



Common Event Infrastructure

Note

Before using this information, be sure to read the general information in "Notices" on page 41.

30 March 2007

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

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

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

© Copyright International Business Machines Corporation 2006, 2007. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Chapter 1. Removing the Common Event Infrastructure configuration	1
Removing the Common Event Infrastructure application.	1
Removing the event messaging enterprise application	2
Removing the event database.	2
Chapter 2. Administering the Common Event Infrastructure	5
Post-installation configuration	5
Configuring the event database	5
Deploying the Common Event Infrastructure application	21
Configuring event messaging	23
Runtime configuration.	25
Default configuration	25
Configuring the Events service	27
DB2 database maintenance	28
Purging events from the event database	30
Chapter 3. Viewing events with the Common Base Event browser	33
Specifying the events to view	33
Working with events returned from the event browser	34
Chapter 4. Security and the Common Event Infrastructure	35
Chapter 5. Troubleshooting the Common Base Event browser.	39
Notices	41
Programming interface information	43
Trademarks and service marks	43

Chapter 1. Removing the Common Event Infrastructure configuration

If you need to remove the configuration for the Common Event Infrastructure, in preparation for uninstalling WebSphere Process Server, you must first remove the deployed enterprise applications and the database configuration.

The Common Event Infrastructure is installed in the base installation of WebSphere Process Server, but will not be active unless it is already configured. This topic tells you only how to remove a previously configured instance of the Common Event Infrastructure. To remove the configuration for the Common Event Infrastructure, follow these steps:

Removing the Common Event Infrastructure application

If you need to remove the event server enterprise application and resources from WebSphere Process Server, you can use the `event-application.jacl` script.

If you prefer, you can remove the event server enterprise applications manually using the administrative console rather than using the `event-application.jacl` script. If use the administrative console, you must also manually remove the Common Event Infrastructure resources. For more information about these resources, see *Default configuration* in the section on Configuring the Common Event Infrastructure.

To remove the event server enterprise application, use the `wsadmin` tool to run the `event-application.jacl` script.

To run the script on a Windows system, go to the `profile_root\event\application` directory and run the following command (all on one line):

```
wsadmin -f event-application.jacl -profile event-profile.jacl -action uninstall  
-node node_name -server server_name
```

To run the script on a Linux or UNIX system, go to the `profile_root/event/application` directory and run the following command (all on one line):

```
wsadmin.sh -f event-application.jacl -profile event-profile.jacl  
-wsadmin_classpath install_root/event/lib/cei_installer.jar -action  
uninstall -node node_name -server server_name  
[-appname app_name] [-trace]
```

The `event-application.jacl` script uses these parameters:

node_name

The WebSphere Process Server node from which you want to remove the event server enterprise application.

server_name

The WebSphere Process Server from which you want to remove the event server enterprise application. This parameter is optional. If you do not specify a server, the enterprise application is removed from all servers in the node.

app_name

The name of the deployed event server enterprise application you want to

remove. This parameter is optional. If you do not specify an application name, all registered Common Event Infrastructure enterprise applications are removed.

The optional **-trace** parameter causes additional debugging information to display on the standard output.

Removing the event messaging enterprise application

Before uninstalling the Common Event Infrastructure, you must remove the event messaging enterprise application.

To remove the event messaging enterprise application, use the wsadmin tool to run the `event-message.jacl` script.

The parameters of the `event-message.jacl` script are as follows:

node_name

The WebSphere Process Server node from which you want to remove the event messaging enterprise application.

To discover the node name, run the `profile_root/bin/setupCmdLine` and then the `echo $WAS_NODE` command.

server_name

The WebSphere Process Server from which you want to remove the event messaging enterprise application. This parameter is optional. If you do not specify a server, the application is removed from all servers in the specified node.

app_name

The name of the deployed messaging enterprise application you want to remove. This parameter is required; if you installed the Common Event Infrastructure with the default values, then the application name is `EventServerMdb`.

The optional **-trace** parameter causes additional debugging information to display on the standard output.

Removing the event database

If you need to remove the event database, you can use the provided scripts. You must remove the database before you uninstall the Common Event Infrastructure.

When the database is configured, the configuration script also creates scripts for removing the database and the Java database connectivity (JDBC) provider. The scripts for removing the event database are placed in database-specific subdirectories of the `profile_root/event/dbscripts` directory. The scripts for removing the JDBC provider are placed in database-specific subdirectories of the `profile_root/event/dsscripts` directory.

Note: The event database can be shared among multiple event servers using the same JDBC provider configuration. Therefore, remove the JDBC provider configuration only if you have uninstalled the associated event database.

To remove the event database and JDBC provider, run the appropriate scripts from the following table.

Type	Operating system	Database script	JDBC configuration script
Cloudscape	Windows	rm_event_cloudscape.bat	rm_cloudscape_jdbc_provider.bat
Cloudscape	Linux/UNIX	rm_event_cloudscape.sh	rm_cloudscape_jdbc_provider.sh
DB2	Windows	rm_event_db2.bat	rm_db2_jdbc_provider.bat
DB2	Linux/UNIX	rm_event_db2.sh	rm_db2_jdbc_provider.sh
Oracle	Windows	rm_event_oracle.bat	rm_oracle_jdbc_provider.bat
Oracle	Linux/UNIX	rm_event_oracle.sh	rm_oracle_jdbc_provider.sh
Cloudscape	z/OS (Windows script)	rm_event_cloudscape.bat	rm_cloudscape_jdbc_provider.bat
Cloudscape	z/OS (Linux/UNIX script)	rm_event_cloudscape.sh	rm_cloudscape_jdbc_provider.sh
DB2	z/OS (Windows script)	rm_event_db2zos.bat	rm_db2zos_jdbc_provider.bat
DB2	z/OS (Linux/UNIX script)	rm_event_db2zos.sh	rm_db2zos_jdbc_provider.sh

You can remove the event database or JDBC provider at any time by running the appropriate script. To remove the JDBC provider, use the appropriate script and specify the scope in which you want to remove the JDBC provider:

```
rm_db_jdbc_provider scope [server_name]
```

The generated scripts use these parameters:

scope The scope in which you want to remove the JDBC provider. The valid values are cell, node, and server.

server_name

The name of the WebSphere Process Server from which you want to remove the JDBC provider, if **scope** is server. (If **scope** is cell or node, this parameter is ignored.)

Chapter 2. Administering the Common Event Infrastructure

The Common Event Infrastructure is a core component of WebSphere Process Server.

The Common Event Infrastructure provides facilities for the runtime environment to persistently store and retrieve events from many different programming environments. Events are represented using the Common Base Event model a standard, XML-based format that defines the structure of an event (see the [../com.ibm.wsps.602.mon.doc/doc/cmon_event_catalog.dita](#) topic for further details)

The Common Event Infrastructure provides WebSphere Process Server with standard formats and mechanisms for managing event data. The following facilities are provided:

- Standard interfaces and services for WebSphere applications to create event objects, store them, send them, and retrieve them later.
- Facilities that pass event objects to registered applications either directly, in the context of the producing (source) application, or indirectly through Java Message Service (JMS).

There are event emitters for Business Process Execution Language (BPEL) -based processes and for Enterprise JavaBeans (EJB) invocations based on deployment descriptor extensions.

- The Common Base Event browser for browsing stored events.

Post-installation configuration

Before you can begin using the Common Event Infrastructure, you must complete several post-installation configuration tasks.

Configuring the event database

Database configuration includes creating the required tables and configuring JDBC data sources.

The event database is required to support persistence of events. You must configure a new event database under any of the following circumstances:

- You are setting up a new installation of the Common Event Infrastructure.
- You have migrated from a previous version of Common Event Infrastructure which was configured to use a Cloudscape event database. This is because migration is not supported for a Cloudscape event database.

Configuring the event database (Linux, UNIX, and Windows systems)

You can configure an event database on a Linux, UNIX, or Windows system using Cloudscape, DB2, or Oracle database software.

To configure the event database:

1. Go to the *profile_path/event/dbconfig* directory for the profile defining the WebSphere Process Server runtime environment where you want to configure the database. (Replace *profile_path* with the path to the directory containing the WebSphere Process Server profile.)

- Using an ASCII text editor, open one of the provided sample database response files. Select the sample response file for the database software you are using:

Database	Sample response file
Cloudscape	CloudscapeResponseFile.txt
DB2 Universal Database for Linux, UNIX, and Windows	DB2ResponseFile.txt
Oracle Database	OracleResponseFile.txt

- Modify the database response file with the correct information for your environment. (See the comments in the sample response file for more information about the parameters, including complete syntax information.)
If you are using a DB2 database on a Linux, UNIX, or Windows system, specify the following parameter values:

JDBC_CLASSPATH

The path to the DB2 JDBC driver.

UNIVERSAL_JDBC_CLASSPATH

The path to the Universal JDBC driver.

If you are using a DB2 database on a z/OS system, specify the following parameter values:

DB_NAME

The name of the z/OS database you created for the event database.

JDBC_CLASSPATH

The path to the DB2 JDBC driver.

UNIVERSAL_JDBC_DRIVER_NATIVEPATH

The path to the Universal JDBC native library path.

UNIVERSAL_JDBC_CLASSPATH

The path to the Universal JDBC driver.

If you are using an Oracle database, specify the following parameter values:

ORACLE_HOME

The path to the Oracle home directory.

JDBC_CLASSPATH

The JDBC driver class path.

- Run the database configuration script for your operating system, specifying the name of the database response file as a parameter. Use one of the following commands:

Windows systems

`config_event_database.bat response_file`

Linux and UNIX systems

`config_event_database.sh response_file`

For example, the following command would configure a DB2 event database on a Windows system: `config_event_database.bat DB2ResponseFile.txt`

The script configures the event database and creates two JDBC data sources: one for the event database and one for the event catalog. A message is displayed when database configuration is complete.

Note: If your database response file specifies `EXECUTE_SCRIPTS=NO`, you must complete the database configuration by manually running the generated

scripts. The default value in the sample database response file is EXECUTE_SCRIPTS=YES. If you are using a remote database you need to change the value of EXECUTE_SCRIPTS to NO, copy the folder containing the scripts to the remote database server, and run the scripts manually from that server.

After you configure the event database, you must restart the application server.

Manually running database configuration scripts:

If your database response file specifies EXECUTE_SCRIPTS=NO, you must complete the database configuration process by manually running the generated scripts.

Database configuration is a two-step process. The **config_event_database** script first generates database-specific scripts for your environment; these generated scripts then create the database and data sources. If your database response file specifies EXECUTE_SCRIPTS=YES, the **config_event_database** performs both steps automatically. However, if your database response file specifies EXECUTE_SCRIPTS=NO, you must complete the database configuration by manually running the generated scripts.

The default value in the sample database response file specifies EXECUTE_SCRIPTS=YES. Unless you have changed this value in your customized response file, you do not need to run the generated scripts manually.

To manually run the generated scripts:

1. Go to the database-specific subdirectory containing the generated scripts for creating the event database.

The scripts are placed in one of the following directories, depending upon the database software you are using:

- *profile_path*/event/dbscripts/cloudscape
- *profile_path*/event/dbscripts/db2
- *profile_path*/event/dbscripts/oracle

Replace *profile_path* with the path to the directory containing the profile for the WebSphere Process Server runtime environment in which you are configuring the event database.

2. Run the generated database creation script for your operating system and database software:

Type	Operating system	Database script
Cloudscape	Windows	cr_event_cloudscape.bat
Cloudscape	Linux/UNIX	cr_event_cloudscape.sh
DB2	Windows	cr_event_db2.bat
DB2	Linux/UNIX	cr_event_db2.sh
Oracle	Windows	cr_event_oracle.bat
Oracle	Linux/UNIX	cr_event_oracle.sh

3. Go to the database-specific subdirectory containing the generated scripts for creating the JDBC data sources.

The scripts are placed in one of the following directories, depending upon the database software you are using:

- *profile_path*/event/dsscripts/db2
- *profile_path*/event/dsscripts/db2zos
- *profile_path*/event/dsscripts/cloudscape
- *profile_path*/event/dsscripts/oracle

Replace *profile_path* with the path to the directory containing the profile for the WebSphere Process Server runtime environment in which you are configuring the event database. If you are using a database that is installed on another server, copy the entire folder (for example, the *profile_path*/event/dsscripts/db2 folder) of generated database scripts over to that server. On the database server, open a command prompt to the location of the copied scripts, and run the script to create the database:

Windows systems

```
cr_event_db.bat server username
```

Linux and UNIX systems

```
cr_event_db.sh server username
```

The parameters are as follows:

server Indicates that you are executing this command on the same computer as the database

username

A valid user id for accessing the database.

4. On the server where you originally generated the scripts, run the generated JDBC configuration script for your operating system and database software:

Type	Operating system	JDBC configuration script
Cloudscape	Windows	cr_cloudscape_jdbc_provider.bat
Cloudscape	Linux/UNIX	cr_cloudscape_jdbc_provider.sh
DB2	Windows	cr_db2_jdbc_provider.sh
DB2	Linux/UNIX	cr_db2_jdbc_provider.sh
Oracle	Windows	cr_oracle_jdbc_provider.bat
Oracle	Linux/UNIX	cr_oracle_jdbc_provider.sh

Use the following command to run the JDBC configuration script:

Windows systems

```
cr_db_jdbc_provider scope [server_name]
```

Linux and UNIX systems

```
cr_db_jdbc_provider.sh scope [server_name]
```

The parameters are as follows:

scope The scope at which you want to configure the JDBC provider. The valid values are cell, node, server, or cluster.

server_name

The name of the WebSphere server where you want to configure the JDBC provider, if **scope** is server. (If **scope** is cell, this parameter is ignored.)

Note: You will be prompted to provide a user id and password that is valid on the database server.

After the event database is configured, you must restart the application server.

Configuring a Cloudscape database:

There are two steps required for configuring a Cloudscape event database.

You must configure the Cloudscape event database before using the Common Event Infrastructure.

1. Create a database response file. A database response file is a text file that specifies parameters for configuring the event database. These parameters vary depending on the type of database being used.
2. Run the database configuration scripts. The Common Event Infrastructure provides scripts for configuring or upgrading the event database. These scripts in turn generate customized, database-specific scripts for creating or modifying the necessary database configuration using the parameters in your response file.

Configuring a DB2 database on a Linux, UNIX, or Windows system:

Follow these steps to configure a DB2 event database on a Linux, UNIX, or Windows system.

Although you can configure a DB2 event database using the Profile Wizard for the WebSphere Process Server profile creation, you need to manually do steps 1 and 2, below, *before* configuring the DB2 database.

1. If you are configuring a DB2 database on a DB2 client with the server on a remote system, make sure the client system is configured to communicate with the server and that the DB2 node is cataloged. For more information, refer to the DB2 Universal Database documentation.
2. If you are configuring a DB2 database on a Linux or UNIX system, source the database environment:
 - a. Modify `/etc/group` and make sure `root` is in the same group as the `db2instance`.
 - b. Source the database environment by running the `db2instance/sql1lib/db2profile` script (replace `db2instance` with the name of your database instance).
3. Create a database response file. A database response file is a text file that specifies parameters for configuring the event database. These parameters vary depending on the type of database being used.
4. Run the database configuration scripts. If you are configuring a DB2 database on a DB2 client with the server on a remote system, copy the folder containing the generated database configuration scripts to the remote database server. The Common Event Infrastructure provides scripts for configuring the event database. These scripts generate customized, database-specific scripts for creating the database configuration using the parameters in your response file.

Configuring an Oracle database:

Follow these steps to configure an Oracle event database.

You must configure the Oracle event database before using the Common Event Infrastructure.

1. If you are using Oracle8i Database Release 8.1.7 and a type 2 JDBC driver, use the WebSphere administrative console to modify the Common Event Infrastructure data source `Event_Oracle_JDBC_Provider` to use the class path `$ORACLE_HOME/jdbc/lib/classes12.zip`.

2. If you are using Oracle8i Database Release 8.1.7 and a type 4 JDBC driver, perform one of the following commands:
 - Use the WebSphere administrative console to modify the Common Event Infrastructure data source Event_Oracle_JDBC_Provider to use class path `$ORACLE_HOME/jdbc/lib/classes12.zip`
 - Download the `ojdbc14.jar` file from <http://www.oracle.com> and place the file in the `$ORACLE_HOME/jdbc/lib` directory. (You do not need to modify the data source.)
3. Create a database to use for the event database. When you create the database, select the appropriate character set:
 - For an Oracle 9 database, select Unicode (AL32UTF8).
 - For an Oracle 8 database, select UTF-8.

If you are configuring the event database from an Oracle client, the client must be configured to communicate with this database instance. In the next step, specify the Oracle system identifier (SID) in the database response file; the default value is `event`.
4. Create a database response file. A database response file is a text file that specifies parameters for configuring the event database. These parameters vary depending on the type of database being used.

If you are installing multiple event servers using the same Oracle database, remember that table space names must be unique within a database instance. Therefore, you must modify the response file to specify different table space names for each event server you install.
5. Run the database configuration scripts. The Common Event Infrastructure provides scripts for configuring the event database. These scripts generate customized, database-specific scripts for creating the database configuration using the parameters specified in your response file.
6. Make sure the following environment variables are set:
 - `ORACLE_HOME` must be set to the Oracle installation directory.
 - `PATH` must include `$ORACLE_HOME/bin` and `$ORACLE_HOME/lib`.
 - `LD_LIBRARY_PATH` and `LIBPATH` must be set to `$ORACLE_HOME/lib`.
 - `CLASSPATH` must contain `$ORACLE_HOME/jdbc/lib/ojdbc14.jar` and `$ORACLE_HOME/jdbc/lib/nls_charset12.zip`.

Oracle database limitations:

Some limitations apply to configurations using Oracle database software. If you configured an Oracle event database, keep the following considerations in mind:

- The Oracle 10g JDBC thin driver imposes some size restrictions for string values if you are using a Unicode character set. This can result in an Oracle ORA-01461 error when events containing large values (such as a long message attribute) are stored in the event database. For more information about this restriction, refer to the Oracle 10g documentation.

To avoid this problem, use the Oracle 10g OCI driver or the Oracle 9i thin driver.
- Oracle database software treats a blank string as a NULL value. If you specify a blank string as an event attribute value, that string is converted to a NULL when it is stored in an Oracle event database.

Creating a database response file:

A database response file is a text file specifying parameters for configuring the event database. These parameters vary depending on the type of database being used.

If you are upgrading an existing Cloudscape event database, you must use the same response file you used when you originally configured the database. A backup copy of this response file is created during the Common Event Infrastructure installation and saved as *profile_root/event/dbconfig/CloudscapeResponseFile.bak*.

To create a database response file, follow these steps:

1. Using an ASCII text editor, open one of the sample database response files. These files are located in the *profile_root/event/dbconfig* directory. Select the sample response file for the database software you are using:

Database	Sample response file
Cloudscape	CloudscapeResponseFile.txt
DB2 Universal Database for z/OS	DB2ZOSResponseFile.txt

2. Modify the parameters in the response file as appropriate for your database configuration.
3. Save the file to your Common Event Infrastructure installation directory. You can give the modified response file any name you want to use; you specify this file when you run the database configuration script.

Cloudscape database response file:

A Cloudscape database response file specifies parameters for configuring a Cloudscape event database.

A sample Cloudscape database response file called *CloudscapeResponseFile.txt* is available in the *profile_root/event/dbconfig* directory.

This response file specifies the following parameters:

SERVER_NAME=*server*

The name of the WebSphere Process Server where the database is installed.

This parameter is applicable only if the **SCOPE** parameter is set to **server**. The parameter will be ignored when scope is **cell** or **node**, and is not valid if the scope is **cluster**. If you do not specify a server name, the default value is **server1**.

CLUSTER_NAME=

The name of the cluster containing the WebSphere Process Server where the database is installed. This parameter is applicable only if the **SCOPE** parameter is set to **cluster**. The parameter will be ignored when scope is **cell** or **node**, and is not valid if the scope is **server**.

SCOPE=[server | node | cell | cluster]

The scope in which the configured database is shared. This is the scope in which Java database connectivity (JDBC) data sources are created. This parameter is optional. The default value is **server**.

DB_NAME=*name*

The name of the event database. This parameter is optional. The default value is **event**.

DB_SYSTEM_DIR=

The directory where the Common Event Infrastructure database will be created. If not specified then the default will be *profile_root/event/CloudScapeEventDB*

JDBC_PROVIDER=*provider*

The name of the JDBC provider to configure. The value must be the name of a JDBC driver supported by WebSphere Process Server Version 5.1, and later. The Cloudscape JDBC Provider (XA) driver is recommended.

DB_TYPE=CLOUDSCAPE

The type of database to be configured. For a Cloudscape database, this must be **CLOUDSCAPE**.

PAGE_CACHE_SIZE=*size*

The number of memory pages to use for caching data. Increasing the page cache size can improve performance, but also requires more memory. See the Cloudscape documentation for more information about caching. This parameter is optional. The default value is 4000.

LOG_DEVICE=*path*

The path to the location where the transaction logs are written. Using a separate device for logs can improve performance, but it also complicates backup and recovery. This parameter is optional.

DB2 database response file for Linux, UNIX, and Windows systems:

A DB2 database response file for Linux, UNIX, and Windows systems specifies parameters for configuring a DB2 event database that runs on any of these operating systems.

A sample DB2 database response file for Linux, UNIX, and Windows, called *DB2ResponseFile.txt*, is available in the *profile_root/event/dbconfig* directory. This response file for Linux, UNIX, and Windows systems specifies the following parameters:

SERVER_NAME=*server*

The name of the WebSphere Process Server where the database is installed. This parameter is applicable only if the **SHARE_DB** parameter is set to **server**. If you do not specify a server name, the default value is *server1*.

CLUSTER_NAME=

The name of the cluster containing the WebSphere Process Server where the database is installed. This parameter is applicable only if the **SCOPE** parameter is set to **cluster**. The parameter will be ignored when scope is **cell** or **node**, and is not valid if the scope is **server**.

SCOPE=[server | node | cell | cluster]

The scope in which the configured database is shared. This is the scope in which Java database connectivity (JDBC) data sources are created. This parameter is optional. The default value is **server**.

DB_NAME=*name*

The name of the event database. This name must be no longer than 8 characters. This parameter is optional; the default value is *event*.

DB_NODE_NAME=*name*

The database server node name. This parameter is required only if the database server is remote; it is not required if the database server is on the local system.

JDBC_PROVIDER=*provider*

The name of the JDBC provider to configure. The value must be the name of a JDBC driver supported by WebSphere Process Server Version 5.1, and later. The following drivers are recommended:

- DB2 Universal JDBC Driver Provider (XA)
- DB2 Legacy CLI-based Type 2 JDBC Provider (XA)

JDBC_CLASSPATH=*path*

This should be one of the following, containing paths to the JDBC driver (not including file name):

- For DB2 Universal JDBC Driver Provider (XA): the path to the db2jcc_license_cu.jar and db2jcc_license_cisuz.jar files
- For DB2 Legacy CLI-based Type 2 JDBC Driver (XA), the path to the db2java.zip file

UNIVERSAL_JDBC_CLASSPATH=*path*

For DB2 Universal JDBC Driver Provider or DB2 Universal JDBC Driver Provider (XA), the path to the JDBC driver (not including file name). This should be the path to the db2jcc_license_cu.jar file. This parameter is optional.

JDBC_DRIVER_TYPE=*type*

The JDBC driver type. This should be either 2 or 4.

DB_HOST_NAME=*hostname*

The database server host name. This parameter is required if JDBC_DRIVER_TYPE is set to 4. The default value is localhost.

DB_INSTANCE_PORT=*port*

The database instance port number. This parameter is required if JDBC_DRIVER_TYPE is set to 4. The default port number is 50000.

EXECUTE_SCRIPTS=[YES|NO]

Specifies whether the database configuration scripts are automatically run. If this parameter is set to **NO**, the scripts for configuring the database are generated but do not run.

DB_TYPE=DB2

The type of database to configure. For a DB2 database, this must be **DB2**.

PAGE_SIZE_4K_BUFFER_POOL=*size*

The database 4K buffer pool size, in KB. The default value is 1500.

PAGE_SIZE_8K_BUFFER_POOL=*size*

The database 8K buffer pool size, in KB. The default value is 1500.

PAGE_SIZE_16K_BUFFER_POOL=*size*

The database 16K buffer pool page size, in KB. The default value is 1000.

LOG_FILE_SIZE=*size*

The size of the primary and secondary log files, in 4 KB pages. This parameter is optional. The default value is 1000.

NUM_PRIMARY_LOG=*n*

The number of primary log files to use. This parameter is optional. The default value is 3.

NUM_SECONDARY_LOG=*n*

The number of primary log files to use. This parameter is optional. The default value is 2.

TRANSACTION_LOG_FILE=*path*

The path to the location where the transaction log files are written. If you place the database transaction logs on a different disk from the event database tables and indexes, this can significantly improve performance. This parameter is optional. If no value is specified, the default location is used.

Oracle database response file:

An Oracle database response file specifies parameters for configuring an Oracle event database.

Purpose

These parameters can also be specified in the installation response file if **CONFIGURE_DB** is set to **true** and **DB_TYPE** is set to **ORACLE**. A sample Oracle database response file called OracleResponseFile.txt is available in the *profile_home/event/dbconfig/oracle* directory of the Common Event Infrastructure development system.

Parameters**SERVER_NAME=***server*

The name of the WebSphere Process Server where the database is installed. This parameter is applicable only if the **SHARE_DB** parameter is set to **server**. If you do not specify a server name, the default value is **server1**.

CLUSTER_NAME=

The name of the cluster containing the WebSphere Process Server where the database is installed. This parameter is applicable only if the **SCOPE** parameter is set to **cluster**. The parameter will be ignored when scope is **cell** or **node**, and is invalid if the scope is **server**.

SCOPE=[server | node | cell | cluster]

The scope in which the configured database is shared. This is the scope in which Java database connectivity (JDBC) data sources are created. This parameter is optional. The default value is **server**.

ORACLE_HOME=*path*

The path to the Oracle home directory.

DB_NAME=*name*

The name of the event database. This name is used as the Oracle System ID (SID). This parameter is optional; the default value is **event**.

JDBC_PROVIDER=*provider*

The name of the Java Database Connectivity (JDBC) provider to configure. The value must be the name of a JDBC driver supported by WebSphere Process Server Version 5.1. The Oracle JDBC Provider (XA) driver is recommended:

JDBC_CLASSPATH=*path*

The path to the JDBC driver (not including file name).

JDBC_DRIVER_TYPE=[thin | oci8]

The JDBC driver type.

DB_HOST_NAME=*hostname*

The host name of the database server.

DB_INSTANCE_PORT=*port*

The database instance port number.

EXECUTE_SCRIPTS=[YES | NO]

Specifies whether the database configuration scripts should be automatically run. If this parameter is set to **NO**, the scripts for configuring the database will be generated but will not be run.

DB_TYPE=ORACLE

The type of database being configured. For an Oracle database, this must be **ORACLE**.

SCHEMA_NAME=CEI

The database schema name to use. This should always be **CEI**.

PROFILE=*profile*

Specifies the profile for the schema user in the Oracle database. The default value is `cei_profile`.

ROLE=*role*

Specifies the application role for the schema user in the Oracle database. The default value is `cei_role`.

TABLE_SPACE_BASE_NAME=*name*

Specifies the default table space name. This name must be no longer than 18 characters. The default value is `cei_ts_base`.

TABLE_SPACE_BASE_PATH=*path*

Specifies the default table space path. The default value is `cei_ts_base`.

TABLE_SPACE_BASE_INITIAL_SIZE_MB=*size*

Specifies the initial size (in MB) of the default table space. The default size is 102 MB.

TABLE_SPACE_NEXT_EXTENT_SIZE_MB=*size*

Specifies the automatic increment size (in MB) of the default table space. The default size is 10 MB.

TABLE_SPACE_TEMP_NAME=*temp*

Specifies the temporary table space name. This name must be no longer than 18 characters. The default value is `cei_ts_temp`.

TABLE_SPACE_TEMP_PATH=*path*

Specifies the temporary table space path. The default value is `cei_ts_temp`.

TABLE_SPACE_TEMP_SIZE_MB=*size*

Specifies the initial size (in MB) of the temporary table space. The default size is 7 MB.

TABLE_SPACE_EXTENDED_NAME=*name*

Specifies the extended table space name. This name must be no longer than 18 characters. The default value is `cei_ts_extended`.

TABLE_SPACE_EXTENDED_PATH=*path*

Specifies the extended table space path. The default value is `cei_ts_extended`.

TABLE_SPACE_EXTENDED_INITIAL_SIZE_MB=*size*

Specifies the initial size (in MB) of the extended table space. The default size is 204 MB.

TABLE_SPACE_EXTENDED_NEXT_EXTENT_SIZE_MB=*size*

Specifies the automatic increment size (in MB) of the extended table space. The default size is 20 MB.

TABLE_SPACE_CATALOG_NAME=*name*

Specifies the table space name for the event catalog. This name must be no longer than 18 characters. The default value is `cei_ts_catalog`.

TABLE_SPACE_CATALOG_PATH

Specifies the table space path for the event catalog. The default value is `cei_ts_catalog_path`.

TABLE_SPACE_CATALOG_INITIAL_SIZE_MB

Specifies the initial size (in MB) of the event catalog table space. The default size is 10 MB.

TABLE_SPACE_CATALOG_NEXT_EXTENT_SIZE_MB=*size*

Specifies the automatic increment size (in MB) of the event catalog table space. The default size is 5 MB.

Configuring a DB2 database on a z/OS system

You can configure an event database on a z/OS system using DB2 database software.

To configure the DB2 database from a remote client, you must have the DB2 Connect product installed with the latest fix packs.

Note: Follow this procedure only if you are configuring a DB2 event database on a z/OS system.

To configure the event database:

1. On the z/OS system, use the DB2 administration menu to create a new subsystem.
2. Create a storage group. You will also need to specify the storage group name in the database response file; the default value is `sysdef1t`.
3. Grant the necessary permissions to the user ID you want the WebSphere Process Server data source to use. This user ID must have rights to access the database and storage group you created; it must also have permission to create new tables, table spaces, and indexes for the database.
4. Catalog the remote database. Use the following commands, either in a script or in a DB2 command-line window:

```
catalog tcpip node zosnode remote hostname server IP_port system db_subsystem
catalog database db_name as db_name at node zosnode authentication DCS
```

For more information about how to catalog a nodes and databases, refer to the DB2 Connect documentation.

5. Verify that you can establish a connection to the remote subsystem. You can check this by running the following command:

```
db2 connect to subsystem user userid using password
```
6. Bind to the host database. Use the following commands:

```
db2 connect to db_name user userid using password
db2 bind path/bnd/@ddcsmvs.lst blocking all sqlerror continue message
    mvs.msg grant public
db2 connect reset
```

For more information about binding a client to a host database, refer to the DB2 Connect documentation.
7. Go to the `profile_path/event/dbconfig` directory for the profile defining the WebSphere Process Server runtime environment where you want to configure the database. (Replace `profile_path` with the path to the directory containing the WebSphere Process Server profile.)
8. Using an ASCII text editor, open the `DB2ZOSResponseFile.txt` sample database response file.

9. Modify the database response file with the correct information for your environment. (See the comments in the sample response file for more information about the parameters, including complete syntax information.) Specify the following parameter values:

DB_NAME

The name of the z/OS database you created for the event database.

JDBC_CLASSPATH

The path to the DB2 JDBC driver.

UNIVERSAL_JDBC_DRIVER_NATIVEPATH

The path to the Universal JDBC native library path.

UNIVERSAL_JDBC_CLASSPATH

The path to the Universal JDBC driver.

10. Run the database configuration script for your client operating system, specifying the name of the database response file as a parameter. Use the following command:

```
config_event_database.sh response_file
```

The script configures the event database and creates two JDBC data sources: one for the event database and one for the event catalog. A message is displayed when database configuration is complete.

Note: If your database response file specifies EXECUTE_SCRIPTS=NO, you must complete the database configuration by manually running the generated scripts. The default value in the sample database response file is EXECUTE_SCRIPTS=YES.

After you configure the event database, you must restart the application server.

Manually running z/OS database configuration scripts:

If your database response file specifies EXECUTE_SCRIPTS=NO, you must complete the database configuration process by manually running the generated scripts.

To run the scripts manually:

1. Go to the *profile_path/event/dbscripts/db2zos* directory.
Replace *profile_path* with the path to the directory containing the profile for the WebSphere Process Server runtime environment in which you are configuring the event database.
2. Use the SQL Processor Using File Input (SPUFI) to load and run the generated DDL scripts. Run the scripts in the following order:
 - *profile_path/event/dbscripts/db2zos/ddl/cr_db.db2*
 - *profile_path/event/dbscripts/db2zos/ddl/cr_db_catalog.db2*
 - *profile_path/event/dbscripts/db2zos/ddl/cr_tbl.db2*
 - *profile_path/event/dbscripts/db2zos/ddl/cr_tbl_catalog.db2*
 - *profile_path/event/dbscripts/db2zos/ddl/ins_metadata.db2*
 - *profile_path/event/dbscripts/db2zos/ddl/catalogSeed.db2*
3. Go to the *profile_path/event/dsscripts/db2zos* directory.
Replace *profile_path* with the path to the directory containing the profile for the WebSphere Process Server runtime environment in which you are configuring the event database.

4. Run the `profile_path/event/dsscripts/cr_db2zos_jdbc_provider` script to create the event data source. Specify the scope at which the JDBC provider is to be configured:

```
cr_db2zos_jdbc_provider scope [server_name]
```

After the event database is configured, you must restart the application server.

DB2 Universal Database response file for z/OS systems:

A DB2 Universal Database response file specifies parameters for configuring a DB2 event database on a z/OS system.

A sample DB2 response file for z/OS systems, called `DB2ZOSResponseFile.txt`, is available in the `install_root/event/dbconfig` directory. This response file specifies the following parameters:

SERVER_NAME=*server*

The name of the WebSphere Process Server where the database is installed.

This parameter is applicable only if the **SHARE_DB** parameter is set to **server**.

If you do not specify a server name, the default value is `server1`.

CLUSTER_NAME=

The name of the cluster containing the WebSphere Process Server where the database is installed. This parameter is applicable only if the **SCOPE** parameter is set to **cluster**. The parameter will be ignored when scope is **cell** or **node**, and is not valid if the scope is **server**.

SCOPE=[server | node | cell | cluster]

The scope in which the configured database is shared. This is the scope in which Java database connectivity (JDBC) data sources are created. This parameter is optional. The default value is **server**.

DB_NAME=*name*

The name of the event database. This name must be no longer than 8 characters and must be the name of an existing database. This parameter is optional. The default value is `ceizos`.

JDBC_PROVIDER=*provider*

The name of the JDBC provider to configure. The value must be the name of a JDBC driver supported by WebSphere Process Server Version 5.1, and later. The following drivers are recommended:

- DB2 Universal JDBC Driver Provider (XA)
- DB2 Legacy CLI-based Type 2 JDBC Provider (XA)

JDBC_CLASSPATH=*path*

The path to the JDBC driver (not including file name). This should be one of the following drivers:

- For DB2 Universal JDBC Driver Provider (XA): the path to the `db2jcc_license_cu.jar` and `db2jcc_license_cisuz.jar` files.
- For DB2 Legacy CLI-based Type 2 JDBC Driver (XA), the path to the `db2java.zip` file.

UNIVERSAL_JDBC_CLASSPATH=*path*

For DB2 Universal JDBC Driver Provider or DB2 Universal JDBC Driver Provider (XA), the path to the JDBC driver (not including file name). This should be the path to the `db2jcc_license_cu.jar` file. This parameter is optional.

JDBC_DRIVER_TYPE=*type*

The JDBC driver type. This should be either 2 or 4.

DB_HOST_NAME=*hostname*

The database server host name. This parameter is required if JDBC_DRIVER_TYPE is set to 4. The default value is localhost.

DB_INSTANCE_PORT=*port*

The database instance port number. This parameter is required if JDBC_DRIVER_TYPE is set to 4. The default port number is 5027.

EXECUTE_SCRIPTS=[YES|NO]

Specifies whether the database configuration scripts are automatically run. If you are configuring the database on a z/OS system with UNIX System Services, set this value to NO.

DB_TYPE=DB2ZOS

The type of database to configure. For a DB2 for z/OS database, this must be DB2ZOS.

EVENT_DB_NAME=*name*

The database name for the event database. This name must be no longer than 8 characters. The default value is event.

CATALOG_DB_NAME=*name*

The database name for the event catalog database. This name must be no longer than 8 characters. The default value is eventcat.

STORAGE_GROUP=*group*

The storage group for the event database and catalog database. This must be the name of an existing storage group. The default value is sysdeflt.

BUFFER_POOL_4K=*name*

The name of the 4K buffer pool. The default value is BP9.

BUFFER_POOL_8K=*name*

The name of the 8K buffer pool. The default value is BP8K9.

BUFFER_POOL_16K=*name*

The name of the 16K buffer pool. The default value is BP16K9.

DAYS_TO_KEEP_EVENTS=*days*

The number of days that events are kept in the database before they are purged. Changes to this value significantly affect the amount of storage allocated for the table spaces that store event data. The default value is 1.

AVERAGE_EVENTS_PER_SECOND=*events*

The average number of events that are stored in the database each second. Changes to this value significantly affect the amount of storage allocated for the table spaces that store event data. The default value is 1.

AVERAGE_NUMBER_CONTEXT_PER_EVENT=*number*

The average number of context elements per event instance. The default value is 1.

AVERAGE_NUMBER_EXTENDED_DATA_ELEMENT_PER_EVENT=*number*

The average number of extended data elements per event instance. Changes to this value significantly affect the amount of storage allocated for the table spaces that store extended data element data. The default value is 5.

AVERAGE_NUMBER_EXTENDED_DATA_ELEMENT_ARRAY_ELEMENTS=*number*

The average number of values for extended data elements that are array data types. The default value is 5.

AVERAGE_NUMBER_MSG_TOKENS_PER_EVENT=*number*

The average number of message tokens per event. The default value is 1.

AVERAGE_ASSOCIATIONS_PER_EVENT=*number*

The average number of event associations per event. The default value is 2.

TABLESPACE_EXTENDED_BINARY_VALUE_PRIMARY=*size*

The primary allocation for the large object (LOB) table space that contains hexBinary extended data element values. This allocation can be small if events do not typically contain hexBinary extended data element values. The default value is 1000.

TABLESPACE_EXTENDED_BINARY_VALUE_SECONDARY=*size*

The secondary allocation for the large object (LOB) table space that contains hexBinary extended data element values. This allocation can be small if events do not typically contain hexBinary extended data element values. The default value is 200.

TABLESPACE_ANY_VALUE_PRIMARY=*size*

The primary allocation for the large object (LOB) table space that contains the values for the *any* element, which is a character large object (CLOB). This allocation can be small if events do not typically contain *any* elements. The default value is 1000.

TABLESPACE_ANY_VALUE_SECONDARY=*size*

The secondary allocation for the large object (LOB) table space that contains the values for the *any* element, which is a character large object (CLOB). This allocation can be small if events do not typically contain *any* elements. The default value is 200.

PERCENTAGE_FREE_SPACE=*percent*

The amount of free space, as a percentage, to leave on each page. Increase this value as the number of inserted rows increases. Free space makes updates more efficient, but a larger value uses more disk space. The default value is 20.

FREE_PAGE=*pages*

The number of pages to fill before leaving a free page. If this parameter is set to 0, free pages are not left. Set this parameter to a nonzero value if a large amount of SQL INSERT processing is expected. (A nonzero value uses more disk space.) The default value is 10.

NUMBER_EVENT_DEFINITIONS=*definitions*

The number of event definitions stored in the event catalog. The default value is 100.

AVERAGE_SOURCE_CATEGORY_PER_EVENT_DEFINITION=*categories*

The average number of source categories per event definition in the event catalog. The default value is 1.

AVERAGE_EXTENDED_DATA_ELEMENT_PER_EVENT_DEFINITION=*definitions*

The average number of extended data element descriptions for each event definition in the event catalog. The default value is 5.

AVERAGE_PROPERTY_DESCRIPTIONS_PER_EVENT_DEFINITION=*definitions*

The average number of property descriptions for each event definition in the event catalog. The default value is 5.

TABLESPACE_HEX_DEFAULT_PRIMARY=*size*

The primary allocation for the large object (LOB) table space that contains the default values for hexBinary extended data elements. The default value is 100.

TABLESPACE_HEX_DEFAULT_SECONDARY=*size*

The secondary allocation for the large object (LOB) table space that contains the default values for hexBinary extended data elements. The default value is 10.

Database configuration logs and messages

The scripts for configuring and removing the event database create two log files.

- The *profile_root*/logs/events_db_install_trc.log log file contains detailed trace information.
- The *profile_root*/logs/events_db_install_msg.log log file contains any messages generated by the database configuration script.

Log file messages are in the following format:

```
<Date> <month><year> <time><Class> <Methods><Type> <Message>
```

The fields in the message statements are as follows:

Class The name of the class generating the message.

Method

The method generating the log message.

Type This can be any of the following types of messages:

- Entry
- Exit
- Error
- Information
- Warning

Message

The text of the message.

Deploying the Common Event Infrastructure application

The event server enterprise application must be deployed in each WebSphere runtime environment where the profile has been augmented to use the Common Event Infrastructure.

The event server enterprise application is packaged in the event-application.ear EAR file. The **event-application.jacl** script installs this application in the WebSphere Process Server.

1. Go to the *profile_path*/event/application directory for the profile defining the WebSphere Process Server runtime environment where you want to deploy the application. (Replace *profile_path* with the path to the directory containing the WebSphere Process Server profile.)
2. Run the **event-application.jacl** script using the **wsadmin** command:

Windows systems

```
profile_path\bin\wsadmin [-conntype none] -profile event-profile.jacl  
-f event-application.jacl -action action -earfile event-application.ear  
-backendid backend_id -node node_name -server server_name  
[-cluster cluster_name] [-appname app_name] [-trace]
```

Linux and UNIX systems

```
profile_path/bin/wsadmin.sh [-conntype none] -profile event-profile.jacl  
-f event-application.jacl -action action -earfile event-application.ear  
-backendid backend_id -node node_name -server server_name  
[-cluster cluster_name] [-appname app_name] [-trace]
```

The parameters are as follows:

action

The action to perform. To install the enterprise application, specify `install`. To update an existing event server application that is already installed, specify `update`. During an update, the script makes a backup copy of the existing application EAR file in the current directory; if necessary, you can later use this backup copy to restore the previous version of the application.

backend_id

The type of database back end to be used by the enterprise application. This must be one of the following values:

- `CLOUDSCAPE_V51_1`
- `DB2UDBNT_V82_1`
- `DB2UDBNT_V8_1`
- `ORACLE_V10_1`
- `ORACLE_V9_1`
- `DB2UDBOS390_V8_1`
- `DB2UDBOS390_V7_1`

node_name

The WebSphere Process Server node in which the event server is to be deployed. To discover the node name, follow these steps:

- a. Run the `profile_path/bin/setupCmdLine` script.
- b. Run the command `echo $WAS_NODE` (Linux/UNIX systems) or `echo %WAS_NODE%` (Windows systems).

This value is case-sensitive. If you are deploying the enterprise application in a cluster, omit this parameter.

server_name

The WebSphere server into which the event server enterprise application is to be deployed. This value is case-sensitive. If you are deploying the enterprise application in a cluster, omit this parameter.

app_name

The name to use for the Common Event Infrastructure enterprise application. This parameter is optional; the default value is `CommonEventInfrastructureServer`.

The optional `-trace` parameter causes additional debugging information to be displayed on the standard output.

Note:

- If you specify a fully qualified path for the location of the `event-application.ear` file, make sure you use forward slashes (/) in the path, even on Windows systems.
- If you are deploying the application on a stand-alone node, specify the optional `-conntype none` parameter to run `wsadmin` in local mode.

For more information about the `wsadmin` utility, refer to the WebSphere Application Server documentation.

After the `event-application.jacl` script completes, the Common Event Infrastructure enterprise application is deployed in the specified server or cluster. In a WebSphere Process Server Network Deployment environment, if the application is already installed, the script only adds the deployment information for the specified node and server.

Configuring event messaging

If you want to use Java Message Service (JMS) queues for asynchronous message transmission to the event server, you must configure event messaging.

Configuring default event messaging

The default messaging configuration for asynchronous event transport uses the WebSphere Process Server default messaging feature as the Java Message Service (JMS) provider.

The **default-event-message.jacl** script provides a way to quickly set up a default messaging configuration, using the WebSphere default messaging feature as the JMS provider. This script sets up all of the configuration objects required for asynchronous event transmission:

- It creates a JMS queue and a queue connection factory using the default messaging feature.
- It creates a service integration bus and adds members to the bus, associating the bus with queues, topics, and connection factories.
- It creates a JMS transmission profile using the created queue and connection factory.
- It configures the default emitter factory profile to use the created JMS transmission profile for asynchronous event transmission.
- It deploys the message-driven bean used by the Common Event Infrastructure to receive events sent asynchronously to the event server.

To configure default messaging:

1. Go to the *profile_path*/event/application directory for the profile defining the WebSphere Process Server runtime environment where you want to configure default messaging. (Replace *profile_path* with the path to the directory containing the WebSphere Process Server profile.)
2. Run the **default-event-message.jacl** script using the **wsadmin** command:

```
profile_path/bin/wsadmin.sh [conntype -none] -profile event-profile.jacl  
-f default-event-message.jacl -action install -earfile event-message.ear  
-node node_name -server server_name  
[-cluster cluster_name] [-appname app_name] [-trace]
```

The parameters are as follows:

node_name

The WebSphere Process Server node in which the messaging application is to be deployed. To determine the node name:

- a. Run the *profile_path*/bin/setupCmdLine script.
- b. Run the command `echo $WAS_NODE` (Linux/UNIX systems) or `echo %WAS_NODE%` (Windows systems).

This value is case-sensitive. If you are deploying the application in a cluster, omit this parameter.

server_name

The WebSphere server into which the messaging application is to be deployed. This value is case-sensitive. If you are deploying the application in a cluster, omit this parameter.

app_name

The name to use for the messaging enterprise application. This parameter is optional; the default value is EventServerMdb.

The optional **-trace** parameter causes additional debugging information to be displayed on the standard output.

Note:

- If you specify a fully qualified path for the location of the event-message.ear file, make sure you use forward slashes (/) in the path, even on Windows systems.
- If you are deploying the messaging application on a stand-alone node, specify the optional **-conntype none** parameter to run **wsadmin** in local mode

After you start the script, you are prompted for your JMS user ID and password.

Configuring event messaging using another JMS provider

If you do not want to use the WebSphere Process Server default messaging feature for event transmission, you can configure asynchronous message transport to use a different Java Message Service (JMS) provider.

Before you can configure event messaging using an external JMS provider, you must first create a JMS queue and connection factory using the appropriate interfaces for your JMS provider.

The **event-message.jacl** script sets up the configuration objects required for asynchronous event transmission using an external JMS provider such as WebSphere MQ:

- If a scope is specified, then it creates a JMS transmission profile using the JMS queue and connection factory you specify.
- It creates an emitter factory profile using the created JMS transmission profile for asynchronous event transmission.
- It deploys the message-driven bean used by the Common Event Infrastructure to receive events sent asynchronously to the event server, using either a listener port or a JMS activation specification.

If you want to set up more than one JMS queue to the event server, you can run this script multiple times, specifying different enterprise application names and JMS queues. Each time you run the script, it deploys an additional message-driven bean and configures new resources to use the specified JMS queue.

To configure event messaging to use an external JMS provider:

1. Go to the *profile_path*/event/application directory for the profile defining the WebSphere Process Server runtime environment where you want to configure default messaging. (Replace *profile_path* with the path to the directory containing the WebSphere Process Server profile.)
2. Run the **event-message.jacl** script using the **wsadmin** command:

```
profile_path/bin/wsadmin -profile event-profile.jacl -f event-message.jacl  
-action install -earfile event-message.ear -node node_name  
[-server server_name] [cluster cluster_name]  
-appname app_name -qjndi queue -qcfjndi connection_factory  
[-listenerport listener_port] [-activationspecjndi spec_name]  
[-eventprofilescope scope] [-trace]
```

The parameters of the **event-message.jacl** script are as follows:

node_name

The WebSphere Process Server node in which the messaging application is to be deployed. To determine the node name:

- a. Run the *profile_path/bin/setupCmdLine* script.
- b. Run the command `echo $WAS_NODE` (Linux/UNIX systems) or `echo %WAS_NODE%` (Windows systems).

This value is case-sensitive. If you are deploying the application in a cluster, omit this parameter.

server_name

The WebSphere server into which the messaging application is to be deployed. This value is case-sensitive. If you are deploying the application in a cluster, omit this parameter.

The optional **-trace** parameter causes additional debugging information to be displayed on the standard output.

Note:

- If you specify a fully qualified path for the location of the event-message.ear file, make sure you use forward slashes (/) in the path.
- If you are deploying the messaging application on a stand-alone node, specify the optional `-conntype none` parameter to run **wsadmin** in local mode

After you start the script, you are prompted for your JMS user ID and password.

Runtime configuration

You can configure the Common Event Infrastructure resources using the WebSphere Process Server administrative console, or from the command line with the **wsadmin** tool.

To see the Common Event Infrastructure resources in the administrative console, expand the **Resources** list and click **Common Event Infrastructure Provider**. The types of resources are displayed in the **Additional Properties** list. From here you can navigate to the individual resources of each type (for example, you can view a list of all event group profiles or emitter factory profiles). To change the configuration of a resource, click the resource name in the list and then edit the properties you want to change.

Refer to the WebSphere Process Server topic `../com.ibm.wsps.602.adm.doc/overview/cadm_overviewadmin.dita` for more information about the administrative console and the **wsadmin** tool.

In most circumstances, only certain properties need to be configured. For more information about these resources and their properties, refer to the online help for the Common Event Infrastructure resources in the WebSphere administrative console.

Note: After changing the Common Event Infrastructure configuration, you must restart the WebSphere server.

Default configuration

The Common Event Infrastructure components are installed as a set of WebSphere Process Server applications, services, and default resources.

You can customize the Common Event Infrastructure by configuring the provided resources or creating additional resources; for more information, see “Runtime configuration” on page 25.

The default configuration consists of the following objects:

Common Event Infrastructure service

A service installed into the WebSphere server. This service enables WebSphere applications and clients to use the Common Event Infrastructure.

Common Event Infrastructure enterprise application

The enterprise application for the event server. The deployment descriptor of the enterprise application associates the event server with the Common Event Infrastructure resources it uses.

Common Event Infrastructure messaging application

The enterprise application for the message-driven bean that supports asynchronous event transmission to the event server. This application is available only if you have configured event messaging; for more information, see “Configuring default event messaging” on page 23.

Common Event Infrastructure Provider

A collection object containing the resources used by Common Event Infrastructure components, event sources, and event consumers.

Data store profile

A data store profile defines properties used by the default data store plug-in, which is used to persistently store events received by the event server. A default data store profile is provided; typically, no configuration is necessary for this resource, but in some circumstances you might want to adjust some properties for your environment. You might also need to create additional data store profiles if you want to set up multiple event servers in the same cell.

Event bus transmission profile

An event bus transmission profile defines properties used by emitters to access the event server synchronously using EJB calls; these profiles are used by emitter factory profiles. A default transmission profile is provided; typically, no configuration is necessary for this resource.

Event group profile list

An event group profile list is a collection containing the event group profiles used by the event server. The event group profile list used by an event server is specified in the deployment descriptor of the event server enterprise application. Typically, no configuration is necessary for this resource, but you might need to create additional event group profile lists if you want to set up multiple event servers in the same cell.

Event group profile

An event group profile defines an event group (a logical collection of events). Event groups are used to categorize events according to their content; when querying events from the event server or subscribing to event distribution, an event consumer can specify an event group to retrieve only the events in that group.

A default event group profile is provided; this profile defines an event group containing all events, and is associated with the JMS topic

jms/cei/notification/AllEventsTopic. You can create additional event group profiles specifying whatever event criteria are appropriate for your application.

Emitter factory profile

An emitter factory profile defines properties used by emitters. The properties in an emitter factory profile affect the behavior of any emitter that is created using the associated emitter factory. The default emitter factory profile specifies synchronous transmission, no filtering, and sending each event as part of the current transaction. You might want to create an additional emitter factory profile to specify a different transaction mode or transmission profile.

Event server profile

A profile defining properties used by the event server. The default event server profile enables event distribution and persistence, and it is configured to use the default data store plug-in. Typically, no configuration is necessary for this resource, but you might need to create additional event server profiles if you want to set up multiple event servers in the same cell.

JMS Transmission Profile

A JMS transmission profile defines properties used by emitters to access the event server asynchronously using a JMS queue; it is referenced by emitter factory profiles. This profile is available only if you have configured event messaging; for more information, see “Configuring default event messaging” on page 23.

Configuring the Events service

Perform these steps to switch the Events service on or off, and specify the JNDI name of the event emitter profile factory that is used to submit events.

This task assumes you have

- installed the Common Event Infrastructure server
- installed any Common Event Infrastructure Message-Driven Bean applications
- configured all the Common Event Infrastructure resources
- logged into the WebSphere Process Server administrative console

The Events service provides access to the Common Event Infrastructure server and ensures that information about the WebSphere server, and correlation sphere information, is automatically included in each event passed to the event infrastructure.

You configure the Events service by modifying the **Enable service at server startup** property; this specifies whether the service is started automatically for a specific application server. (You can also change the JNDI name of the Common Event Infrastructure emitter factory. See Step 5.)

CAUTION: Disabling the Events service will prevent the Common Event Infrastructure Server from processing any events. No event information will be persisted in the database, nor will events be distributed to JMS destinations.

1. In the administrative console navigation pane, click **Servers > Application Servers**. A list of the application servers is displayed in the content pane.
2. Click into the Application server you want to configure.

3. In the **Business Integration** table, select **Common Event Infrastructure Destination** . The Events service properties are displayed in the Configuration tab.
4. Under General Properties, select or clear the **Enable service at server startup** property check box.

Selected

[Default] The Events service starts when the application server starts. This enables applications that generate events to run on the application server.

Important: The Events service is utilized by event monitoring and should be enabled all the time.

An enabled Events service allows information about the WebSphere server, and correlation sphere information, to be automatically included in each event that will be passed to the event infrastructure.

Cleared

The Events service does not start when the application server starts. The Events service is utilized by event monitoring, and a disabled Events service will not automatically gather and insert information about the WebSphere server into each event, and also will not create any correlation sphere information.

Any attempt to start an application that uses the Events service is rejected, and a message is issued. The server will continue to start without the application.

5. Review the Java Naming and Directory Interface (JNDI) name of the event emitter profile factory that is used to submit events to the event infrastructure. You can select the JNDI name using one of two methods:

Drop-down menu

By default, this method is selected, and will specify a default emitter factory JNDI from the Common Event Infrastructure Provider, although there may be other names available on this menu. The default value on this menu will be `com/ibm/events/configuration/emitter/Default`, which is the default emitter factory JNDI from the Common Event Infrastructure Provider and is part of the WebSphere default profile. Unless you have generated an alternate profile, accept the default.

Text entry box

You can select this and fill in the JNDI name if you do not see the appropriate name in the pull-down menu.

6. To save and apply your configuration, click the **OK** button.
7. Stop and restart the Application server for the changes to take effect.

DB2 database maintenance

If you are using a DB2 event database, you should periodically perform database maintenance by running the provided scripts.

The DB2 versions currently supported for running these maintenance scripts are

- DB2 Universal Version 8.1
- DB2 Universal Version 8.2.1

Updating database statistics

To enable the DB2 database to optimize queries and find free space, update the database statistics using the `runstats` script.

It is recommended that you update the database statistics regularly, and especially under any of these circumstances:

- Events have been purged from the database
- A large number of events have been inserted into the database
- Tables have been reorganized using the `reorg` script
- Indexes have been added or removed from a table

The `runstats` script is located in the `install_root/event/dbscripts/db2` directory.

To update the database statistics, run the following command:

```
runstats.sh db_alias db_user [db_password]
```

The parameters are as follows:

db_alias

The database alias. The event database must be catalogued on the DB2 client; if you are running the script on the DB2 server, the database is already catalogued.

db_user

The database user ID to use. This parameter is required.

db_password

The database password. This parameter is optional. If you do not specify the password on the command line, the DB2 database will prompt you for it.

For example, the following command updates the DB2 database statistics where the event database name is `event`, the database user ID is `dbadmin`, and the password is `mypassword`:

```
runstats.sh event dbadmin mypassword
```

Reorganizing database tables

After events are purged from a DB2 event database, reorganize the database tables using the `reorg` script.

The `reorg` script is located in the `profile_root/event/dbscripts/db2` directory.

To reorganize the event database tables, run the following command:

```
reorg.sh db2_alias db_user [db_password]
```

The parameters are as follows:

db2_alias

The database alias. The event database must be catalogued on the DB2 client; if you are running the script on the DB2 server, the database is already catalogued.

db_user

The database user ID to use. This parameter is required.

db_password

The database password. This parameter is optional. If you do not specify the password on the command line, the DB2 database prompts you.

For example, the following command reorganizes the event database tables using the database user ID `dbadmin`, the password `mypassword`, and the database name is `event`:

```
reorg.sh event dbadmin mypassword
```

After you run the `reorg` script, you should update the database statistics using the `runstats` script. For more information, see [Updating database statistics](#).

Purging events from the event database

You can use the provided scripts to rapidly purge large numbers of events from the event database.

The default data store plug-in provides a set of utilities you can use to periodically perform a rapid purge of large numbers of old events from the event database. These utilities are distinct from the **eventpurge.jacl** event server command, which deletes events matching specified criteria.

The rapid purge capability uses the concept of **buckets**. A bucket is a set of tables used to store events in the event database. The default data store plug-in uses two buckets:

- The **active bucket** is the bucket containing the most recent events; new events are stored in the active bucket. The active bucket cannot be purged using the rapid purge utility.
- The **inactive bucket** contains older events. Events stored in the inactive bucket can be queried, deleted, or modified, but typically no new events are stored in the inactive bucket. The inactive bucket can also be purged by the rapid purge utility.

Each event is stored in only one bucket. From the perspective of an event consumer, the distinction between the active and inactive buckets is invisible; a consumer can query, modify, or delete a specific event without knowing which bucket the event is stored in. The advantage of this approach is that the inactive bucket can be rapidly purged using database-specific interfaces without affecting the active bucket; normal event traffic can continue even while the purge operation is taking place.

After the inactive bucket is purged, you can then swap the buckets so that the active bucket becomes inactive and the inactive bucket becomes active. Swapping buckets is possible only when the inactive bucket is empty.

Note: Although new events are generally stored only in the active bucket, under some circumstances events might be stored in the inactive bucket immediately after the buckets are swapped. The data store plug-in checks periodically to determine which bucket is currently marked as active, but until the next check takes place, some events might continue to be stored in the inactive bucket. Similarly, events sent as part of a batch are all stored in the same bucket, even if that bucket becomes inactive while the batch is still being processed.

If you want to use the fast purge capability, it is your responsibility to determine how frequently to swap buckets or purge the inactive bucket, depending upon event traffic, storage space, archival requirements, or other considerations.

Viewing or changing the active bucket status

The active bucket status indicates which bucket is currently active and which is currently inactive.

To view or change the active bucket status, use the **eventbucket.jacl** script (found in the *profile_root/event/bin* folder):

```
wsadmin -f eventbucket.jacl [-status] [-change]
```

This command has the following options:

-status

Use this option to see information about the current bucket configuration, including the active bucket setting and the bucket check interval (the frequency with which the data store plug-in checks to determine which bucket is active).

-change

Use this option to swap the active and inactive buckets. The inactive bucket must be empty before you can use this option.

Purging the inactive bucket

The method used to purge the inactive bucket varies depending on the database software.

Note: The rapid purge utility is not supported for Cloudscape databases.

Purging the inactive bucket for a DB2 database (z/OS systems):

On z/OS systems, the rapid purge utility for a DB2 event database is implemented using the DB2 load utility.

To purge the inactive bucket:

1. Use the **eventbucket.jacl** command to identify the inactive bucket (bucket 0 or bucket 1).
2. Upload the appropriate utility control file. These files are generated during database configuration and are located in the *profile_path/event/dbscripts/db2zos* directory (*profile_path* is the path to the directory containing the profile for the WebSphere Process Server runtime environment. Upload one of the following files:
 - If bucket 0 is inactive, upload fastpurge00.ctl.
 - If bucket 1 is inactive, upload fastpurge01.ctl.

Note: The control file must be uploaded with a fixed record format and a logical record length of 80.

3. On the z/OS host, go to the ISPF DB2I Primary Option Menu and click the **Utilities** option.
4. Specify the following information:

Field	Value
Function	EDITJCL
Utility	LOAD

Field	Value
Statement Data Set	The name of the data set containing the uploaded control file
LISTDEF	NO
Template	NO

5. Press the Enter key to continue to the next panel.
6. In the recdsn entry field, specify the name of the data set containing the uploaded control file.
7. Press the Enter key. The JCL script to purge the inactive bucket is generated.
8. Press the Enter key to clear the output messages.
9. Edit the generated JCL script as needed.
10. Submit the JCL script.

Changing the bucket check interval

The bucket check interval is specified in the DataStoreEjb.jar file.

This value specifies how frequently the data store plug-in checks to determine which bucket is active. The default value is 5 minutes (300 seconds). A shorter interval reduces the likelihood that events will be stored in the inactive bucket after swapping, but might decrease performance.

To change the bucket check interval:

1. In the WebSphere Process Server administrative console, navigate to **Enterprise Applications > EventServer > EJB Modules > DataStoreEjb.jar > View Deployment Descriptor**.
 - a. Look for the DefaultDataStoreEJB parameters.
 - b. Expand the `<session id="DataStoreHelperEJB">` menu, and look for the `<env-entry-value>`.

This is the bucket check interval value, which is set in number of seconds.
2. Modify the value of the BucketCheckInterval environment variable to specify the bucket check interval in seconds.

Chapter 3. Viewing events with the Common Base Event browser

Use the Common Base Event browser to select, sort, and view events.

This task assumes you are logged into the WebSphere Process Server administrative console.

The event browser uses the event access interface to query event data. The results of the query are shown in the browser.

1. Begin by opening the event browser. Click **Integration Applications** and then **Common Base Event Browser** in the navigation pane of the administrative console.
2. Specify the events you want to view.
3. Select the view of the returned events.
4. In any of the browser panels, when you have finished selecting search or sort criteria, click the **Get Events** button at the bottom of the browser panel to display the events.

Specifying the events to view

How to use the Common Base Event browser to specify search criteria for querying events in the event database.

This task assumes that you have already opened the event browser and are viewing the Get Events panel.

The Event Data Store Properties fields require completion. The Event Filter Properties fields are optional, and you can narrow your events search based on time, date, server name, sub-component name, and event severity parameters.

1. **Required:** Specify the Event Data Store to search.

The field is a Java Naming and Directory Interface (JNDI) name, an Enterprise JavaBeans (EJB) reference that can be configured in the administrative console. The WebSphere Process Server default is `java:comp/env/eventsaccess`, but you may need to specify a different name if you are working in a Network Deployment environment (see the WebSphere Application Server for z/OS Information Center for more details about JNDI naming).

2. **Required:** Specify the Event Group to search.

This is the event group from which events are retrieved. The default group is All events.

3. **Required:** Specify the number of events to retrieve.

The maximum number of events to search is 500.

4. **Optional:** Specify the Creation Date (calendar period) for the report.

Enter the start and end dates.

5. **Optional:** Specify the Creation Time (time period) for the report.

Enter the start and end times.

6. **Optional:** Specify the server name.

7. **Optional:** Specify the sub-component name, if applicable.

8. **Optional:** Specify an event's priority. The range of events priorities to retrieve is from 0 (lowest priority) to 100 (highest priority).
9. **Optional:** Specify an event's severity.
The range of events severities to retrieve is from 0 (least severe) to 70 (most severe).
10. Click **Get Events**.

The number of Common Base Events matching the search criteria is displayed. If the results you queried do not appear, refer to the *Troubleshooting and Support* PDF file.

To view the returned events, select a view from the navigation bar. You can click **All Events**, **BPEL Process Events**, **User Data Events**, or **Server Events**. When you view event data, you can change your search criteria at any time by clicking **Get Events**.

After events are returned, you can work with them to get various levels of event detail.

Working with events returned from the event browser

You use the event browser to view the events returned from a query.

This task acts on data that is returned by a submitted query, as described in the topic *Specifying the events to view*.

The query returns all the events that meet your criteria.

1. Click a view from the navigation bar.

The navigation bar offers the following views of the returned query:

All Events

All the events returned.

BPEL Process Events

Business Process Choreographer events for a specific process instance.

User Data Events

Events with the extension name ECS:UserDataEvent. This event type is created by the addUserDataEvent method of the ECSEmitter class.

Server Events

Events for a specific server.

2. Perform one of the following actions.
 - If you click **BPEL Process Events** in step 1, you must click a process template, and then a process instance.
 - If you click **Server Events** in step 1, you must click a server.
3. Click an event, to display the event data in the pane at the bottom of the browser window.

Chapter 4. Security and the Common Event Infrastructure

You can use WebSphere method-level declarative security to restrict access to Common Event Infrastructure functions.

The Common Event Infrastructure defines six security roles, each one associated with a related group of functions. These security roles control access to both programming interfaces and commands. (The default configuration of the Common Event Infrastructure does not require the use of these roles; however, in a Network Deployment environment, the WebSphere Process Server needs to be authenticated with the same users assigned to the Common Event Infrastructure security roles. For more information about security roles, see *Learning about security and Role-based authorization in the WebSphere Application Server Information Center*.) If you are already a WebSphere Process Server authenticated user, and global security is turned on, you can access the Common Event Infrastructure resources.

Note:

If the security roles are used by mapping specific users to the roles, the authenticated users need to be the same users as assigned to the security role. For additional information about authenticated users and the RunAs role, see *Assigning users to RunAs roles*.

The following table describes the security roles and the types of users associated with each role.

Table 1. Security roles and user types

Security role	User types
eventAdministrator	Event consumers that need to query, update, and delete events stored in the event database. This role provides access to the following interfaces: <ul style="list-style-type: none">• EventAccess.purgeEvents()• EventAccess.eventExists()• EventAccess.queryEventByGlobalInstanceId()• EventAccess.queryEventsByAssociation()• EventAccess.queryEventsByEventGroup()• EventAccess.updateEvents()• Emitter.sendEvent()• Emitter.sendEvents()• eventquery.jacl• eventpurge.jacl• emitevent.jacl• eventbucket.jacl
eventConsumer	Event consumers that need to query events stored in the event database. This role provides access to the following interfaces: <ul style="list-style-type: none">• EventAccess.eventExists()• EventAccess.queryEventByGlobalInstanceId()• EventAccess.queryEventsByAssociation()• EventAccess.queryEventsByEventGroup()• eventquery.jacl

Table 1. Security roles and user types (continued)

Security role	User types
eventUpdater	<p>Event consumers that need to update events stored in the event database. This role provides access to the following interfaces:</p> <ul style="list-style-type: none"> • EventAccess.updateEvents() • EventAccess.eventExists() • EventAccess.queryEventByGlobalInstanceId() • EventAccess.queryEventsByAssociation() • EventAccess.queryEventsByEventGroup() • eventquery.jacl
eventCreator	<p>Event sources that need to submit events to an emitter using synchronous EJB calls. This role provides access to the following interfaces:</p> <ul style="list-style-type: none"> • Emitter.sendEvent() • Emitter.sendEvents() • emitevent.jacl <p>Note: The eventCreator role restricts access to event submission only if the emitter is configured to use synchronous EJB calls for event transmission. If the emitter uses asynchronous JMS messaging for event transmission, you must use JMS security to restrict access to the destination used to submit events.</p>
catalogAdministrator	<p>Event catalog applications that need to create, update, delete, or retrieve event definitions in the event catalog. This role provides access to all methods of the EventCatalog interface and all functions of the eventcatalog.jacl script. Because changes to the event catalog can result in generation of events, this role also provides access to event submission interfaces.</p>
catalogReader	<p>Event catalog applications that need to retrieve event definitions from the event catalog. This role provides access to the following interfaces:</p> <ul style="list-style-type: none"> • EventCatalog.getAncestors() • EventCatalog.getChildren() • EventCatalog.getDescendants() • EventCatalog.getEventDefinition() • EventCatalog.getEventDefinitions() • EventCatalog.getEventExtensionNamesForSourceCategory() • EventCatalog.getEventExtensionToSourceCategoryBindings() • EventCatalog.getParent() • EventCatalog.getRoot() • EventCatalog.getSourceCategoriesForEventExtension() • eventcatalog.jacl (-listdefinitions option) • eventcatalog.jacl (-listcategories option) • eventcatalog.jacl (-exportdefinitions option)

Note:

The security roles most relevant to utilizing the functionality of the Common Event Infrastructure are **eventAdministrator** and **eventConsumer** .

The event server message-driven bean runs using the WebSphere Process Server user identity. If you are using asynchronous JMS transmission to submit events to the event server, and you have enabled method-based security, you must map this user identity to the eventCreator role.

Note:

If Java 2 security is enabled, you must modify your policy file to enable access to certain functions:

- If you are running an event source application and you want to generate your own globally unique identifiers (GUIDs), add the following entries:
permission java.io.FilePermission "\${java.io.tmpdir}\${/}guid.lock",
"read, write, delete";
permission java.net.SocketPermission "*", "resolve";
- If you are using the default filter plug-in or the notification helper to filter events using XPath event selectors, add the following entries:
permission java.util.PropertyPermission "*", "read";
permission java.io.FilePermission
"\${was.install.root}\${/}java\${/}jre\${/}lib\${/}jxpath.properties",
"read";

Chapter 5. Troubleshooting the Common Base Event browser

There are four primary conditions under which you are unable to access the Common Base Event browser.

Conditions

“Cannot find server”

WebSphere Process Server (or network server) is unavailable. When you attempt to launch the event browser URI, a “Cannot find server” browser page will be returned, which indicates that the server is unavailable. In this case, you need to contact the process server administrator to determine the cause of the problem.

“File not found”

WebSphere Process Server is available; however, the event browser application may not be installed or started on the server. When you attempt to launch the event browser URI, a “File not found” browser page will be returned, which indicates that the server is available, but the URI is not available on that server. In this case, you need to contact the process server administrator to determine the cause of the problem.

Logon panel appears

The WebSphere Process Server and the event browser are available; however, you have not been mapped to the proper role to allow access to the event browser. You will be prompted with a logon panel. When you enter your userID and password, attempting to log in, the login will fail. In this case, you need to contact the process server administrator to get the proper authorization to launch the event browser.

Error message on “Get event data” panel

The WebSphere Process Server and the event browser are available, and you have the proper authority to gain access; however, the Common Event Infrastructure server is unavailable. An error message will be displayed on the event browser **Get Events** panel, when you click the **Get Events** button. The error information is logged to the message log.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing 2-31 Roppongi 3-chome, Minato-ku Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation 577 Airport Blvd., Suite 800 Burlingame, CA 94010 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows: © (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks and service marks

IBM and related trademarks: <http://www.ibm.com/legal/copytrade.shtml>

Other company, product, or service names may be trademarks or service marks of others.

This product includes software developed by the Eclipse Project (<http://www.eclipse.org/>).



WebSphere Process Server for z/OS, Version 6.0.2



Printed in USA