trigger files

By default, the File System connector begins to process a file immediately after it is detected in the specified directory. When processing begins, a lock is obtained on the file. If another process has a lock on the file, which might be the case when the file is being copied to the directory, processing is delayed until the lock can be obtained. An attempt to gain the lock is made at each polling interval.

Some methods of copying files do not use locking, in particular FTP. When using such methods, the File System connector might attempt to process a file before it has been transferred to the directory, and by obtaining a lock might cause the file transfer to abort. To prevent this problem, the File System connector supports the use of trigger files. A trigger file is a file with a specified name and extension that must exist before the corresponding event file is processed. The use of trigger files for an event is enabled by specifying the TRIGGER_FILE_EXT connector custom property. The value of the property is the file extension for the trigger file. For example, if you specify a value of `trigger`:

1. The File System connector detects a file corresponding to the directory and pattern for the event.
2. The connector checks if a file with the name `<matched_file>.trigger` exists in the same directory, where *<matched_file>* is the file matched in step 1. For example, a matched file named `example.dat` with TRIGGER_FILE_EXT set to `trigger` is processed only when a file named `example.dat.trigger` is also found.

By using trigger files, a large file can be transferred to the monitored directory without any processing occurring. When the transfer is complete, a trigger file can be transferred. If you select the **Delete server file after processing** option for the event, the trigger file is automatically deleted when the event file has been deleted.

Accessing mapped network drives in Microsoft Windows

The File System connector can process files on a mapped network drive in Microsoft Windows. If WebSphere Business Events is running as a service, the mapping might not be visible to the user under which the service is running. An alternative to specifying the path by using a mapped drive letter is to specify the host name in the connector settings, producing a Universal Naming Convention (UNC) style path. Using the UNC approach, no drive mappings are required; however the credentials for the user under which the service is running must be sufficient to access the remote file system. For example:

If you have:	**The following settings are equivalent:**
Microsoft Windows mapped network drive called: `Z -> \\myhost.example.com\shared`	Host specified in connector settings for the event: `myhost.example.com`
Directory specified in connector settings for the event: `Z:\mydir`	Directory specified in connector settings for the event: `\shared\mydir`

Restricting event delivery rate to the WebSphere Business Events Runtime

By default, the File System connector processes a batch of lines from a file as soon a possible after the previous batch, if it exists, has been committed. In scenarios where many files or large files are being processed, the rate at which messages are put on the event queue for the Websphere Business Events Runtime might outpace the rate at which the runtime is consuming them.

Reducing the event production rate

To reduce the likelihood of filling the event queue with events from the File System connector, configure the connector with a recommended maximum event production rate. Specify the rate in events/sec. When set, the connector attempts to avoid producing events at a rate beyond the specified rate, however the value is not a strict limit and the true rate might fluctuate beyond the value.

To specify the production rate, for each server on which the connector is running, set the following JVM property:

Name	**Value**
com.ibm.wbe.csp.file.rate	The rate in events/sec

Handling errors for full queues

On receipt of errors relating to full queues, the connector delays the resending of the last batch. When the batch is resent, if errors are encountered then the delay is increased. When successful sending of the batch occurs, attempts are made to reduce the delay and increase the batch processing rate again. If further errors are detected in response to an increased rate, the rate is reduced again and maintained at the lower level for a longer period.

Example of a cluster configuration

An example of a cluster configuration for the File System connector and the WebSphere Business Events Runtime is shown in Figure 7:

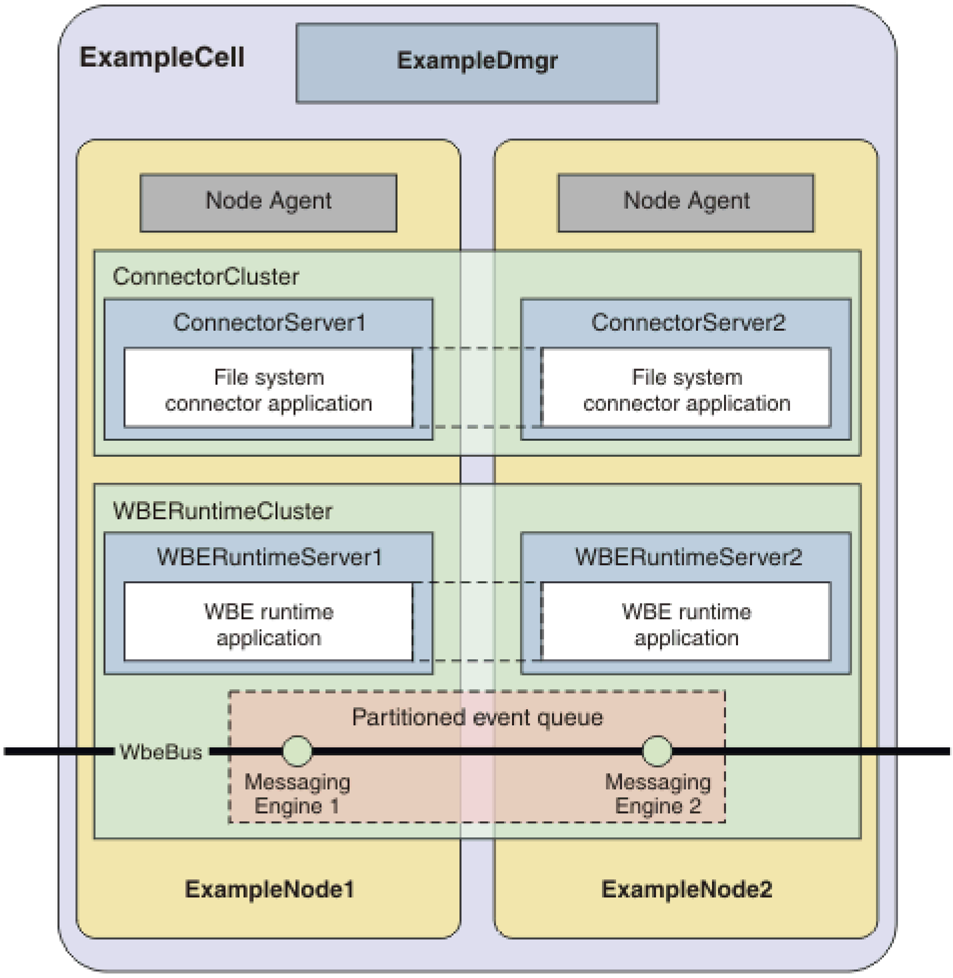


Figure 7: Example cluster configuration.

The topology consists of:

- A single cell:

Name	`ExampleCell`

- A deployment manager on the host `dmgr.example.com` port:

Name	`ExampleDMgr`
WebSphere Application Server profile name	`dmgr`
Host name	`dmgr.example.com`
Bootstrap port	`9809`

- Two clusters within the cell, each with two servers:

Name	`ConnectorCluster`
Description	Cluster to which the File System connector is

	deployed
Server names	`ConnectorServer1, ConnectorServer2`

Server host names and bootstrap ports	`host1.example.com:2809,`
	`host2.example.com:2809`

Name	`WBERuntimeCluster`
Description	Cluster to which the WebSphere Business Events Runtime is deployed. The cluster is a member of the bus named WbeBus
Server names	`WBERuntimeServer1, WBERuntimeServer2`

Server host names and bootstrap ports	`host1.example.com:2810,`
	`host2.example.com:2810`

You must configure the File System connector on the machine where the deployment manager exists. On that machine, the following commands apply:

To disable the existing File System connector for this scenario, use the following command:

```
disableTechnologyConnectors.bat -type=file -profileName=dmgr
-host=dmgr.example.com -port=9809 -clusterName=WBERuntimeCluster
```

To deploy the new File System connector for this scenario, use the following command:

```
deployConnectors.bat -type=file -profileName=dmgr -host=dmgr.example.com
-port=9809 -clusterName=ConnectorCluster
-providerURL=corbaloc:iiop:host1.example.com:2810,iiop:host2.example.com:2
810 -messagingClusterName=WBERuntimeCluster -outputDir=C:\temp\connectors
```

To update the new File System connector for this scenario, use the following command:

```
deployConnectors.bat  -type=file  -profileName=dmgr  -host=dmgr.example.com
-port=9809                                    -clusterName=ConnectorCluster
-providerURL=corbaloc:iiop:host1.example.com:2810,iiop:host2.example.com:2
810  -messagingClusterName=WBERuntimeCluster  -outputDir=C:\temp\connectors
-update
```

The HTTP connector

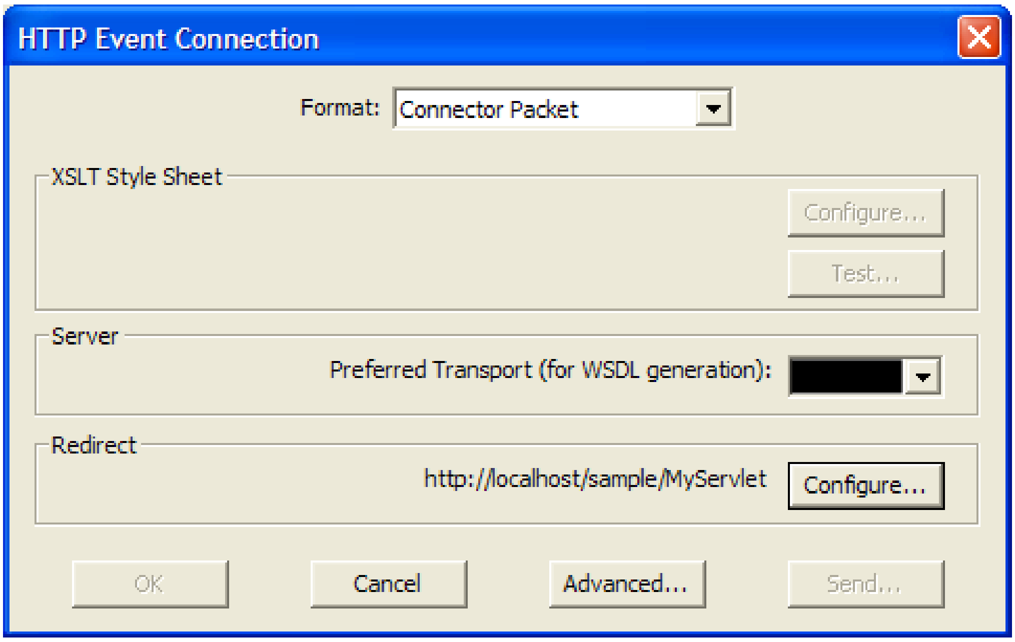
You can use the HTTP connector to receive events and send actions by using HTTP. The HTTP connector also supports sending REST based actions.

Consuming events by using the HTTP connector

The HTTP connector supports a number of formats for events consumed using HTTP. The format is specified in the connection settings for the event by using the Design Data tool.

Configuring an event to use the HTTP connector

In Design Data, right-click the event and select **Event Properties**. Select the **Connection** tab then **HTTP Connection** and then click **Configure**. The following screen capture shows the HTTP Event Connection window:



HTTP Event Connection

Format: Connector Packet

XSLT Style Sheet

Server

Preferred Transport (for WSDL generation):

Redirect

http://localhost/sample/MyServlet

You can configure the following settings:

Format

The format of the event data that is expected by the connector. The HTTP connector expects the event to be received in the format specified. There are three formats:

Connector Packet

The format must match the format of the WebSphere Business Events event packet schema.

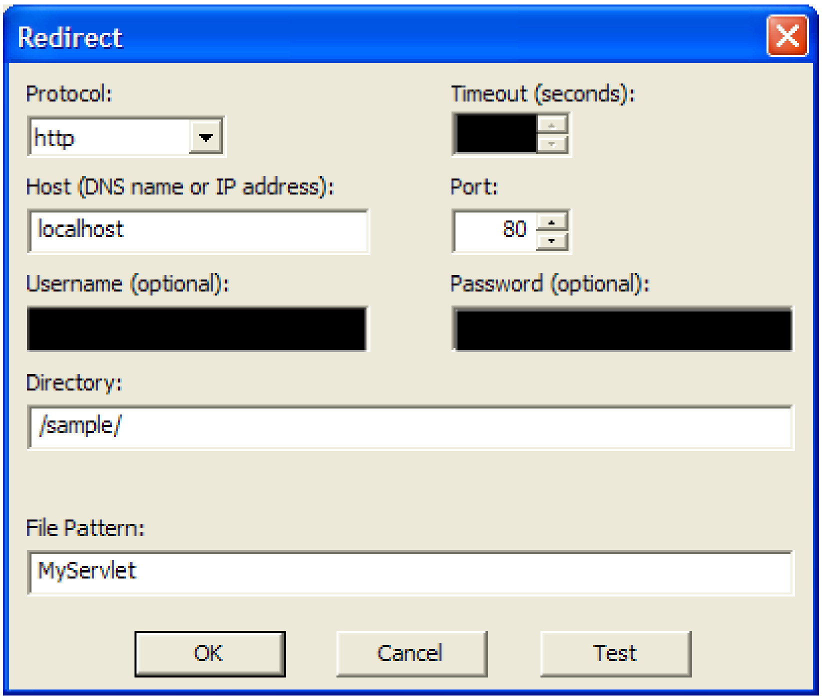
Custom XML (XSL input)
The event data can be any custom xml document. The specification of an XSL style sheet is required. The style sheet must transform the custom xml to the WebSphere Business Events event packet format.

HTML Form
The event data is in HTML form (application/x-www-form-urlencoded) format.

XSLT Style Sheet
Required if **Custom XML** is specified as the format. Specifies the XSL style sheet used to transform the custom xml into the WebSphere Business Events event packet format.

Server
This value is not used. The HTTP connector can be invoked via HTTP or HTTPS regardless of this setting.

Redirect
The components of a URL send as the location in an HTTP redirect (HTTP 302 header) in response to receiving the event at the connector. Click **Configure**. The following screen capture shows the Redirect window:



Protocol: http

Timeout (seconds): [redacted]

Host (DNS name or IP address): localhost

Port: 80

Username (optional): [redacted]

Password (optional): [redacted]

Directory: /sample/

File Pattern: MyServlet

You can configure the following redirect settings:

Protocol
The protocol of the redirect URL.

Timeout
This field is not used by the HTTP connector for redirects.

Host
The host of the redirect URL.

Port
The port of the redirect URL.

Username and Password
These fields are not used by the HTTP connector for redirects.

Directory
The directory structure of the path of the redirect URL.

File Pattern
The file name of the redirect URL.

You can configure the following advanced settings, by clicking **Advanced**:

Topic Type
Specifies how messages are transmitted between the connector and the WebSphere Business Events Runtime.

Express
Indicates that messages are non-persistent and that the WebSphere Business Events Runtime does not make use of durable subscriptions when consuming the messages.

Reliable
Indicates that messages are persistent and that the WebSphere Business Events Runtime makes use of durable subscriptions when consuming the messages.

Compress
Specifies whether the data in the event messages sent by the connector to the WebSphere Business Events Runtime is compressed.

Requirements for sending the HTTP event when using Connector Packet format

Connector packet format events must be sent to the URI of the HTTP connector:

`http://server:port/wbecsp/HTTPEventConnector`

The request must be sent using HTTP POST with the event packet xml in the request body. Set the HTTP `Content-Type` header to `text/xml`.

Requirements for sending the HTTP event when using custom XML format

Custom xml format events must be sent to the URI of the HTTP connector with a request parameter specifying the name of the event as follows:

`http://server:port/wbecsp/HTTPEventConnector?_event=<event_name>`

where *`<event_name>`* is the name of the event

The request must be sent using HTTP POST with the custom xml in the request body. Set the HTTP `Content-Type` header to `text/xml`.

Requirements for sending the HTTP event when using HTML Form format

HTML form format events must be sent to the URI of the HTTP connector:

```
http://server:port/wbecsp/HTTPEventConnector
```

using the HTTP POST method. The contents of the form data must include the following parameters:

`_event`

The name of the event.

`_object`

The name of the event object within the event. Only one event object is allowed for an event send using the HTML Form format.

In addition, specify the parameters for the names and values of the fields within the event object.

The following example shows an HTML form for sending an event called SampleEvent containing a event object called Person where the Person object has three fields, firstname, surname, and age:

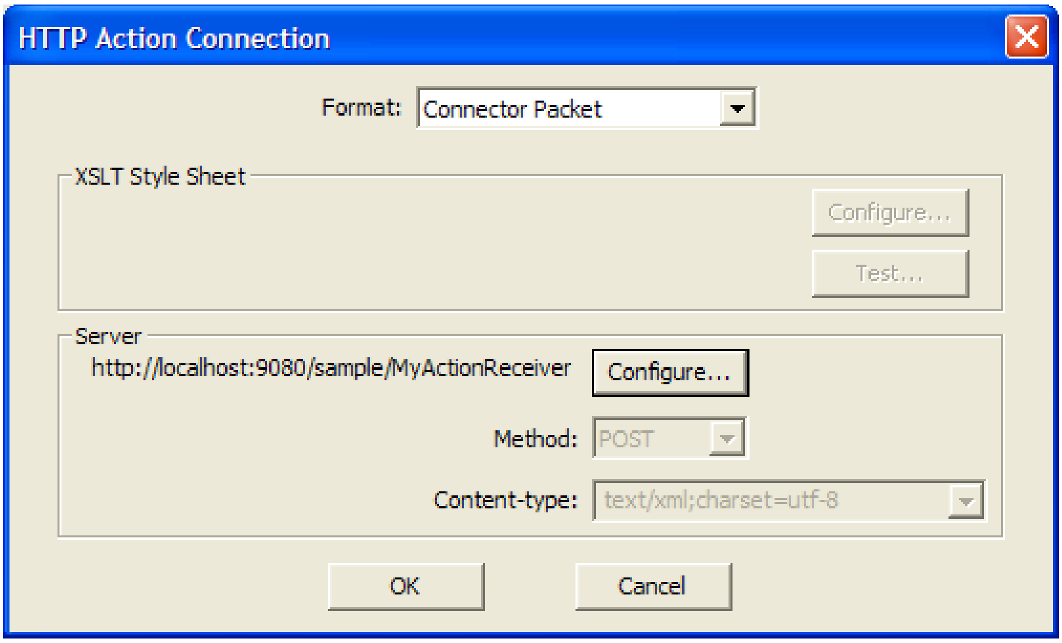
```
<form method="post"
action="http://localhost:9080/wbecsp/HTTPEventConnector">
<input type="text" name="firstname">
<input type="text" name="surname">
<input type="text" name="age">
<input type="hidden" name="_event" value="SampleEvent">
<input type="hidden" name="_object" value="Person">
</form>
```

Producing actions by using the HTTP connector

The HTTP connector supports a number of formats for actions sent using HTTP including REST. The format is specified in the connection settings for the action by using the Design Data tool.

Configuring an action to use the HTTP connector with an HTTP connection

In Design Data, right-click the action and select **Action Properties**. Select the **Connection** tab then **HTTP Connection** and then click **Configure**. The following screen capture shows the HTTP Action Connection window:



You can configure the following settings:

Format
The format of the action data that is sent by the HTTP connector. Choose between:

CBE (Monitor)
The action data is formatted in Common Business Event format.

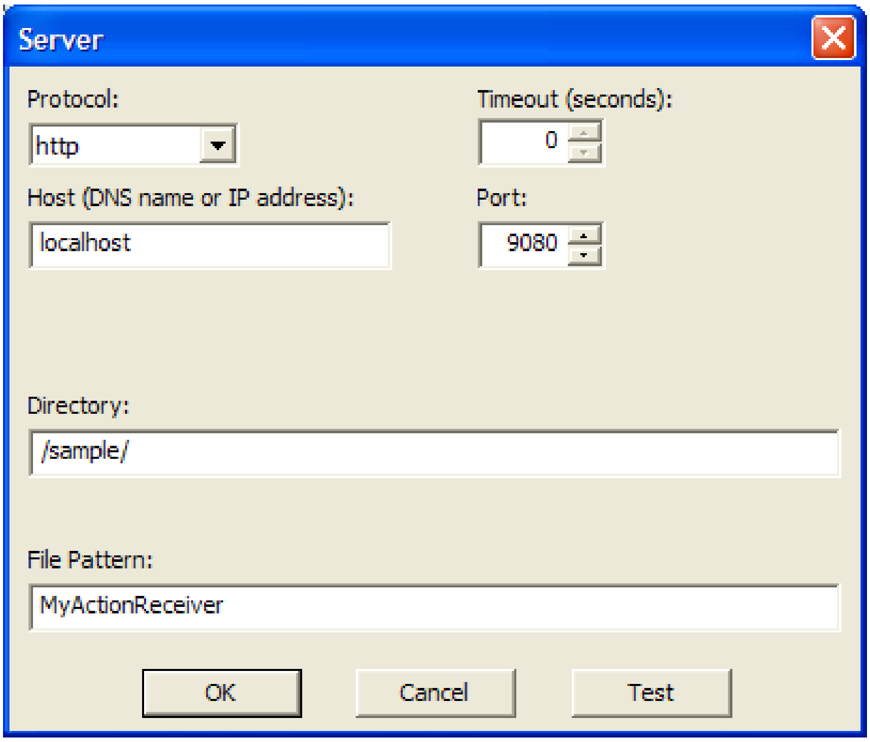
Connector Packet
The action data is formatted in the WebSphere Business Events action packet format.

Custom XML (XSL output)
The action packet is transformed using an XSL style sheet into a custom xml format. The specification of the XSL style sheet is required.

HTML Form
The action data is formatted into HTML form (application/x-www-form-urlencoded) format.

XSLT Style Sheet
Required if Custom XML is specified as the format. Specifies the XSL style sheet used to transform the WebSphere Business Events action packet format into the required custom xml.

Server
To specify the URL to which the connector sends the action data and how it is sent. Click **Configure**. The following screen capture shows the Server window:



Protocol: http

Timeout (seconds): 0

Host (DNS name or IP address): localhost

Port: 9080

Directory: /sample/

File Pattern: MyActionReceiver

You can configure the following Server settings:

Protocol
The protocol of the URL.

Host
The host of the URL.

Port
The port of the URL.

Directory
The directory structure of the path of the URL

File Pattern
The file name of the URL.

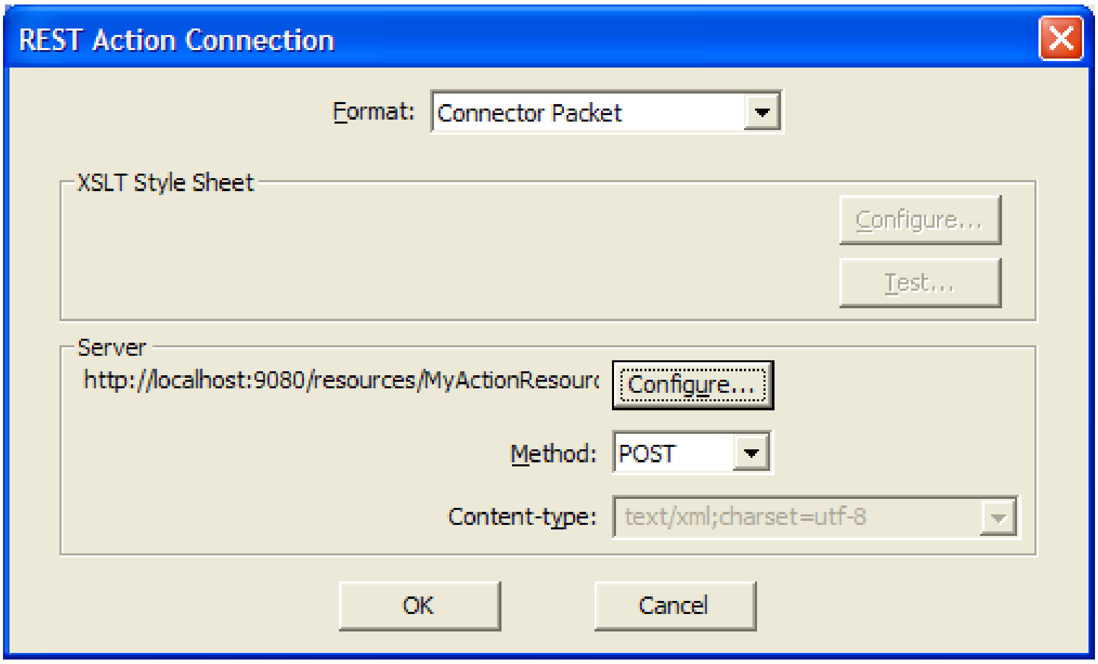
Timeout
The timeout, in seconds, for the request made by the HTTP connector. A value of `0` indicates no timeout.

Method
Only used when the format is specified as HTML Form. Indicates the HTTP method used in the request made by the connector.

Content-type
Only used when the format is specified as Custom XML. The value for the Content-Type header send in the request made by the connector.

Configuring an action to use the HTTP connector with a REST connection

In Design Data, right-click the action and select **Action Propertie***s.* Select the **Connection** tab then **REST Connection** and then click **Configure**. The following screen capture shows the REST Action Connection window:


REST Action Connection

Format: Connector Packet

XSLT Style Sheet

Server

http://localhost:9080/resources/MyActionResourc

Method: POST

Content-type: text/xml;charset=utf-8

You can configure the following settings:

Format

The format of the action data that is sent by the HTTP connector. Choose between:

Connector Packet

The action data is formatted in the WebSphere Business Events action packet format.

Custom XML (XSL output)

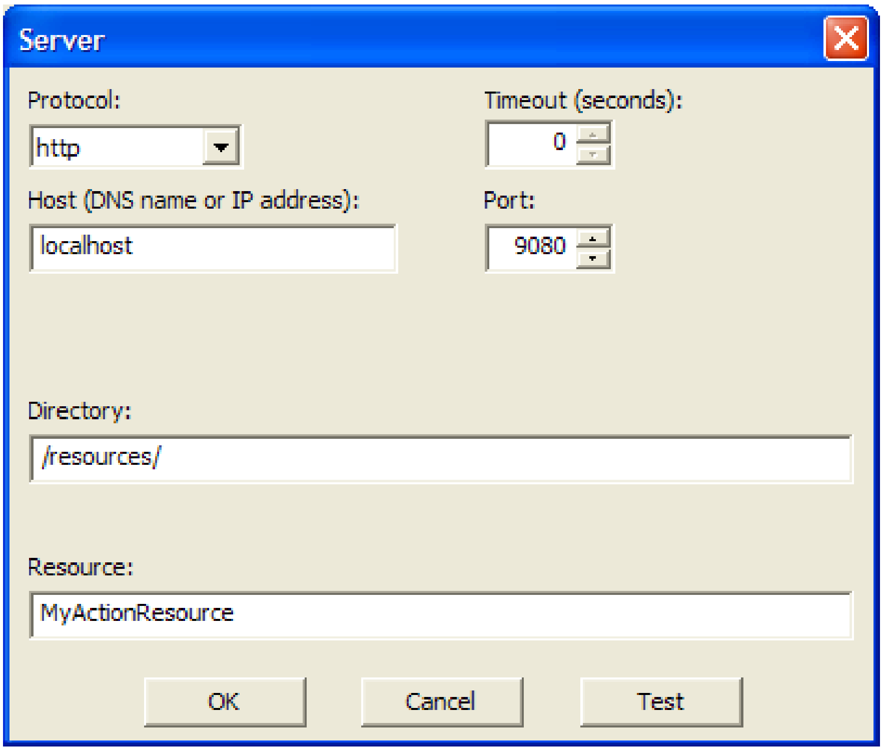
The action packet is transformed using an XSL style sheet into a custom xml format. The specification of the XSL style sheet is required.

XSLT Style Sheet

Required if Custom XML is specified as the format. Specifies the XSL style sheet used to transform the WebSphere Business Events action packet format into the required custom xml.

Server

To specify the URL to which the connector sends the action data and how it is sent. Click **Configure**. The following screen capture shows the Server window:



Protocol: http

Timeout (seconds): 0

Host (DNS name or IP address): localhost

Port: 9080

Directory: /resources/

Resource: MyActionResource

You can configure the following server settings:

Protocol
The protocol of the URL.

Host
The host of the URL.

Port
The port of the URL.

Directory
The directory structure of the path of the URL.

Resource
The resource name in the URL.

Timeout
The timeout, in seconds, for the request made by the HTTP connector. A value of `0` indicates no timeout.

Method

The HTTP method used in the request made by the connector. Supported methods are

- POST: the action data is contained in the body of a POST request
- PUT: the action data is contained in the body of a PUT request
- DELETE: a DELETE request is made to the specified URL. No data from the action is directly included in the request. The _rest template action object can be used to override values in the URL based on the current invocation.

Content-type

Only used when the format is specified as Custom XML. The value for the Content-Type header send in the request made by the connector.

Using the HTTP connector in a secure environment

Application security and the HTTP connector

If application server security is enabled, the HTTP connector requires a user role mapping to be specified to determine which users have permission to send events via HTTP to the connector. To specify the user role mapping by using the WebSphere Application Server administrative console, complete the following steps:

1. Under **WebSphere enterprise applications**, click the HTTP connector application, **wbecsphttpconnector.**
2. Under **Detailed Properties**, click **Security role to user/group mapping**.
3. Select the **HTTPEventConnectorUser** role and map one or more users to the role. To allow unrestricted access when security is enabled, map the role to the **Everyone** special subject.
4. Click **Save**.

Bus security and the HTTP connector

If bus security is enabled, the HTTP connector might require user credentials to be specified to send events to the WebSphere Business Events Runtime or consume actions from the WebSphere Business Events Runtime .

The user credentials used by the connector when sending events to the WebSphere Business Events Runtime can be specified by setting values for the following WebSphere Business Events properties:

```
as.director.server.jms.username
as.director.server.jms.password
```

After making changes to these properties, if the HTTP connector application has been deployed, it must be updated by running the deployConnectors command specifying the –update argument in addition to any other arguments required for your topology.

The user credentials used by the connector when consuming actions from the WebSphere Business Events Runtime can be specified by setting an authentication alias in the following activation specification definition:

```
wbecsp_http_as
```

Important: This activation specification is created during the deployConnectors process and must not be created manually.

Configuring connections when sending events to the WebSphere Business Events Runtime

See Configuring retry values for sending JMS messages for information regarding the handling of failed connections when the HTTP connector sends events to the WebSphere Business Events Runtime.

The JDBC connector

You can use the JDBC event connector to generate events by retrieving data from relational database tables defined as data connections. You can use the JDBC action connector to perform basic SQL functions (SELECT, INSERT, UPDATE, DELETE) against relational database tables. This SupportPac documentation uses the term *JDBC connector*, which is also known as the *RDBMS connector*.

To consume events or produce actions using the JDBC connector, you must create a datasource in Design Data. You must set a custom property for the datasource to define the JNDI name of a WebSphere Application Server data source. For more information, see Configuring the WebSphere Business Events datasource.

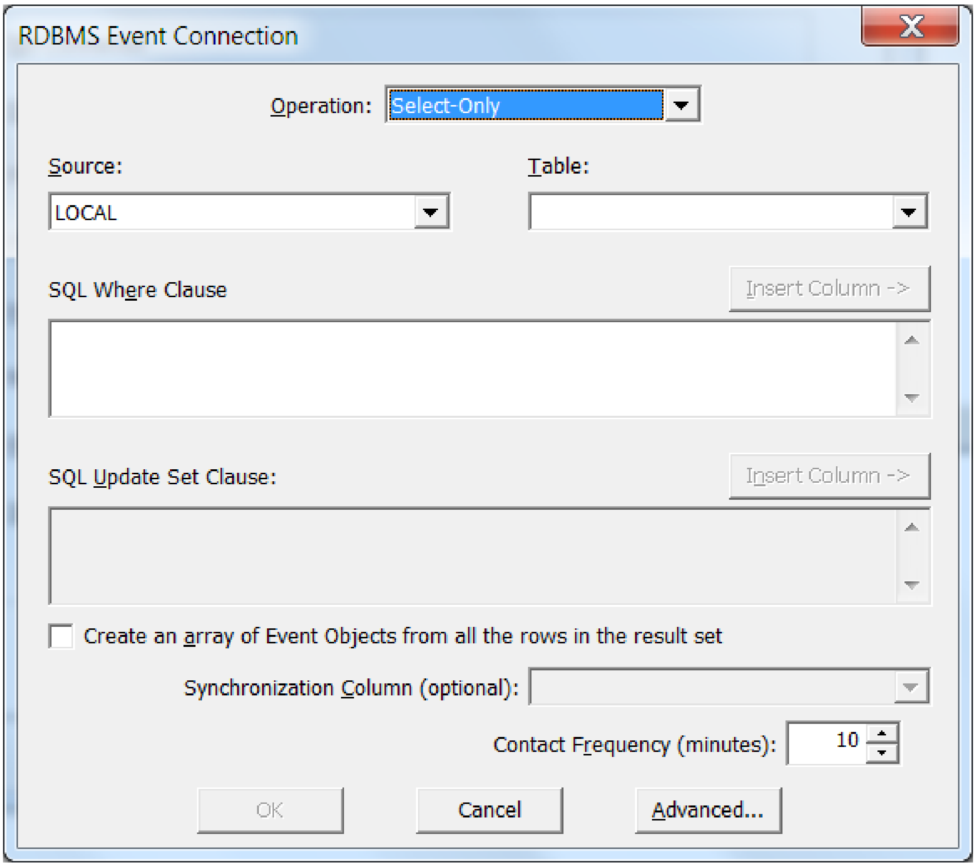
The relevant datasource must be configured in WebSphere Application Server. For more information see Configuring a data source using the administrative console in the *WebSphere Application Server information center*.

Consuming events by using the JDBC connector

You can configure the JDBC connector to monitor a relational database table for rows of data that match a specific WHERE clause. Events are generated that contain the information from a single row and a collection of rows. The JDBC connector can perform a standard SELECT, a SELECT & UPDATE or a SELECT & DELETE against the table, or it can also call a stored procedure.

Configuring an event to use the JDBC connector

In Design Data, right-click the event and select **Event Properties***.* Click the **Connection** tab then **RDBMS Connection** and then click **Configure***.* The following screen capture shows the **RDBMS Event Connection** configuration window:



Operation: Select-Only

Source: LOCAL

Table:

SQL Where Clause

SQL Update Set Clause:

□ Create an array of Event Objects from all the rows in the result set

Synchronization Column (optional):

Contact Frequency (minutes): 10

You can configure the following settings:

Operation
The type of operation that is to be performed against the database table.

Select-only
This operation will run a simple SELECT query against the data from a single relational database table. The data will not be modified during this operation.

Select & Update
Use this operation if you want to select data for one or more events from a single relational database table and then update that data. You can use this operation when you want to ensure that the same rows are not selected again, by updating the data so that it no longer matches the WHERE clause.

Select & Delete
Use this operation if you want to select data for one or more events from a single relational database table and then delete that data.

Call
The Call operation allows a predefined stored procedure to be executed periodically and its results to be used to generate events in WebSphere Business Events. This operation allows for more complex SQL statements or custom queries to be executed to generate events.

Only stored procedures that return a single result set are supported. Stored procedures returning single values or multiple result sets are not supported.

Source

The datasource name associated with the RDBMS, as defined under **Hosted Databases** in the Data Sources window of the workspace

Table

The name of the table or view. Only one table can be selected.

SQL Where Clause

The selection criteria to identify the rows that are selected. Use standard WHERE clause syntax for the DBMS being referenced, but leave out the WHERE keyword.

SQL Update Set Clause

For use with the Select & Update Operation, the SET clause used to identify columns to be updated along with their updated values. Use standard SET clause syntax for the DBMS being referenced, but leave out the SET keyword. The value is updated after the event has been generated. Use this setting if you want to stop generating an event from a row after the event was initially generated.

Synchronization Column (optional)

If you do not specify an value for this column, the Select & Update operation might lock the database table to ensure that rows are not inserted between the Select operation and the Update operation. The lock might impact other applications that are accessing that table. To avoid this issue, you can provide a synchronization column that is used by the connector to track table rows that have been retrieved as events and columns that have been updated, as identified in the SQL Update set clause field. The column selected must be a varchar field of 256 bytes or larger.

Contact Frequency (minutes)

The interval, in minutes, that the connection waits before performing queries (default is 10 minutes).

Create an array of Event Objects from all the rows in the result set

If this field is selected, then a single event containing multiple Event Object occurrences, one for each retrieved row, is created.

If the field is not selected, then a separate event, containing one Event Object occurrence, is created for each retrieved row.

You can configure the following advanced settings, by clicking **Advanced**:

Topic Type

Specifies how messages are transmitted between the connector and the WebSphere Business Events Runtime.

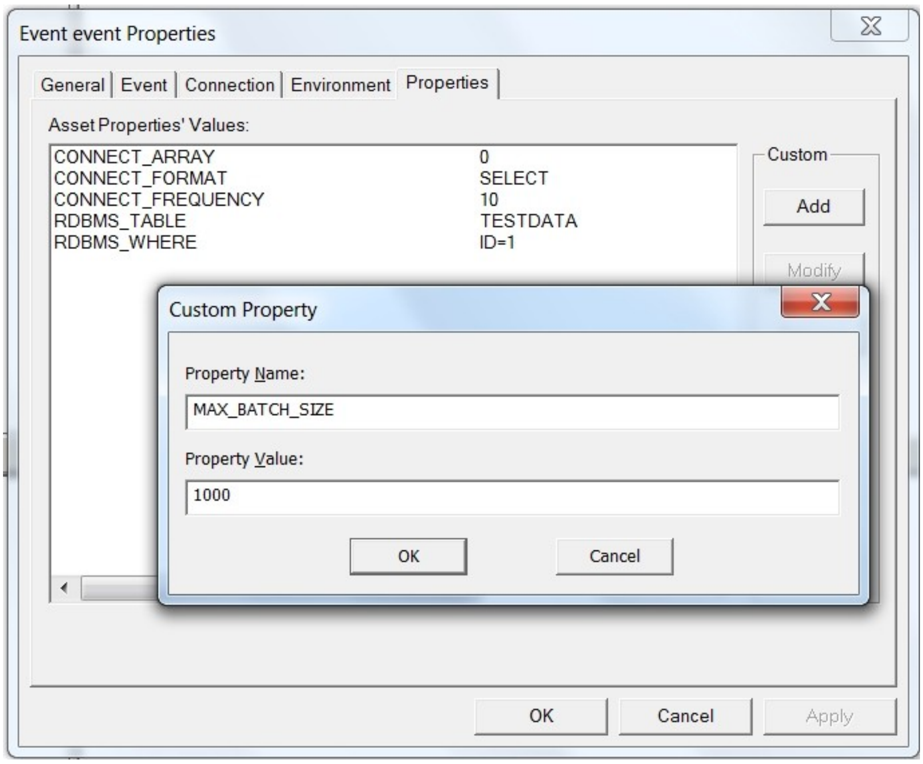
Express
Indicates that messages are non-persistent and that the WebSphere Business Events Runtime does not make use of durable subscriptions when consuming the messages.

Reliable
Indicates that messages are persistent and that the WebSphere Business Events Runtime makes use of durable subscriptions when consuming the messages.

Compress
Specifies whether the data in the event messages sent by the connector to the WebSphere Business Events Runtime is compressed.

Configuring custom properties for the JDBC event connector

In Design Data, right-click the event and select **Event Properties**. Click the **Properties** tab and then click **Add**. The following screen capture shows the Event Properties window, with an example of a custom property for an event named `Event1`:



Asset Properties' Values:

CONNECT_ARRAY	0
CONNECT_FORMAT	SELECT
CONNECT_FREQUENCY	10
RDBMS_TABLE	TESTDATA
RDBMS_WHERE	ID=1

Property Name: MAX_BATCH_SIZE

Property Value: 1000

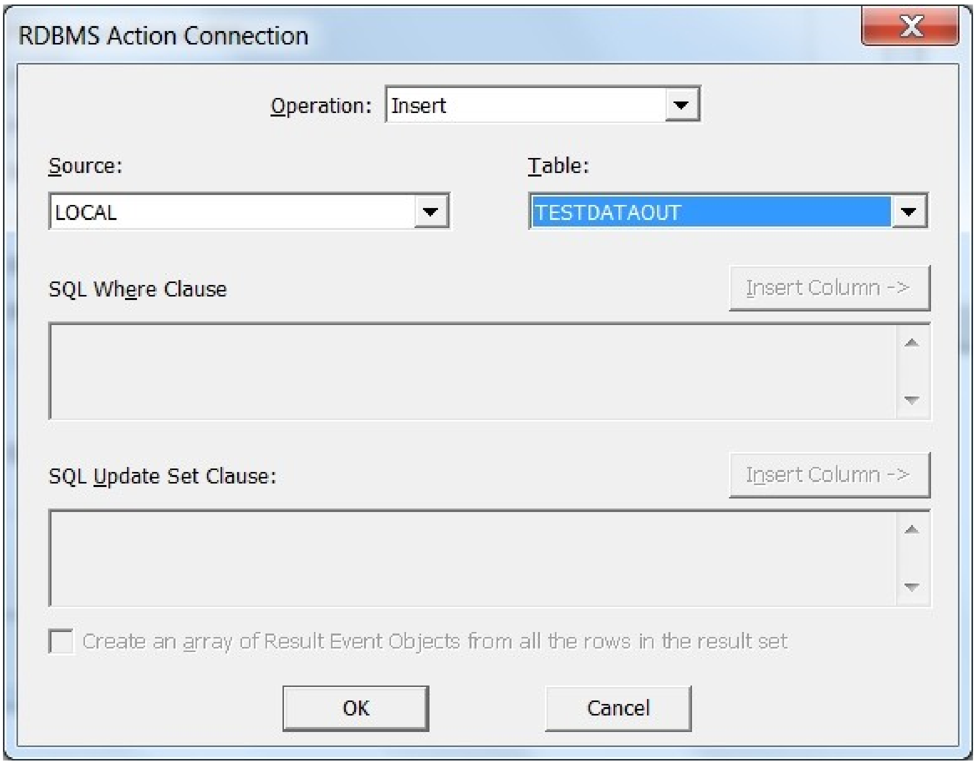
MAX_BATCH_SIZE
Specifies the maximum number of rows to include in each batch retrieved from the database. This setting will not limit the number of rows processed by the connector during any polling interval. The default value is 500.

Producing actions by using the JDBC connector

The JDBC connector can be configured to send the action to change a relational database table by using basic SQL (SELECT, INSERT, UPDATE, DELETE). The UPDATE operation allows a further extension enabling you to INSERT the action data if there are no matching rows to UPDATE.

Configuring an action to use the JDBC connector

In Design Data, right-click the action and select **Action Properties**. Click the **Connection** tab then **RDBMS Connection** and then click **Configure**. The following screen capture shows the **RDBMS Action Connection** configuration window:



RDBMS Action Connection

Operation: Insert

Source: LOCAL

Table: TESTDATAOUT

SQL Where Clause

SQL Update Set Clause:

□ Create an array of Result Event Objects from all the rows in the result set

You can configure the following settings:

Operation

Indicate the SQL command that the connection performs when fired:

Delete: Delete one or more rows in the selected table.

Insert: Insert one or more rows into the selected table (this is the default option).

Select: Retrieve one or more rows from the selected table.

Update: Update one or more rows in the selected table. You can also set Update so that an Insert occurs if no matching rows are found.

Source

The datasource name associated with the RDBMS, as defined under **Hosted Databases** in the Data Sources window of the workspace.

Table

The name of the table or view. Only one table can be updated. A value in the Table field of the _Rdbms Action Object overrides this setting.

SQL Where Clause

The selection criteria to identify the rows that are to be updated. Use standard WHERE clause syntax for the DBMS being referenced, but leave out the WHERE keyword. A value in the Where field of the _Rdbms Action Object overrides this setting.

SQL Update Set Clause

For use with the Update Operation, the SET clause used to identify columns to be updated along with their updated values, but leave out the SET keyword. A value in the Update field of the _Rdbms Action Object overrides this setting.

Create an array of Result Event Objects from all the rows in the result set

This applies to the Select operation only. Indicates that, if multiple rows are retrieved, a single Result Event with multiple Result Event Objects (one for each row) is returned. If not selected, the default (a separate Result Event for each retrieved row) is returned.

Perform an insert if the Where clause finds no rows to update

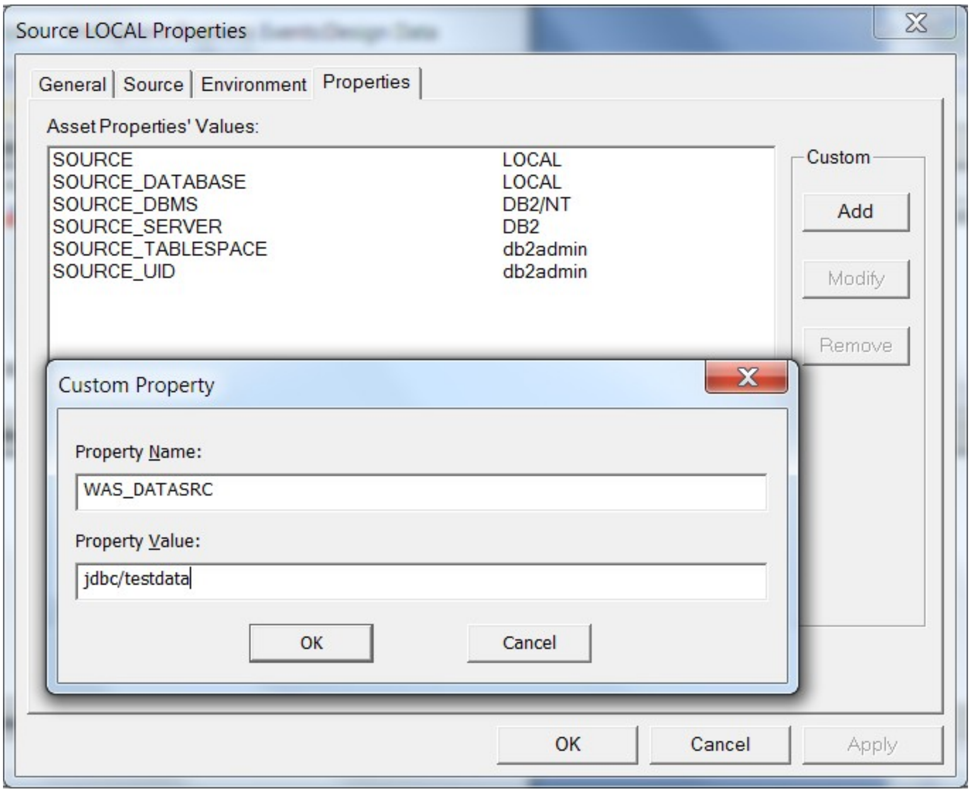
This applies to the Update operation only. Indicates that if no rows are found that match the WHERE clause of the UPDATE, then the Action is treated as an INSERT

Configuring the WebSphere Business Events datasource

The ability for a WebSphere Business Events datasource to specify a WebSphere Application Server data source is not natively supported by Design Data. The JDBC connector supports a WebSphere Application Server data source so that the connector can take advantage of the transaction management system that is used by WebSphere Application Server.

You need to specify a custom property for the WebSphere Business Events datasource to tell the JDBC connector which WebSphere Application Server data source to use when the connector is processing events and actions.

In Design Data, right-click the datasource and select **Source Properties**. Click the **Properties** tab and then click **Add**. The following screen capture shows the Source Properties window, with an example of a custom property for a source named `Local`:



Asset Properties' Values:

SOURCE	LOCAL
SOURCE_DATABASE	LOCAL
SOURCE_DBMS	DB2/NT
SOURCE_SERVER	DB2
SOURCE_TABLESPACE	db2admin
SOURCE_UID	db2admin

Property Name: WAS_DATASRC

Property Value: jdbc/testdata

WAS_DATASRC
The JNDI lookup name of the datasource for the connector to use when processing events and actions. This name must match the JDNI name specified when creating the WebSphere Application Server data source.

Handling errors for full queues

When the connector receives errors relating to full queues, the connector delays the resending of the last batch. If errors are detected when the batch is resent, the delay is increased. When successful sending of the batch occurs, attempts are made to reduce the delay and increase the batch processing rate again. If further errors are detected in response to an increased rate, the rate is reduced again and maintained at the lower level for a longer period.

Ensuring high availability of JDBC processing

When the JDBC connector is deployed to a cluster, it is enabled for high availability. The connector is active on one member of the cluster at any one time. If the server with the active connector fails, the connector automatically becomes active on another server within the same cluster.

The JMS connector

You can use the JMS connector to receive events from message queues and topics, and to send actions to message queues and topics. This SupportPac documentation uses the term *JMS connector*, which is also known as the *Message Queue connector*. When you define events and actions to have a Message Queue connection, the events and actions are serviced by a JMS connector.

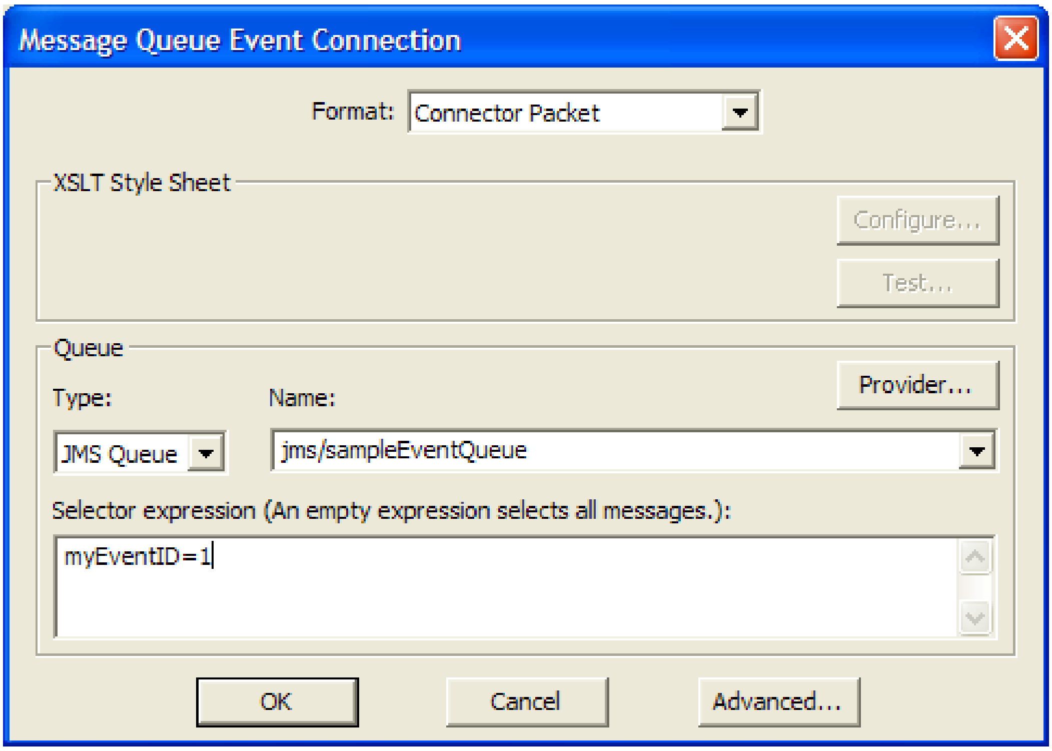
Consuming events by using the JMS connector

The JMS connector can consume events by getting messages from queues or topics. To consume events from message queues or topics you must create the appropriate activation specifications. For more details, see Using activation specifications when consuming JMS events.

All messages received by the JMS connector must be JMS TextMessages. Other JMS message types are not supported.

Configuring an event to use the JMS connector

In Design Data, right-click the event and select **Event Properties**. Click the **Connection** tab then **Message Queue Connection** and then click **Configure**. The following screen capture shows the Message Queue Event Connection window:



Message Queue Event Connection

Format: Connector Packet

Type: JMS Queue

Name: jms/sampleEventQueue

Selector expression (An empty expression selects all messages.): myEventID=1

You can configure the following settings:

Format
The format of the event data that is expected by the connector. The JMS connector expects the event data to be received in the format specified. There are four formats:

Alert (WBM)
Use this format when receiving business alerts from WebSphere Business Monitor. The format must conform to the WebSphere Business Monitor Touchpoint template provided by WebSphere Business Events.

CBE (Monitor)
Use this format when receiving events from WebSphere Business Monitor. The format must conform to the Common Business Event (CBE) payload schema exported from Design Data.

Connector Packet
The format must match that of the WebSphere Business Events event packet schema.

Custom XML (XSL input)
The event data can be any custom xml document. The specification of an XSL style sheet is required. The style sheet must transform the custom xml to the WebSphere Business Events event packet format.

XSLT Style Sheet
Required if Custom XML is specified as the format. The XSL style sheet used to transform the custom xml into the WebSphere Business Events event packet format.

Queue - Type
The type of the destination to consume from, either JMS Queue or JMS Topic.

Queue - Name
The JNDI name of the destination to consume from. The destination must be defined within the application server environment at a scope visible to the connector application. For example if the connector application is installed in a cluster, the destination must be defined at cell scope.

Queue – Selector expression
The message selector expression to be used when consuming messages from the destination. If multiple events are defined to be consumed from the same destination, each must specify a message selector expression that uniquely identifies the event amongst all messages.

You can configure the following provider settings, by clicking **Provider**:

No information from the provider panel is used in consuming events using the Connector SupportPac JMS connector. Messages are consumed by means of MDBs which require activation specifications to be defined within the application server. For more information, see Using activation specifications when consuming JMS events.

You can configure the following advanced settings, by clicking **Advanced**:

Topic Type

Specifies how messages are transmitted between the connector and the WebSphere Business Events Runtime.

Express

Indicates that messages are non-persistent and that the WebSphere Business Events Runtime does not make use of durable subscriptions when consuming the messages.

Reliable

Indicates that messages are persistent and that the WebSphere Business Events Runtime makes use of durable subscriptions when consuming the messages.

Compress

Specifies whether the data in the event messages sent by the connector to the WebSphere Business Events Runtime is compressed.

Using activation specifications when consuming JMS events

The JMS connector requires the use of activation specifications when consuming events. These activation specifications must be created manually before deploying the JMS connector application. The activation specifications define where and how messages are consumed.

In order for the connector application to correctly use an activation specification for an event, it must be able to resolve the activation specification JNDI name. The Enhanced Connector Applications SupportPac supports two ways in which the JNDI name can be resolved.

By default, the JNDI name of the activation specification is assumed to be of the form:

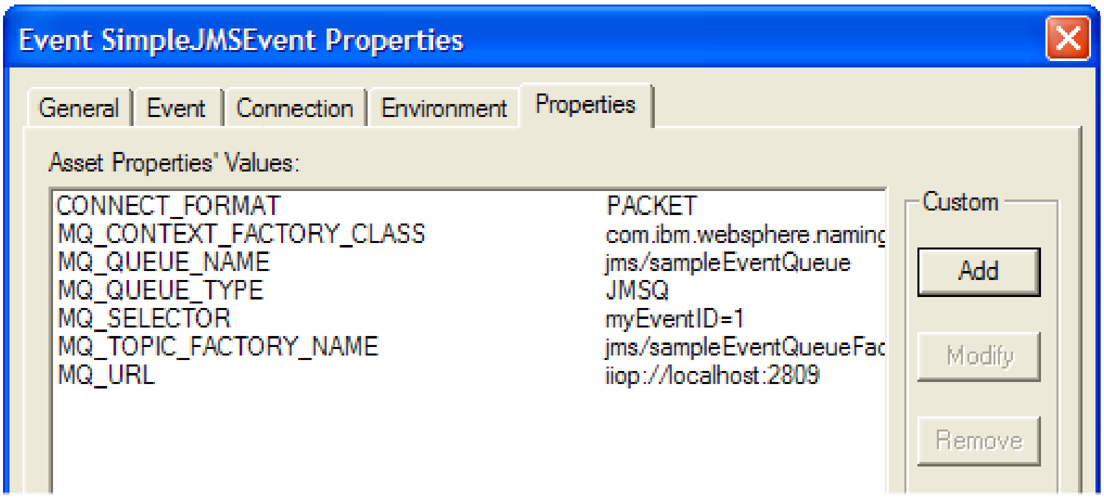
`jca/wbejmsevent/`*`<eventName_with_underscores>`*

where *`<eventName_with_underscores>`* is the name of the event you wish to receive messages for, with any spaces in the name replaced by underscores.

For example, to receive messages for an event called CancelledOrder, you must specify the JNDI name of the activation specification as `jca/wbejmsevent/CancelledOrder`. To receive messages for an event called New Customer, you must specify the JNDI name of the activation specification as `jca/wbejmsevent/New_Customer`.

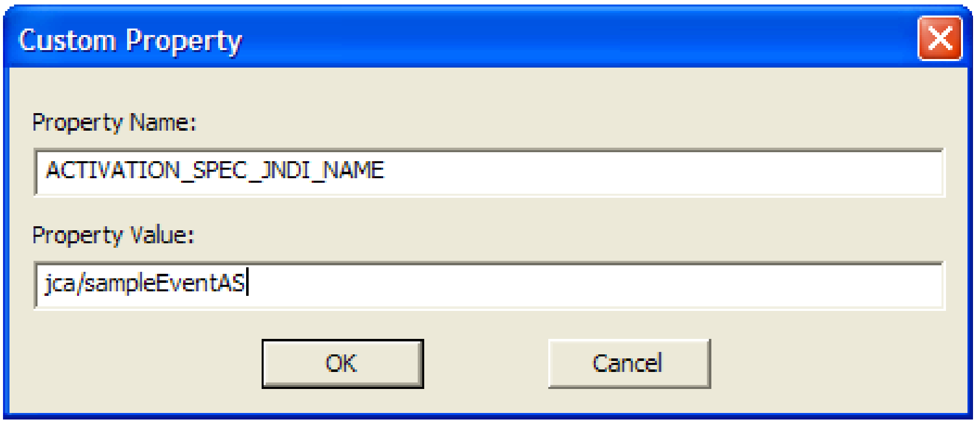
If you want to use an alternative activation specification name, or require the same activation specification to be used by multiple events, you can specify the name of the activation specification within the project. To set the activation specification name, complete the following steps:

1. Open the project in Design Data.
2. Open the properties for the event you want to set the activation specification name for.
3. Select the **Properties** tab. The following screen capture shows the Event SimpleJMSEvent Properties window:



CONNECT_FORMAT	PACKET
MQ_CONTEXT_FACTORY_CLASS	com.ibm.websphere.naming
MQ_QUEUE_NAME	jms/sampleEventQueue
MQ_QUEUE_TYPE	JMSQ
MQ_SELECTOR	myEventID=1
MQ_TOPIC_FACTORY_NAME	jms/sampleEventQueueFac
MQ_URL	iiop://localhost:2809

4. Click **Add** to add a new custom property.
5. Enter `ACTIVATION_SPEC_JNDI_NAME` as the name of the property.
6. Enter the required activation specification JNDI name as the value. The following screen capture shows the Custom Property window:



Property Name: ACTIVATION_SPEC_JNDI_NAME

Property Value: jca/sampleEventAS

Overriding activation specification properties

The following values from the connector properties for an event will override values defined in the activation specification:

- The type of the destination
- The JNDI name of the destination
- The message selector value

Therefore, if necessary, a single activation specification can be used to consume events from multiple destinations, or consume multiple events from the same destination, each with a different message selector value to uniquely identify the messages.

Further information

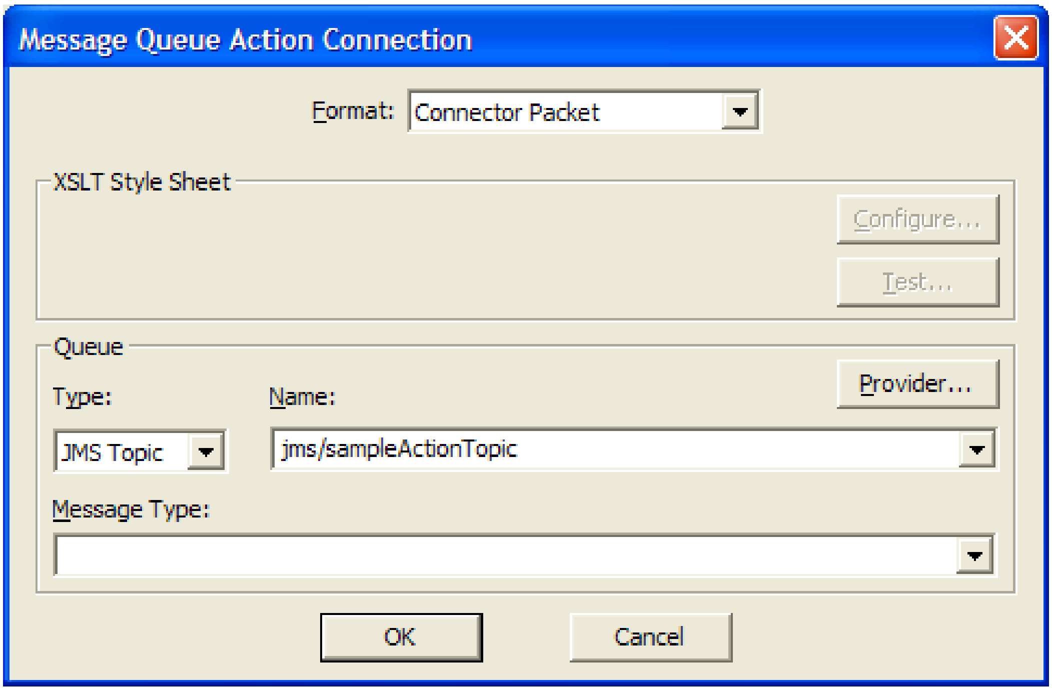
For more information about configuring activation specifications, see Activation specification collection in the *WebSphere Application Server v7 information center*.

Producing actions by using the JMS connector

The JMS connector, also known as the Message Queue connector, can produce actions by putting messages to JMS queues or topics. All messages produced will be JMS TextMessages.

Configuring an action to use the JMS connector

In Design Data, right-click the action and select **Action Properties**. Select the **Connection** tab then **Message Queue Connection** and then click **Configure**. The following screen capture shows the Message Queue Action Connection window:



Message Queue Action Connection

Format: Connector Packet

Type: JMS Topic

Name: jms/sampleActionTopic

Message Type:

You can configure the following settings:

Format
The format of the action data that is sent by the JMS connector. Choose between:

CBE (Monitor)
The action data is formatted in Common Business Event format.

Connector Packet
The action data is formatted in the WebSphere Business Events action packet format.

Custom XML (XSL output)
The action packet is transformed using an XSL style sheet into a custom xml format. The specification of the XSL style sheet is required.

XSLT Style Sheet
Required if Custom XML is specified as the format. Specifies the XSL style sheet used

to transform the WebSphere Business Events action packet format into the required custom xml.

Queue - Type

The type of the destination to send the action to, either JMS Queue or JMS Topic.

Queue - Name

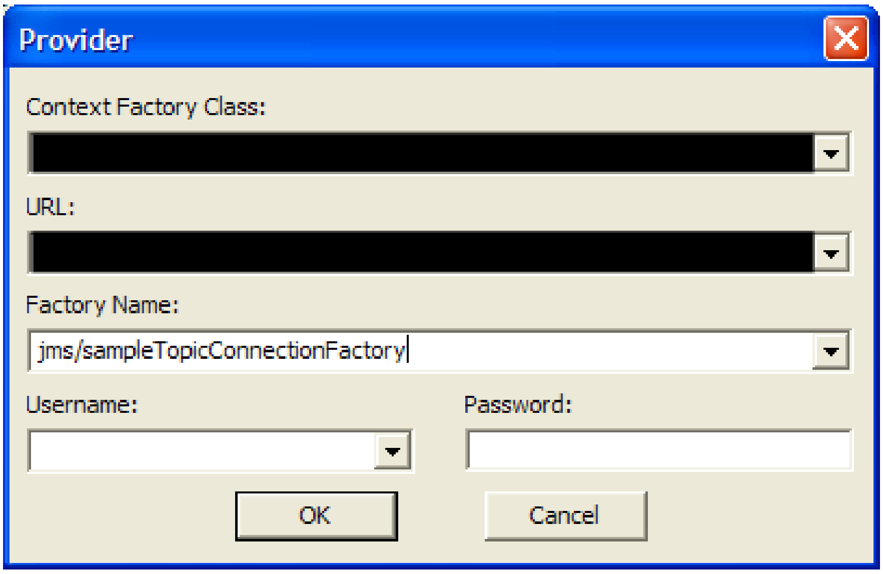
The JNDI name of the destination to send the action to. The destination must be defined within the application server environment at a scope visible to the connector application. For example if the connector application is installed in a cluster, the destination must be defined at cell scope.

Queue – Message Type

The value for the `JMSType` header field in message sent by the connector. If the message format is CBE (Monitor) then the value of the field should be set to `CREATE_EVENTS_NOTIFICATION_V20`.

You can configure the following provider settings, by clicking **Provider**:

The following screen capture shows the Provider window:



Provider

Context Factory Class: [redacted]

URL: [redacted]

Factory Name: jms/sampleTopicConnectionFactory

Username:

Password:

Context Factory class

This value is not used by the Enhanced Connector Applications SupportPac JMS connector.

URL

This value is not used by the Enhanced Connector Applications SupportPac JMS connector.

Factory name

The JNDI name of the JMS connection factory used to create connections when sending actions. The connection factory must be defined within the application server environment at a scope visible to the connector application. For example if the connector application is installed in a cluster, the connection factory must be defined at cell scope.

Username
The user name used when creating a connection.

Password
The password used when creating a connection.

Using the JMS connector in a secure environment

If bus security is enabled, the JMS connector might require user credentials to be specified to publish or consume messages to destinations. The location for specifying credentials varies based on the point of communication as follows:

Security when sending events to by the WebSphere Business Events Runtime

The user credentials used by the connector when sending events to the WebSphere Business Events Runtime can be specified by setting values for the following WebSphere Business Events properties:

```
as.director.server.jms.username
as.director.server.jms.password
```

After making changes to these properties, if the JMS connector application has been deployed, it must be updated by running the **`deployConnectors`** command specifying the **`–update`** argument in addition to any other arguments required for your topology.

Security when consuming actions from the WebSphere Business Events Runtime

The user credentials used by the connector when consuming actions from the WebSphere Business Events Runtime can be specified by setting an authentication alias in the following activation specification definition:

```
wbecsp_jms_as
```

Important: This activation specification is created by during the **`deployConnectors`** process and must not be manually created.

Security when consuming event messages from an non-WebSphere Business Events destination

The user credentials used by the JMS connector when consuming event messages from a destination can be specified by setting an authentication alias in the activation specification used for the event. See Using activation specifications when consuming JMS events for further details regarding the use of activation specifications when consuming JMS events.

Security when sending action messages from the JMS connector

The user credentials used by the JMS connector when sending action messages can be specified within the Provider section of the action’s properties within the Design Data tool.

Configuring retry values for sending JMS messages

When the JMS or HTTP connectors send events to the WebSphere Business Events Runtime, or when the JMS connector sends actions to a queue or topic, JMS connections are established. If the connections cannot be established, the connectors will retry for 2 minutes. The period of time spent in retrying to establish a connection can be overridden by setting one the following WebSphere Business Events properties:

`wbe.csp.jms.connection.max.retry.time`

The period of time, in seconds, that the connectors will retry to establish a connection.

`wbe.csp.jms.connection.max.retry.attempts`

The maximum number of attempts that the connectors will make to establish a connection. Setting this value will override the retry time value.

After changing these properties, if the connector applications have been deployed, you must update the connector applications by running the **`deployConnectors`** command specifying the **`-update`** argument in addition to any other arguments required for your topology.

Configuring the rollback option for the JMS connector

The information in this section applies to SupportPac Version 2.0.0.1 and higher.

When the JMS connector reaches the limit of retries for sending a JMS message, by default it logs an error message containing the contents of the message that failed to send.

If you prefer, you can configure the JMS connectors to roll back the event or action message they have received in response to the retry limit being reached. To enable this option, set the following WebSphere Business Events property with a value of `true`:

`wbe.csp.jms.rollback.upon.failed.delivery`

After you add this property, you must restart the WebSphere Business Events runtime application. If the connector applications have been deployed, you must update the connector applications by running the **`deployConnectors`** command specifying the **`-update`** argument in addition to any other arguments required for your topology.

Important:

- Enabling rollback causes the event or action message to be rolled back onto the destination has been consumed from. The messaging provider then redelivers the message to the connector. The maximum number of times that messages are redelivered is dependent on the configuration of the destination, for example, a SIB destination defaults to a maximum of five attempts after which the message is sent to the associated exception destination, if it exists. Consult the documentation for your messaging provider for information about redelivery options.
- With the rollback option enabled, the JMS action connector does not return result events indicating an error to send the JMS action.

The SOAP connector

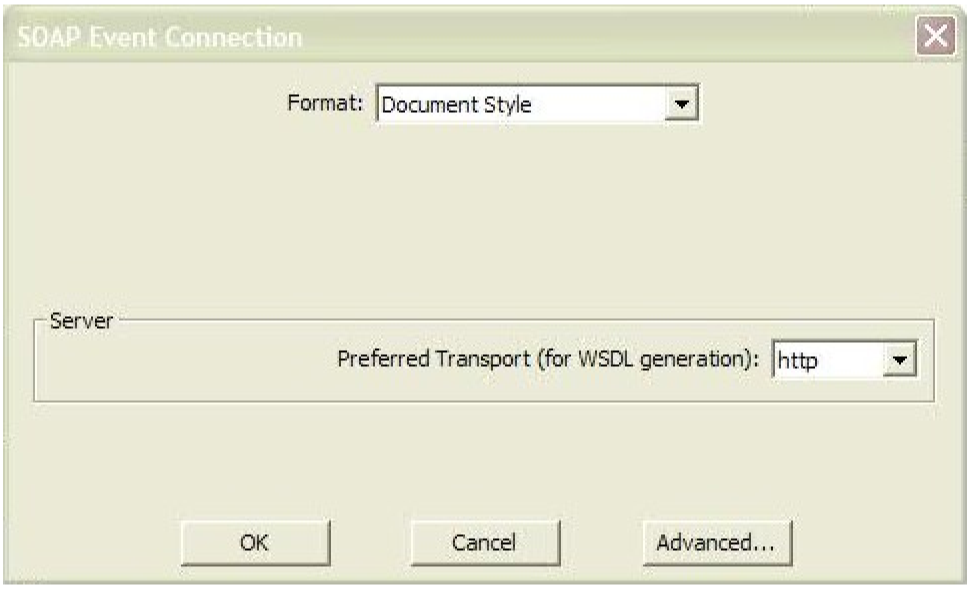
You can use the SOAP connector to receive events and send actions using SOAP over HTTP. The SOAP connector supports the Document-Literal style of SOAP binding.

Consuming events by using the SOAP connector

The SOAP connector consumes events only in a Document-Literal style. The format is configured in the connection settings for the event by using the Design Data tool.

Configuring an event to use the SOAP connector

In Design Data, right-click the event and select **Event Properties**. Click the **Connection** tab then **SOAP Connection** and then click **Configure**. The following screen capture shows the SOAP Event Connection window:



SOAP Event Connection

Format: Document Style

Server

Preferred Transport (for WSDL generation): http

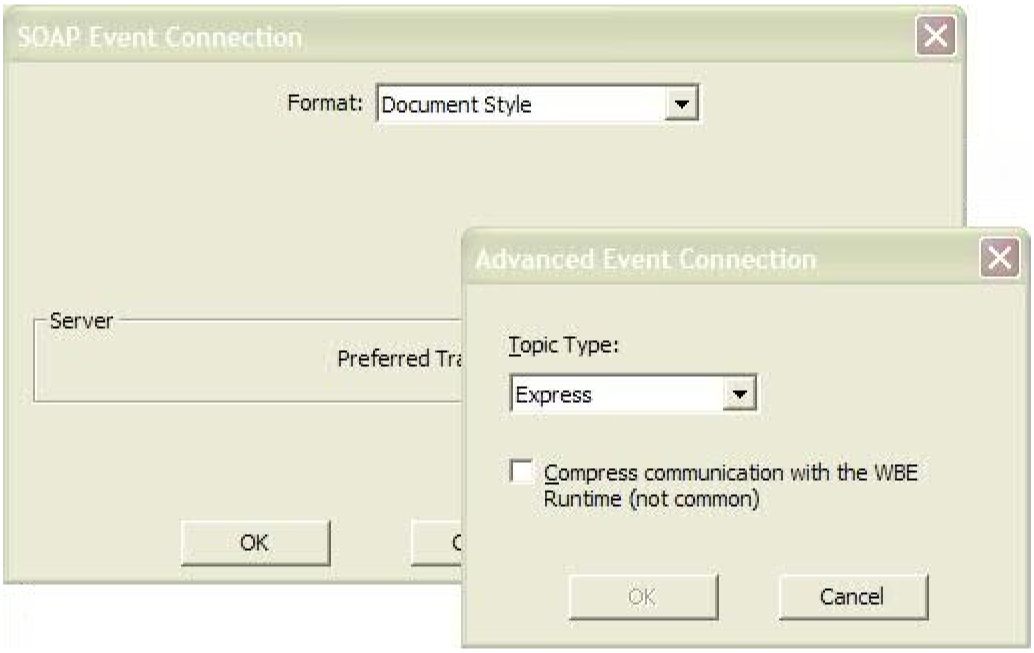
You can configure the following settings:

Format
You can only select **Document Style**.

Server
Set **Preferred Transport (for WSDL generation)** to **http** or **https**.

You can configure the following advanced settings by clicking **Advanced**:

The following screen capture shows the Advanced Event Connection window:



Topic Type

Specifies how messages are transmitted between the connector and the WebSphere Business Events Runtime.

Express

Indicates that messages are non-persistent and that the WebSphere Business Events Runtime does not make use of durable subscriptions when consuming the messages.

Reliable

Indicates that messages are persistent and that the WebSphere Business Events Runtime makes use of durable subscriptions when consuming the messages.

Compress communication with the WBE Runtime (not common)

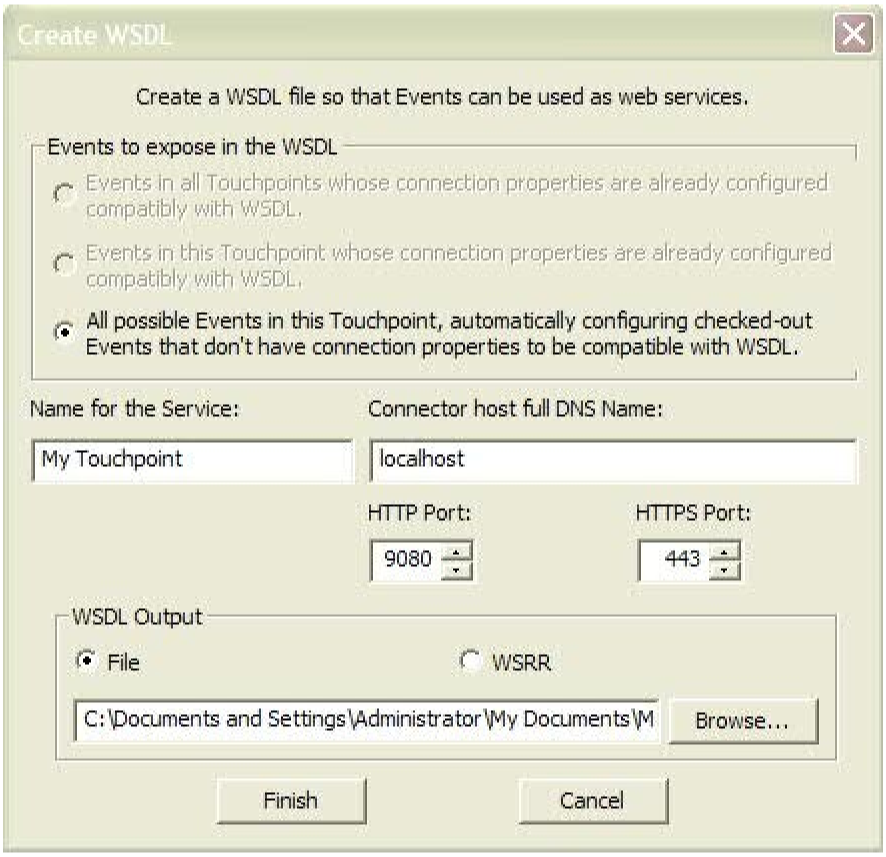
Specifies whether the data in the event messages sent by the connector to the WebSphere Business Events Runtime is compressed.

Generating the WSDL for events received by the SOAP connector

The WSDL defines the expected format of the SOAP message sent to WebSphere Business Events for a given set of events. The SOAP connector is configured to receive a SOAP message matching one that conforms to the WSDL generated from the event using Design Data.

You must use Design Data to generate a WSDL for your SOAP events, from which you can generate an appropriate client.

In Design Data, click **Tools** and **Create WSDL for Events**. The following screen capture shows the Create WSDL window:



Create WSDL

Create a WSDL file so that Events can be used as web services.

Events to expose in the WSDL

○ Events in all Touchpoints whose connection properties are already configured compatibly with WSDL.

○ Events in this Touchpoint whose connection properties are already configured compatibly with WSDL.

● All possible Events in this Touchpoint, automatically configuring checked-out Events that don't have connection properties to be compatible with WSDL.

Name for the Service: My Touchpoint

Connector host full DNS Name: localhost

HTTP Port: 9080

HTTPS Port: 443

WSDL Output

● File ○ WSRR

C:\Documents and Settings\Administrator\My Documents\M

You can configure the following settings:

Events to expose in the WSDL
There are three options to select from:

The first option includes all SOAP events in the WSDL.

The second option includes all events under the current selected touchpoint in the WSDL.

The third option includes all events under the currently selected touchpoint in the WSDL, automatically configuring these events to use HTTP and not SOAP.

Name for the Service
Defaults to the name of the touchpoint.

Connector host full DNS Name
Enter the DNS name for the location of the connectors.

HTTP Port and **HTTPS Port**
Select from a list of available ports.

WSDL Output
Typically select **File** then click **Browse** to set the file location and file name for the WSDL. Click **Finish** to generate the WSDL file.

Or, you can select **WSRR** to send the output to WebSphere Service Registry and Repository. When you click **Finish**, the WSRR Export window is displayed. See Exporting a WSDL file to WebSphere Service Registry and Repository for configuring the WSRR export.

Requirements for sending SOAP events

The new SOAP event connector uses a different endpoint address from the endpoint address in the WSDL generated from Design Data. To use the new connector, the endpoint address must be updated. You can either edit the generated WSDL file before using it to generate a web services client, or you can programmatically set the value in your web services client code. The new endpoint address required is:

`http(s)://`*`host:port`*`/wbecspsoap/SOAPEventConnector`

for example:

```
http://localhost:9080/wbecspsoap/SOAPEventConnector
```

All events sent to the SOAP event connector must adhere to the corresponding WSDL generated by WebSphere Business Events.

Producing actions by using the SOAP connector

The SOAP action connector relies on information provided by Design Data to determine the format of the SOAP request for SOAP actions.

To determine the format, the WSDL for a given web service must be imported into Design Data, which maps any operations on the web service to actions and corresponding result events, if applicable. The generated actions have a corresponding action object that maps to the input parameters of the operation. Likewise, the generated result event contains an event object mapped from the linked action object, and an action object that maps to the SOAP response from the web service. This mapping is shown in Figure 5:

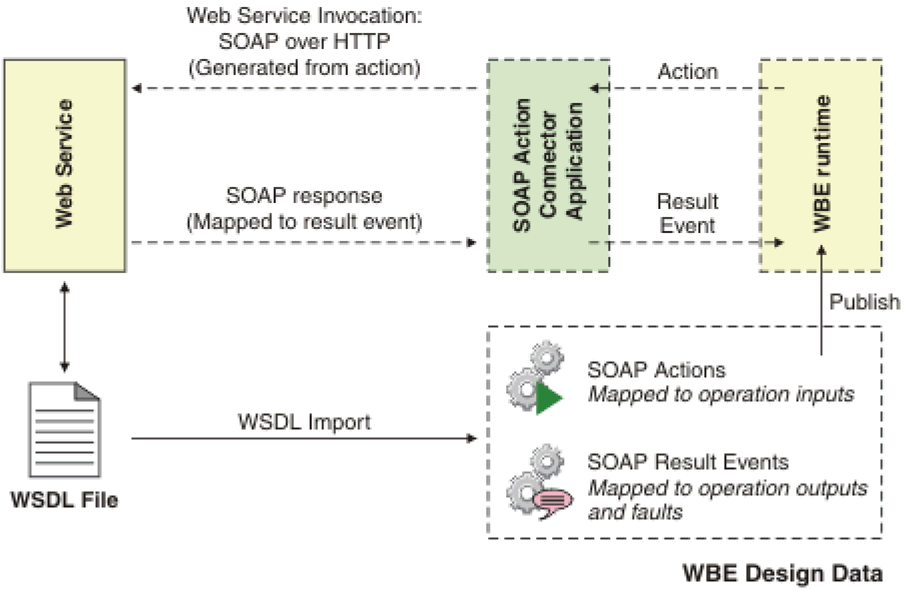


Figure 5: The SOAP action connector.

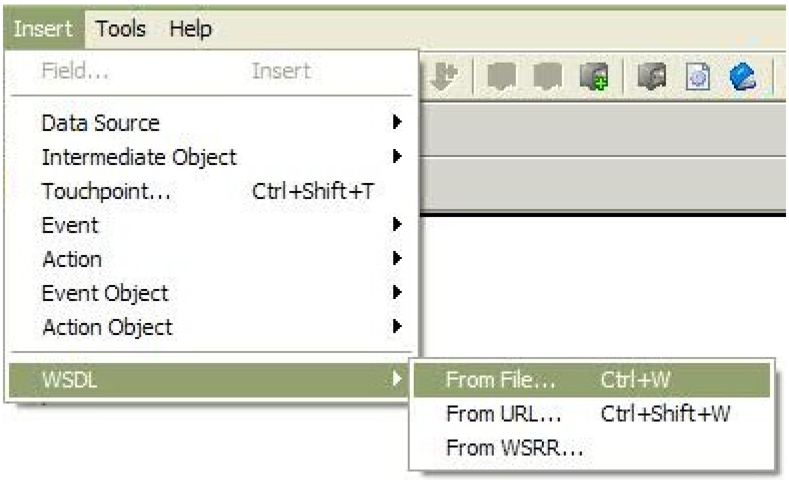
Before you begin working with the SOAP Action Connector, see Considerations for using the SOAP action connector.

Importing the WSDL using Design Data

You must have access to the WSDL in one of the following locations:

- A WSDL file
- A URL pointing to a WSDL
- WebSphere Service Registry and Repository (WSRR)

Next, in Design Data, click **Tools** and **WSDL**, then select the appropriate import option depending on the WSDL location, as shown in the following screen capture:



Design Data automatically generates the following assets, for each operation defined in the WSDL:

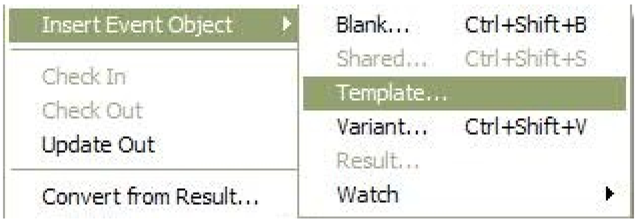
- An action corresponding to the invocation of that operation.
- One or more action objects for that action, containing fields that represent the input parameters of that operation.
- A result event with two or more event objects, if an operation has an output and is a request-response operation rather than a one-way operation. The event objects might contain:
    - One mapping to each action object for the request
    - One or more mappings to the response parameters, containing fields for each output parameter.

Using assets generated from the WSDL

Before using the assets generated from the WSDL, it is important to understand the following concepts:

- For any given operation defined in the WSDL with name ABC, a corresponding action called ABC is generated by Design Data.
- Likewise, for any given operation defined in the WSDL with name ABC that has an output defined, a corresponding result event called ABCResponse is generated by Design Data.

If you want to determine the success of the SOAP request, you must create an additional event object in the result event. Right-click the result event object, click **Insert Event Object** > **Template**, as shown in the following screen capture, then select the `_Soap_Result.xml` template:



The _Soap_Result event object is created in the result event, containing the following fields:

- **Success** (Boolean): whether the SOAP operation succeeded
- **Code** (Integer): the HTTP return code value
- **FaultCode** (String)
- **FaultString** (String)
- **FaultActor** (String)

You might find that when you import the WSDL, a field is generated under an action or event object that has no name or type specified but has a description of:

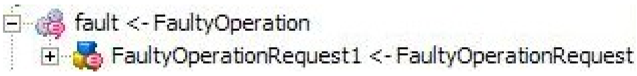
```
Unrecognized <sequence> in <complexType>. Since field definitions
cannot be determined, you may: 1. Delete this object, if its data is
not needed; 2. Change the field's data type to String, and fill this
```

```
field with raw XML for the entire object (can cause failures, if bad
XML syntax); or 3. Replace this field with individual fields, if you
provide a style sheet on the Connection Property Page to transform
the <connector-object> to this object's XML.
```

You must delete this field. The operation that requires this mapping cannot be fully supported by the SOAP action connector unless the operation can be started with no parameters sent.

Mapping result events to SOAP faults

You can define faults for any given operation in a WSDL. When you import the WSDL into Design Data, result events are created for each fault that you can use to trigger event processing logic specific to that fault. The following screen capture shows an example SOAP fault and result event:



fault <- FaultyOperation
FaultyOperationRequest1 <- FaultyOperationRequest

The result event contains an event object that is mapped from the action, like a normal result event for a SOAP action. To get more information about the SOAP fault, the _Soap_Result event object template must be added to the result event.

However, any elements defined within the SOAP fault, which are contained in the SOAP fault "detail" element, cannot be mapped to event objects. Only the following field information can be obtained for a SOAP fault, from the _Soap_Result event object template:

- `FaultCode`
- `FaultString`
- `FaultActor`

Considerations for using the SOAP action connector

The following WSDL styles are supported:

- Document-Literal

The following SOAP versions are supported:

- 1.1

The following data types are supported:

- xsd:boolean
- xsd:dateTime
- xsd:double
- xsd:long
- xsd:string

The following data types are supported, but WebSphere Business Events might invoke the web service with a value that falls outside the numeric range of allowed values for the data type:

- xsd:byte
- xsd:short
- xsd:int
- xsd:float

The following data types are unsupported:

- xsd:date
- xsd:time
- xsd:hexBinary
- xsd:anySimpleType
- Any other data type not listed as supported or supported with restrictions above

The following schema structures are supported:

- Schemas that result in SOAP body content with four or less levels of nested elements. However, not all structures that contain these levels of nesting can be successfully mapped into WebSphere Business Events.

The following schema structures are not supported:

- Schemas that result in SOAP body content with more than four levels of nested elements

The following fault handling is supported:

- A single SOAP fault per operation

The following fault handling is not supported:

- Multiple SOAP faults per operation

Cluster considerations

You must take certain considerations into account so that technology connectors in a clustered WebSphere Business Events environment behave in the required way. This section outlines the key considerations that apply to each of the connectors.

Cluster considerations applicable to all connector types

You must enable the Startup Bean service on all member servers in the cluster on which the technology connector applications are deployed. Complete the following steps for every application server in the cluster:

You must complete these steps for every application server in the cluster.

1. Launch the WebSphere Application Server administrative console.
2. Click **Servers** > **Server Types** > **WebSphere application servers** > ***server-name*** > **Container Services** > **Startup beans service**, where *server-name* is an application server in the cluster.
3. Select **Enable service at server startup**.
4. Click **Apply** and **Save**.

File System and JDBC connector cluster considerations

When running the File System or JDBC connector within a cluster, consider using peer recovery for transactional high availability. For more information, see Transactional high availability.

Review the messaging engine policies to determine the most appropriate combination of high availability and scalability for your environment. For more information, see Adding a cluster to a bus for high availability or scalability in the *WebSphere Application Server information center*.

When configuring messaging, if you choose either the scalability or scalability with high availability policy, consider setting the value of **Local queue point preference** to **Do not prefer a local queue point over other queue points** on both the event topic and durable event topic. This setting allows distribution of messages over the available queue points. For more information, see Workload sharing with queue destinations in the *WebSphere Application Server information center*.

HTTP and SOAP connector cluster considerations

When installing the HTTP and SOAP connectors to a cluster, an instance of the connector is run on each cluster member. You might want to use a proxy to distribute incoming events among the available instances. A proxy can be configured by using the WebSphere Application Server administrative console. For further information, see Setting up the proxy server in the *WebSphere Application Server information center*.

JMS connector cluster considerations

Special consideration is required when consuming events by using the JMS connector in a cluster. An instance of the JMS connector is running on each cluster member. You must use one of the following configurations to prevent each instance from receiving the same event:

- Specify consuming events from queues rather than topics.
- If consuming from topics is required, the activation specifications created for use by the JMS connector in consuming the events must specify the use of durable subscriptions. In addition:
  - For topics based on SIB, in the activation specification set the option **Share durable subscriptions** to **In cluster**.
  - For topics based on WebSphere MQ, in the activation specification select the option **Allow cloned durable subscriptions** and set the client ID field to the value `WBEJMSConnector`.

Uninstalling the SupportPac

Before you begin

The application server or cluster hosting the WebSphere Business Events Runtime must be running. In a cluster environment, you must ensure that all cluster members are running along with the deployment manager.

Note: The WebSphere Business Events Runtime is stopped and started during the uninstall process.

Procedure

To uninstall the SupportPac, follow these steps:

1. On the application server or deployment manager machine, navigate to:

   ```
   <wbe_install_dir>/csp/uninstall
   ```

2. Execute the following script:

   Windows:
   ```
   uninstallConnectors.bat <parameters>
   ```

   Linux and UNIX:
   ```
   uninstallConnectors.sh <parameters>
   ```

3. If you are using WebSphere MQ as the messaging provider, verify that the following durable subscriptions have been removed from your queue manager:

   - `JMS:<Queue_Manager_Name>:fileConnector:file_Sub`
   - `JMS:<Queue_Manager_Name>:fileConnector:file_SubDur`
   - `JMS:<Queue_Manager_Name>:httpConnector:http_Sub`
   - `JMS:<Queue_Manager_Name>:httpConnector:http_SubDur`
   - `JMS:<Queue_Manager_Name>:jdbcConnector:jdbc_Sub`
   - `JMS:<Queue_Manager_Name>:jdbcConnector:jdbc_SubDur`
   - `JMS:<Queue_Manager_Name>:jmsConnector:jms_Sub`
   - `JMS:<Queue_Manager_Name>:jmsConnector:jms_SubDur`
   - `JMS:<Queue_Manager_Name>:soapConnector:soap_Sub`
   - `JMS:<Queue_Manager_Name>:soapConnector:soap_SubDur`

   where <*Queue_Manager_Name*> is the name of your queue manager.

   If the durable subscriptions are still present, they must be manually deleted.

4. To remove the SupportPac files, delete the following directory and all subdirectories and files within it:

   ```
   <wbe_install_dir>/csp
   ```

Optional parameters for uninstallConnectors

`-host`

The hostname of the application server to connect to. For a clustered environment, you must specify the hostname of the deployment manager.

`-port`

The bootstrap port of the application server to connect to. For a clustered environment, you must specify the bootstrap port of the deployment manager.

`-serverName`

For a single server environment only. The name of the application server where the WebSphere Business Events Runtime is installed. If not specified, the name of the server defined within the profile is used.

`-clusterName`

For a clustered environment only. The name of the application server cluster where the WebSphere Business Events Runtime is installed.

`-profileName`

In a single server environment, the name of the WebSphere Application Server profile that contains the WebSphere Business Events Runtime. In a clustered environment, the name of the WebSphere Application Server profile that contains the deployment manager for the cell containing WebSphere Business Events. If this parameter is not specified, the default profile is used. In a cluster environment, the default might not match the required profile.

`-user`
`-password`

When the application server security is enabled, you must specify the administration user ID and password. There are three ways to enter the details:

1. You can enter the user ID and password each time the deploy process requires authentication. You are prompted if you do not specify the user ID and password on the command line or in a properties file.

2. You can specify the **`–user`** and **`–password`** parameters in the script. Although you are not prompted for the details during the deploy process, the user ID and password might be visible in the shell history and process listings.

3. You can specify the details in the `sas.client.props` file, which is located at:

   ```
   <path to WAS profile>/properties/sas.client.props
   ```

   To specify the credentials in this file, configure the following properties:

   ```
   com.ibm.CORBA.loginSource=properties
   com.ibm.CORBA.loginUserid=user ID
   com.ibm.CORBA.loginPassword=password
   ```

Troubleshooting

The following information might be helpful if you encounter problems:

- The deployConnectors process writes a log file to the specified output directory (outputDir parameter). The name of the log file is `deployConnectors_`*`date`*`.log`.

- Errors might also be recorded in the application server System.out and System.err logs.

- Parameter names for the scripts are case sensitive. Incorrect parameters are ignored.
-
- When deploying connectors to a cluster, and not all cluster members are running, you might see the following error:

  ```
  ******* An error occurred attempting to start the connectors
  application wbecsphttpconnector *******
  'exceptions.AttributeError WASL6044E: The Cluster:<cluster_name> MBean
  is not running.'
  ```

  where *<cluster_name>* is the name of your cluster. If you see this error, the connector application might not have started on all the cluster members. Therefore, you must manually start the application by using the WebSphere Administrative console.

- When disabling or enabling the existing connectors, or uninstalling the SupportPac, you might see the error if the WebSphere Business Events Runtime is installed on a cluster and not all cluster members are running:

  ```
  ******* An error occurred attempting to stop the WBE runtime
  application *******
  'exceptions.AttributeError WASL6044E: The Cluster:<cluster_name>  MBean
  is not running.'
  ```

  followed by

  ```
  ******* An error occurred attempting to start the WBE runtime
  application *******
  'exceptions.AttributeError WASL6043E: The Application:wberuntimeear
  MBean is running.'
  ```

  If you see this error, the WebSphere Business Events Runtime application wberuntimeear.ear must be manually stopped and started to complete the process.

- When disabling the existing connectors, if you are using WebSphere MQ as the messaging provider and the `MQ_JAVA_LIB_PATH` environment variable has not been configured to point to the directory containing the WebSphere MQ client jars, you will see the following error message:

  ```
  An error was encountered when removing durable subscriptions:
  com.ibm.websphere.naming.CannotInstantiateObjectException: Exception
  ```

```
occurred while the JNDI NamingManager was processing a
javax.naming.Reference object. [Root exception is
java.lang.NoClassDefFoundError: com.ibm.mq.MQEnvironment]
```

To resolve this problem, you must ensure that the WebSphere MQ client jars are available on the machine and configure the `MQ_JAVA_LIB_PATH` environment variable correctly.

- When sending HTTP events to the HTTP connector on a server or cluster with application security enabled, incorrect or missing credentials in the request might result in the following HTTP responses:

  - An HTTP 401 response indicates that authentication has failed. Either the wrong credentials have been specified or credentials are missing in the request.
  - HTTP 403 errors indicate that authorization has been denied. The user is known to the application server but does not have the **HTTPEventConnectorUser** role.

- When processing large files, threads used by the File System connector might be active for a long time, resulting in the following message appearing in the WebSphere Application Server logs:

  ```
  WSVR0605W: Thread threadname has been active for hangtime and may be
  hung.
  There are totalthreads threads in total in the server that may be hung.
  ```

  You can ignore this warning. If you do not want the warning, adjust the hang detection policy in the server. For more information, see the Configuring the hang detection policy.

- *The information in the following paragraph applies to SupportPac Version 2.0.0.1 and higher.*

  Message driven beans are used when the connector applications receive actions from the runtime, and are used by the JMS connector for consuming events. The default transaction timeout for these beans is set to 240 seconds. If you set the `wbe.csp.jms.connection.max.retry.time` property to a larger value, or if timeout values related to HTTP or JMS connections are larger, you might need to increase the transaction timeout for the beans. To change the transaction timeout, add the following WebSphere Business Events property:

  ```
  wbe.csp.mdb.transaction.timeout
  ```

  The value is the required timeout in seconds.

  After changing this property, you must restart the WebSphere Business Events runtime application. If the connector applications have been deployed, you must update the connector applications by running the **`deployConnectors`** command specifying the **`-update`** argument in addition to any other arguments required for your topology.

Notices

This information was developed for products and services offered in the U.S.A.
IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.
IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Trademarks