WebSphere® Process Server for z/OS

Version 6.0.1

Installing

Note
Before using this information, be sure to read the general information in "Notices" on page 253.

23 June 2006

This edition applies to version 6, release 0, modification 1 of WebSphere Process Server for z/OS (product number 5655-N53) and to all subsequent releases and modifications until otherwise indicated in new editions.

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

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

Contents

Installing

The Installing section describes how to prepare for, install, enable and configure an installation of IBM WebSphere® Process Server for z/OS, Version 6.0.1

WebSphere Process Server documentation (in PDF format)

The following sections provide information on how to:

- obtain overview information about WebSphere Process Server
- prepare your operating system for installation
- unload the product code from the tape onto the z/OS system
- plan for use of the Business Process Choreographer and Common Event Infrastructure components
- use the shell scripts to enable the WebSphere Process Server product for use and to augment its profile configuration data
- download online help files
- prepare for coexistence scenarios
- uninstall WebSphere Process Server
- configure, patch, and troubleshoot the installation.

Install concepts

Install concepts refers to the information that will help you understand the WebSphere Process Server for z/OS installation by providing high-level descriptions of both the processes and methods that one can use to install, enable and configure the product.

Before you begin the task of installing, enabling for use and configuring WebSphere Process Server for z/OS, you should read all of the information contained within the **Install concepts** help container.

Understanding these concepts will facilitate theWebSphere Process Server for z/OS install procedure.

z/OS install terminology

z/OS install terminology refers to the terms and phrases used throughout the installation documentation for WebSphere Process Server for z/OS. Becoming familiar with these terms and phrases will facilitate an understanding of the z/OS install concepts and tasks.

Key Terms

Because the product installs on top of WebSphere Application Server for z/OS, you should also become familiar with WebSphere Application Server for z/OS terminology as well. See WebSphere Application Server for z/OS terminology for descriptions of WebSphere Application Server for z/OS components and configuration types.

Although you may be familiar with these terms in general, the definitions below define the terms *within the context of the product install.*

Custom-built Product Delivery Option (CBPDO)
A software delivery package consisting of uninstalled products and unintegrated service. Installing what you have ordered in the CBPDO (stand-alone products and/or z/OS elements and features) requires the use of SMP/E. CBPDO is one of the two entitled methods for installing z/OS, ServerPac being the other.

Ant script
An Ant script is an XML file that is used to define targets which run various *Ant tasks* as part of your build procedure. The Ant scripts are associated with a configuration type. The actions registry sets the priority / sequence in which the Ant scripts run.

command line
The blank line on a display where commands, option numbers, or selections can be entered.

After you unload the contents of the install media using SMP/E, you run the product install and configuration scripts from a command line. The command line install for the product is silent, in that it does not send messages to the console but instead stores messages and errors in log files.

The command line install for this product does not prompt you for input values.

configuration type
The server types associated with the WebSphere Application Server for z/OS configuration onto which you will install, enable and configure WebSphere Process Server for z/OS. WebSphere Application Server for z/OS configuration types are as follows:

- Stand-alone application server cell
- Network Deployment cell
- Managed node in a Network Deployment cell

The WebSphere Process Server for z/OS configuration script points to a response file that contains property values. These property values are used to augment the WebSphere Application Server for z/OS default profile with WebSphere Process Server for z/OS configuration data.

For detailed information on the server types, including a description of each type and information on how to install and configure WebSphere Application Server for z/OS for each of these server types, see the WebSphere Application Server for z/OS information center.

HFS Hierarchical file system (HFS). A data set that contains a POSIX-compliant file system, which is a collection of files and directories organized in a hierarchical structure, that can be accessed using z/OS UNIX System Services. All files within an HFS are members of a directory. Each directory is, in turn, a member of another directory at a higher level in the hierarchy.

When you unload the contents of the install media using SMP/E, the result is an HFS commonly referred to as the *root HFS*. This root HFS is read-only.

Once UNIX System Services recognizes a root HFS, other HFSs can be mounted at paths off of this root. The root HFS that is defined may contain directories such as /tmp, /bin, /lib, /etc, /usr/lpp and so forth. For any application that resides in UNIX Systems Services, an HFS exists to hold the product executables and the parameter files.

After using SMP/E to create the WebSphere Process Server for z/OS HFS, you can enable the product for use by running the install script named **zSMPInstall.sh**. By running the install script you create *symlinks* to the read-only install HFS. These symlinks point to the read-only files in the install root, such as JAR files, shell scripts, and so on.

ISPF Interactive System Productivity Facility (ISPF). An IBM licensed program that serves as a full-screen editor and dialog manager. Used for writing application programs, it provides a means of generating standard screen panels and interactive dialogs between the application programmer and terminal user. In WebSphere Application Server for z/OS, the WebSphere administrator uses ISPF customization dialogs to configure the WebSphere environment onto which you will install and configure WebSphere Process Server for z/OS.

Although WebSphere Process Server for z/OS itself does not utilize ISPF customization panels, you should understand the role of ISPF customization for the base product and you should be familiar with how base product configuration in general relates to the product install and configuration.

Jacl Java command language. A scripting language for the Java 2 environment that is used to create Web content and to control Java applications.

A product administrator can run post install Jacl scripts from a command line to further configure the environment. These scripts must be run using the WebSphere wsadmin tool (located in the install_root/bin directory).

The types of Jacl scripts and the number of Jacl scripts that a product administrator runs after installing or after configuring the product will depend on environment variables, such as which database will be used (and) how the site intends to use WebSphere Process Server for z/OS.

Installing the product

Installing the product refers to the multiphase process of unloading the contents of the product install media onto the z/OS system and then running the install script to install the WebSphere product definitions that enable the product for use.

At sites that install and run products on z/OS, the responsibility for unloading the contents of install media is usually assigned to a system programmer. In most cases the system programmer uses SMP/E to unload the contents of the install media onto the system.

After successfully loading the contents of the install media onto the system, the system programmer can continue the install process by running the product install script, or the system programmer can inform a product administrator that the product code has been loaded successfully and the product administrator can run the install script.

Configuring the product

On z/OS, *configuring the product* refers to those activities that are performed by a product administrator *after the product has been loaded onto the system and after the install script has been run*. Configuring the product involves the following tasks:

- Augmenting the default profile.

  To augment the default profile, the product administrator can run one of two configuration scripts from the command line as follows:

- Run **zWPSConfig.sh** to augment the default profile with configuration data consisting of all the features and functions of WebSphere Process Server for z/OS.
- Run **zWESBConfig.sh** to augment the default profile with configuration data consisting of the enterprise service bus subset functionality of WebSphere Process Server for z/OS.

  Note: When you license WebSphere Process Server for z/OS, and configure it as an ESB-only server, you always have the option of re-configuring it for full WebSphere Process Server functionality at some time in the future.

- Updating the administrative console using command line scripting implemented as jacl scripts.

Profile

Data that describes the characteristics of a user, group, program, device, or remote location.

In WebSphere Application Server for z/OS, a profile represents the collection of user data, along with the shared product binaries that define the WebSphere run-time environment. On z/OS, there is always one and only one profile and that profile is named **default** in each of the configurations.

When you run the WebSphere Process Server for z/OS configuration script you augment the WebSphere Application Server for z/OS default profile with WebSphere Process Server for z/OS configuration data. The action of augmenting the default profile is known as *profile augmentation*. Profile augmentation actions are Ant scripts that get called with the properties in the response file (the path to the response file is part of the product configuration command). These actions apply a changed template to the existing WebSphere Application Server for z/OS profile.

response file

An EBCDIC file that can be customized with setup and configuration data.

When you run the product configuration script from the command line, the command syntax includes a directory path reference to the response file, which automates the profile augmentation process.

A response file cannot be used interactively. The response files are associated with specific configurations supported by the product. When you run the configuration script, you specify a response file that is associated with the WebSphere Application Server for z/OS configuration into which you are installing the product.

shell script

A file of shell commands. If the file is executable, users can run it by specifying the file's name as a shell command or as an operand on an **sh** shell command or on the TSO/E OMVS command. A shell script is similar to a TSO/E REXX exec.

The product installer includes a shell script that installs and enables the product for use, as well as shell scripts that configure the product by augmenting the default profile with product configuration data and definitions.

key word

One of the predefined words of a programming language, artificial language, application, or command.

The shell scripts that you run to install and configure the product, contain key words.

In the shell script that you use to install the product, the key words are preceded by a dash (-) and include the following:

- -smproot
- -runtime
- -install
- -prereqonly

  To check for installation prerequisites.
- -trace

  Optionally, to record trace information.

In the shell script that you use to configure the product, the key words are preceded by a dash (-) and include the following:

- -augment
- -response
- -Z

  Optionally, to override values in the response file.
- -trace

  Optionally, to record trace information.

In the shell script that you use to uninstall the product, the key words are preceded by a dash (-) and include the following:

- -uninstall
- -response
- -Z

  Optionally, to override values in the response file.
- -trace

  Optionally, to record trace information.

Some of the key words require qualifiers in the form of *parameters* that provide specific information with regard to enabling the product for use. For detailed descriptions of the key words and key word parameters supported by the install script, see About the install script. For detailed descriptions of the key words and key word parameters supported by the configuration script, see About the configuration script.

key word parameter
A parameter that consists of a keyword followed by one or more values.

The shell scripts that you run to install and configure the product, contain key words followed by one or more values.

symbolic link
Also referred to as *symlink*, a symbolic link is a type of file that contains the path name of and acts as a pointer to another file or directory.

After unloading the contents of the install media by using SMP/E, you run the install script named **zSMPInstall.sh**. As a result of running the install script you create the symbolic links to the read-only HFS.

z/OS UNIX System Services (USS)
An element of z/OS that creates a UNIX environment that conforms to the

XPG4 UNIX 1995 specifications and provides the two open systems interfaces on the z/OS operating system:

- an application program interface (API)
- an interactive shell interface

Previously, UNIX System Services was a component of OS/390, formerly called OpenEdition.

For the product install process, z/OS UNIX System Services provides the shell or command interface from which you run the install and configuration scripts.

Reading the syntax diagrams

In this documentation, the commands for the WebSphere Process Server for z/OS install script and configuration script are documented in syntax diagrams.

A syntax diagram is a visual representation for a command that indicates how to enter the command on the command line. The graphical nature of a syntax diagram allows you to understand the connectivity of concepts through the structure of the command grammar.

Note: The syntax diagrams in this documentation are examples only. Although the syntax diagrams are descriptive of the commands that you run to install and configure WebSphere Process Server for z/OS, some of the values that you will enter on the command line are variable.

Getting started with syntax diagrams

To read a syntax diagram, follow the path of the line. Read from left to right and top to bottom.

- The `>>-` symbol indicates the beginning of a syntax diagram.
- The `--->` symbol at the end of a line, indicates that the syntax diagram continues on the next line.
- The `>--` symbol at the beginning of a line indicates that a syntax diagram continues from the previous line.
- The `---><` symbol indicates the end of a syntax diagram.

Syntax items, such as a keyword or a variable, can be:

- On the line (required element)
- Above the line (default element)
- Below the line (optional element).

Table 1. Description of syntax diagram items

Syntax Diagram Description	Example
keywords: Keywords are what drive the actions of the command line. Keywords are preceded by a dash (-) and are followed by a space and either another keyword, or a keyword parameter.	`>>-keyword1 -keyword2------------><` `>>-keyword *keyword parameter*--><`

Table 1. Description of syntax diagram items (continued)

Syntax Diagram Description	Example
Symbols: Enter these symbols exactly as they appear in the syntax diagram.	`- Hyphen`

Required Choices:
 	
When two or more items are in a stack and one of them is on the line, you must select one item.
 	
In this example, you must select either A, B, or C.
	`>>-+-A-+-------------------------------><`
	`  +-B-+`
	`  '-C-'`

Variables: Italicized lowercase items (*var_name*) indicate variables. In this example, you can specify a var_name when you enter the keyword command.	`>>-keyword--var_name----------------><`

Examples of command syntax

The following are examples of WebSphere Process Server for z/OS install and configuration commands that adhere to the syntax diagram guidelines described in the previous section.

Installing the product

```
zSMPInstall.sh -smproot /usr/lpp/zWPS/V6R0 -runtime /WebSphere/V6R0M0/AppServer
-install
```

In the example above, the shell script command includes the keyword of **-smproot** and its keyword parameter, which specifies the directory of the install root*/usr/lpp/zWPS/V6R0* as well as the **-runtime** keyword and its associated keyword parameter (the root directory of the WebSphere Application Server for z/OS code) configuration root of *WebSphere/V6R0M0/AppServer*. The directory paths are italicized because they are variable.

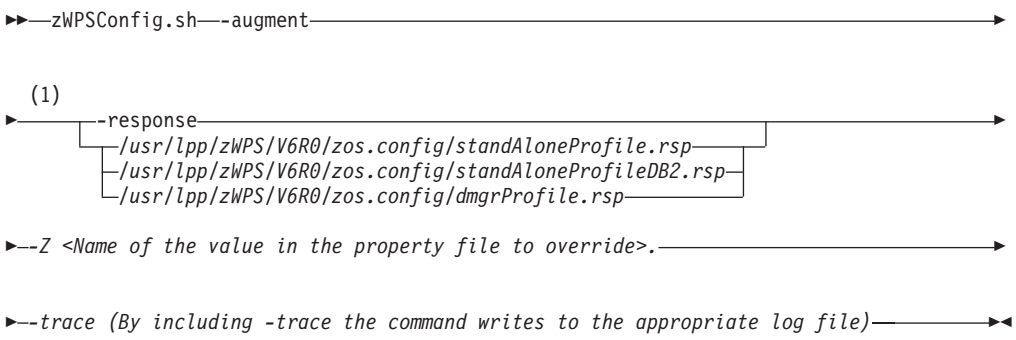
Configure the product for a stand-alone server

```
zWPSConfig.sh -augment -response /usr/lpp/zWPS/V6R0/zos.config/standAloneProfileDB2.rsp
```

In the example above, the configuration command includes the **-augment** keyword to perform profile augmentation actions, which are Ant scripts that get called with the properties in the response file. The **-augment** keyword is followed directly by the **-response** keyword and its associated keyword parameter (the absolute path of the response file). Notice that the response file path is italicized because it is variable.

A syntax diagram showing all of the options for product configuration would display as follows:

Syntax diagram for configuring WebSphere Process Server for z/OS



Notes:

1 Type the absolute path of the response file that is associated with the configuration onto which you are installing WebSphere Process Server for z/OS. This absolute path name in this example assumes the installer is using the default response files. In most cases the user will make a copy of the default response file and edit the values to suit the needs of their environment. So, in most cases the path name would represent the absolute path of a customized response file.

Notice the keyword parameter values for **-response**. They are stacked in the syntax diagram to signify that you have choice with regard which response file to include in your command.

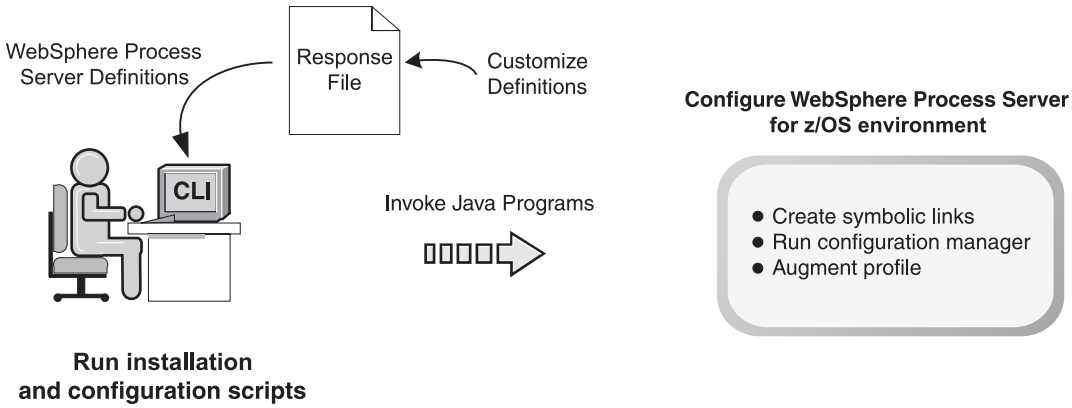
About the command line procedure

You install and configure WebSphere Process Server for z/OS by running shell scripts from the command line.

To perform installation and configuration tasks using scripts, you establish a Telnet session with your z/OS system and then run the scripts. The installation script (**zSMPInstall.sh**) and either the configuration script (**zWPSConfig.sh**) or the configuration script (**zWESBConfig.sh**) use input from a response file to determine how to set up the run-time environment.

Note: Alternatively, you can run the scripts from an OS/390 UNIX command shell by entering the TSO command OMVS at the ISPF Command Shell or TSO OMVS from any other ISPF panel.

The *response file* is an EBCDIC file that can be customized with setup and configuration data. A response file cannot be used interactively, and therefore you must supply the environment definitions you will use before running a script.



When you run a script using the command-line interface (CLI), the script invokes Java programs in the z/OS environment. The programs perform tasks needed to install and configure the run-time environment, including creating symbolic links, running the configuration manager and augmenting the default profile.

Because the scripts rely on WebSphere Process Server definitions that are specified in the response file, the command line procedure is referred to as a *silent* or *unattended* install.

About installing on z/OS

Installing the product on z/OS involves loading the product from the install media onto the z/OS system and running the install script from a command line.

Installing the product on z/OS is a multiphase process that can span multiple roles. For a description of the phases that make up an install, see "Install phases."

You must install and configure WebSphere Application Server for z/OS, prior to installing WebSphere Process Server for z/OS. If you do not have WebSphere Application Server for z/OS installed and configured, it is included as part of the WebSphere Process Server for z/OS package.

Install and configuration scripts

Unlike many products installed on z/OS, WebSphere Process Server for z/OS does not rely on ISPF dialogs to create the installation-specific configuration data. Instead, the product comes with an install script named **zSMPInstall.sh** and two configuration scripts named **zWPSConfig.sh** and **zWESBConfig.sh**. For detailed information on the configuration script, see About the configuration script.

Install script results

As a result of running the install script, all of the components and features are installed onto the system. In this sense, you cannot ″customize″ the product install by picking and choosing which features to install onto your system. However, through keyword and keyword parameters in the configuration command, you can configure the installed product features. For detailed information on the configuration script, see About the configuration script.

Install phases

Installing WebSphere Process Server for z/OS on z/OS is a multiphase process that can span multiple roles.

In general, phase 1 of the install pertains to *unloading the contents of the install media* onto the z/OS system and is the responsibility of a system programmer.

Phase 2 of the install pertains to *running the install script* to create the required WebSphere definitions that enable the product for use. This second phase is the responsibility of a product administrator role.

The relationship between roles and install phases is illustrated below.

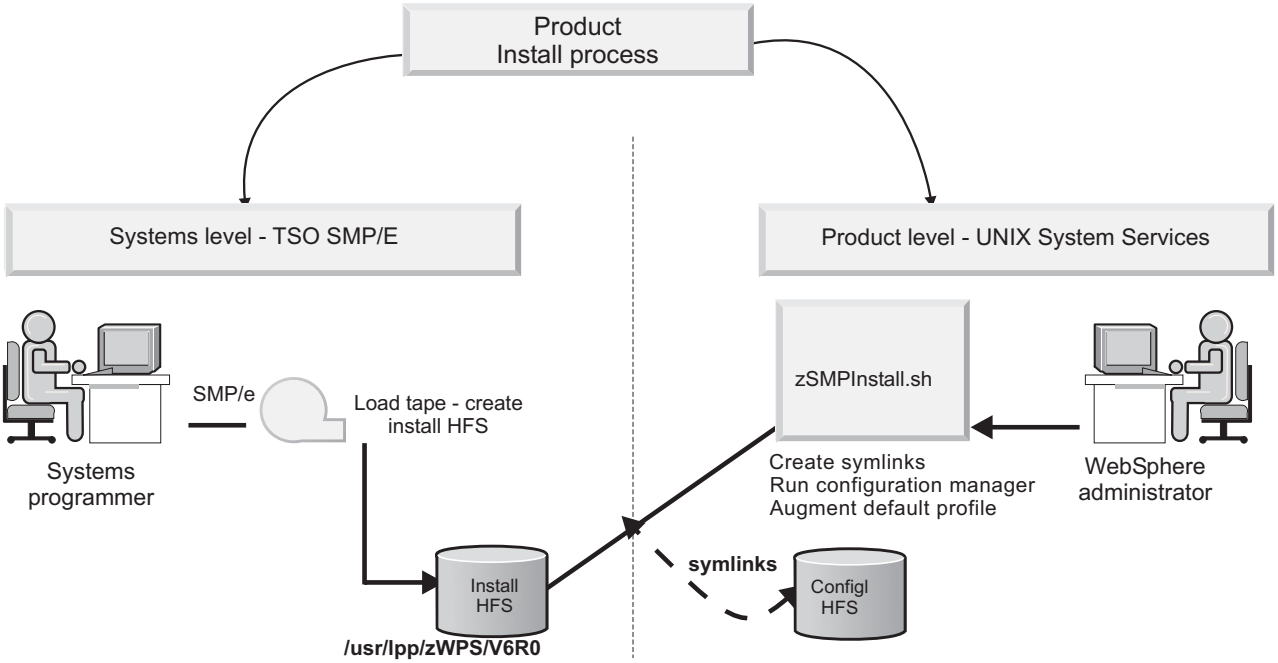


Figure 1. Install process for WebSphere Process Server for z/OS

The product install is not considered to be complete until both phases (loading the product code and running the install script) have been performed successfully.

The following illustration depicts install phases:

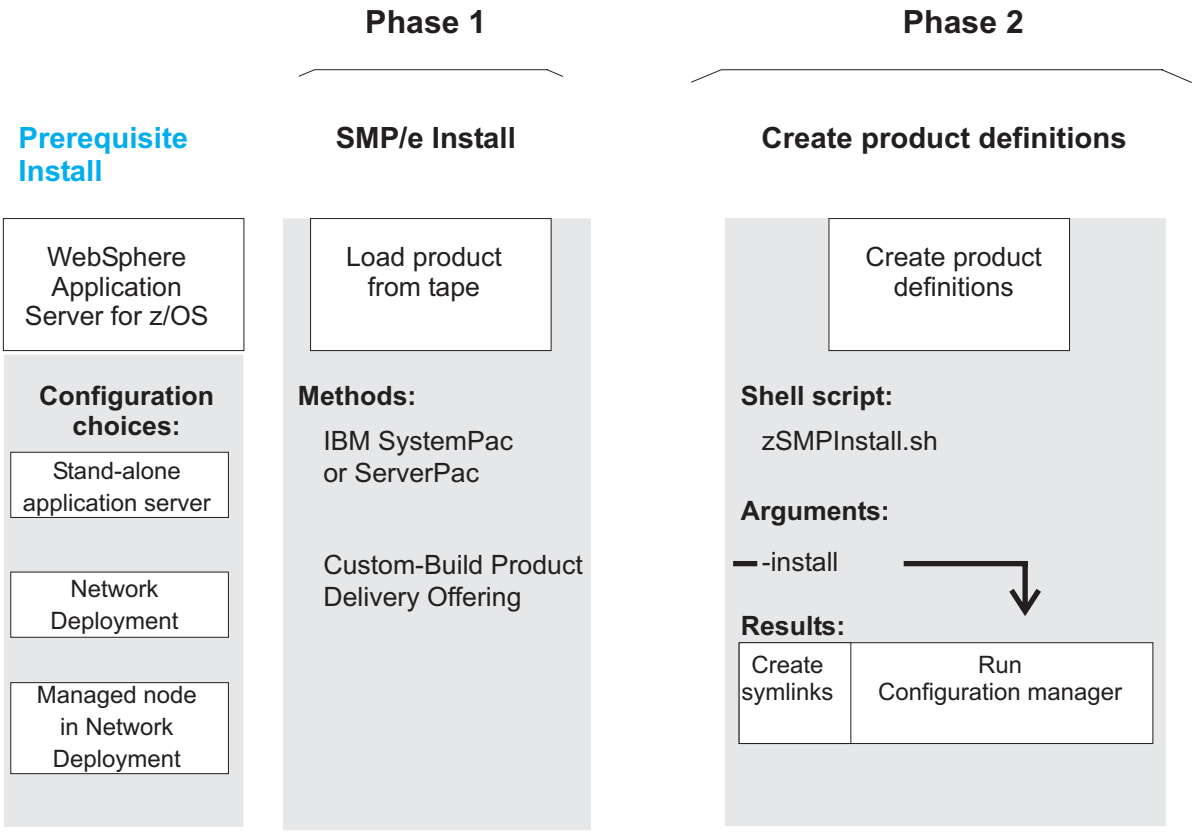


Figure 2. Install phases

Phase 1 - Unloading the product code from the install media

If the product delivery package was an IBM Custom-Built Product Delivery Option (CBPDO), the system programmer uses System Modification Program/Extended (SMP/E) to unload the product code onto the z/OS system. The system programmer relies on the Program Directory to guide them through the process. The Program Directory contains information concerning the material and procedures associated with the installation of the product code. You can download the program directory in PDF format from the WebSphere Process Server for z/OS download page, at http://www-306.ibm.com/software/integration/wps/library/infocenter/.

If the product delivery package was an IBM SystemPac / ServerPac, the system programmer copies SMP/E data sets that correspond to the CustomPac service level onto the z/OS system.

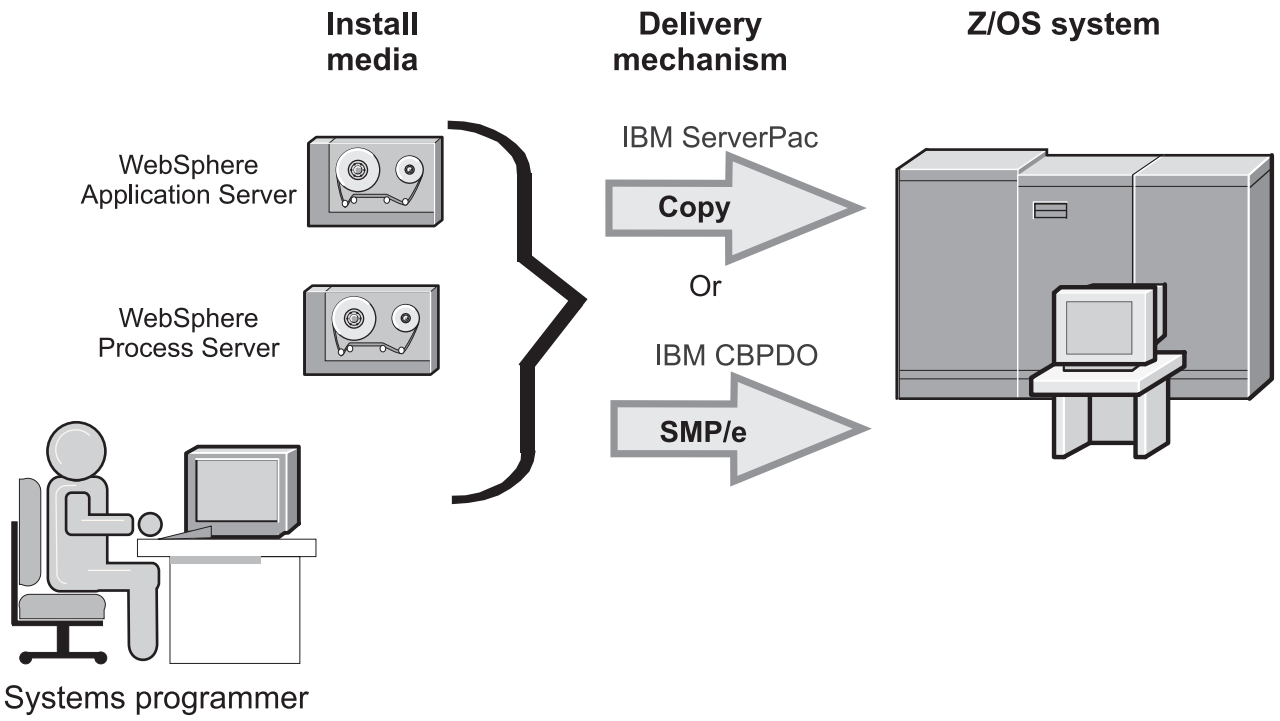


Figure 3. Loading the product code onto z/OS

Note: This phase of the install assumes that the correct version of WebSphere Application Server for z/OS has been installed and customized. The SMP/E install process checks the version level of WebSphere Application Server for z/OS to ensure version compatibility. If the version of WebSphere Application Server for z/OS is not at the level required to support WebSphere Process Server, the system programmer will need to take the appropriate actions to update the version of WebSphere Application Server for z/OS. For information on how to apply service to WebSphere Application Server for z/OS, see Applying product maintenance in the WebSphere Application Server for z/OS information center.

The result of completing the first phase of the install is a read-only HFS.

After successfully unloading the product code from the install media onto the system, the WebSphere administrator implements phase 2 of the installation by running the install script.

The following illustration depicts the methods supported for loading the contents from the install media onto the z/OS system.

About System Modification Program/Extended (SMP/E)

Here are some facts about SMP/E:

- SMP/E is an integral part of the product installation, service, and maintenance process on z/OS.
- SMP/E can be used to install and service any software that is packaged in SMP/E system modification (SYSMOD) format.
- SMP/E can be run either using batch jobs or using dialogs under Interactive System Productivity Facility/Program Development Facility (ISPF/PDF).

- SMP/E dialogs help you interactively query the SMP/E database, as well as create and submit jobs to process SMP/E commands.
- The guidance for doing the SMP/E installation is a program directory.

The system on which SMP/E runs is called the *driving system*.

The product code is unloaded into a directory called *install root*. The install root is read-only and is on its own HFS so it can be mounted and unmounted independently.

Phase 2 - Running the install script

In phase 2 of the install process, the WebSphere administrator creates the required product definitions by running the install script, **zSMPInstall.sh** from a command line.

As a result of completing this phase of the install, the product administrator will have created the symlinks and will have updated the administrative console with WebSphere Process Server for z/OS features.

The install script produces a "configuration root" (sometimes referred to as a *config HFS*) for each server instance.

Note: The configuration root represents the WebSphere Application Server HFS (`default name /WebSphere/V6RxMx/AppServer`), extended with the WebSphere Process Server files and symlinks.

A configuration root holds the data for a single product installation. The configuration root is a writable HFS that holds customized configuration documents and files for the configured product installation.

The configuration root also has symbolic links (sometimes referred to as *symlinks*) to the install root.

The symlinks point to read-only files from the install root, such as JAR files, shell scripts, and so on. SMP/E updates the install root, and the post -SMP/E configuration tasks update the config root on a per-node basis during the start-up of the product.

After completing this phase of the installation successfully, the WebSphere administrator can begin to configure the product for use. For information on the product configuration process, see About product configuration on z/OS.

The figure below shows an installation process that includes loading the contents of the tape via SMP/E and creating the product definitions by running the install script named **zSMPInstall.sh**.

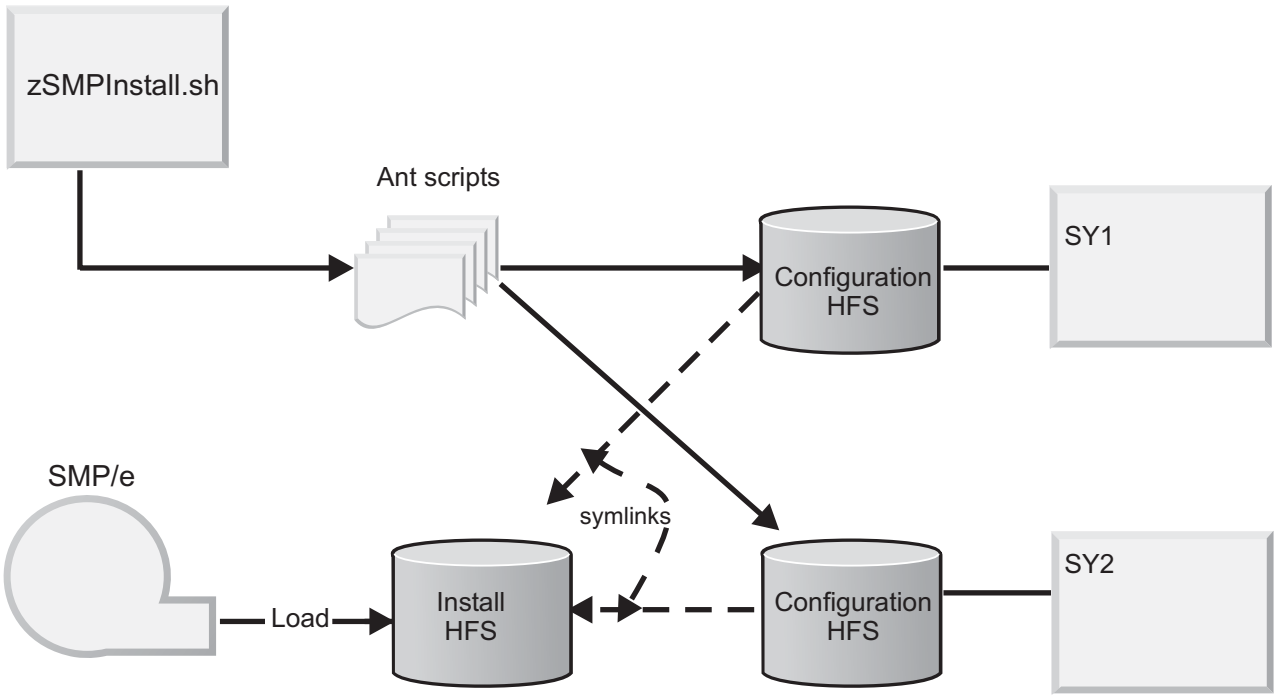


Figure 4. Process flow of install for WebSphere Process Server for z/OS

Notes:

- The read-only shared HFS is shown as shared, however, the system programmer who executes the installation can decide to share the disks or create copies of them. This decision has no impact on the post-installation configuration steps.
- The HFS is installed in the install root, which means that the unpacking steps are controlled by SMP/E and the service of the code is also controlled by SMP/E.

About the install script

When run, the install script creates the WebSphere Process Server for z/OS definitions that enable the product for use.

The commands described in this topic are contained in, and should be run from the SMP/E install HFS. The default install HFS directory for the product is as follows:/usr/lpp/zWPS/V6R0/zos.config/bin.

An administrator can run the install script **(zSMPInstall.sh)** with different command arguments.

The script invokes a series of actions against the product code that was installed via SMP/E.

Running **zSMPInstall.sh** creates symbolic links and updates the administrative console with WebSphere Process Server for z/OS product definitions.

You run the uninstall command from the same script that you use to install WebSphere Process Server for z/OS. For details on the **-uninstall** command, see About the uninstall command.

Command line structure

The following illustration demonstrates the command line structure of **zSMPInstall.sh** by labeling the various portions of the command line and by providing an example of the key words and keyword parameters.

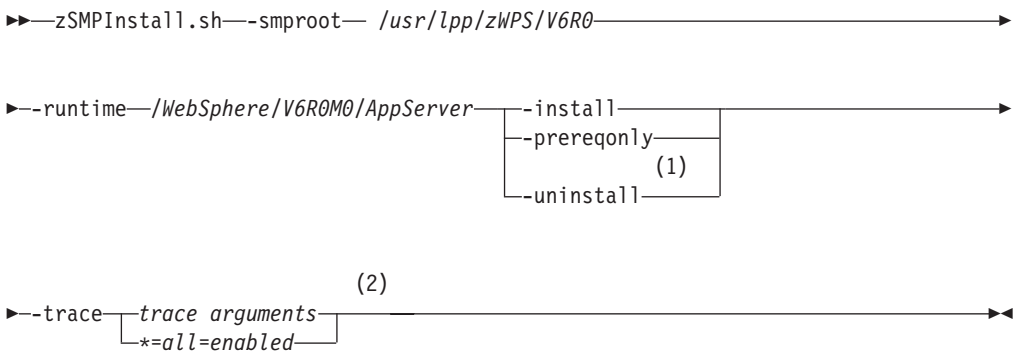
Script name	Install HFS path	Application Server configuration root	Procedure	Trace option
zSMPInstall.sh	-smproot <install HFS>	-runtime <base install HFS>	-install -prereqonly -uninstall	-trace

Command syntax

The following syntax diagram illustrates all of the allowed syntax for the install command.

Note: For information on how to read syntax diagrams, see Reading the syntax diagrams.

Install script syntax diagram



Notes:

1 These arguments determine the product install actions to be performed by **zSMPInstall.sh**. Choose one of these arguments to control how the product install proceeds. The **-install** argument implements actions that update the administrative console with WebSphere Process Server for z/OS product definitions. For information on how to use the **-uninstall** command, see About the uninstall command.

2 By including **-trace** the command writes to the appropriate trace file.

Install script command line arguments and parameters

The following table describes the install script command arguments. For a more detailed description of the various actions that result from running the zSMPInstall.sh , see the **Install actions** section that follows this table.

Table 2. Command line arguments for the install script

Description	Implementation method	Result
Create the WebSphere Process Server definitions and update the administrative console.	Run **zSMPInstall.sh** from command line, incorporating the sequence of command arguments necessary to achieve the desired configuration. You can run **zSMPInstall.sh** with the following command arguments and parameters: • **-smproot** This keyword and its associated keyword parameter are required. The parameter value is the Hierarchical File System (HFS) directory that resulted from the SMP/E install. The standard path to this directory is:*/usr/lpp/zWPS/V6R0* • **-runtime** This keyword and its associated keyword parameter are required. The parameter value is the configuration root of WebSphere Application Server for z/OS. The standard path to this directory */WebSphere/V6R0M0/AppServer*. The following keywords are mutually exclusive. Specify one, and only one in the script command: • **-prereqonly** • **-install** • **-uninstall** The following keyword is optional. • **-trace** If you incorporate this argument into the script command, you must enter a parameter as qualifier to this argument.	Extends WebSphere Process Server features into the WebSphere Process Server for z/OS product directory. Results will vary based on the arguments and parameters incorporated into the command. **-prereqonly** verifies arguments and environment. Results are: • Validates prerequisites • Verifies arguments **-install** results are: • Validates prerequisites (see **-prereqonly**) • Creates *symbolic links* from the WebSphere Process Server read-only HFS directories to the WebSphere Application Server for z/OS **/lib** and **/bin** directories, installing the WebSphere Process Server definitions. • Creates post install file • Updates code base permissions • Enables WebSphere Process Server features by running Configuration Manager scripted actions. This will create any new administrative console plugin extensions. These resources include the following: – Business Process Choreographer – WebSphere Process Server core resources – Common Event Infrastructure (CEI) **-uninstall**. For information on how to use the **-uninstall** command, see About the uninstall command. **-trace** results are: If you choose not to use the -trace argument, then "*=all=disabled" is used as the trace specification. Valid -trace parameters are as follows: • *=all=enabled To record trace information

Install actions

Verifying the arguments and the environment

The *Verifying the arguments and the environment*, encompasses the following actions related to installing the product and enabling the product for use:

1. Verify that the **-smproot** argument and it's parameter value are specified and that the smproot directory specified in the value exists.
2. Verify that the **-runtime** argument and it's parameter value are specified and the run-time directory specified in the value exists.
3. Verify that one and only one of the product install arguments (-prereqonly, -install or -uninstall) is specified.
4. If the -trace keyword is specified, verify a value is also specified.

   If the -trace keyword is not specified, set to default "*=all=disabled".
5. Verify that any unrecognized keywords are not specified.
6. Create the trace file in the run-time directory if it does not exist.

   Default file name is /WebSphere/V6R0M0/AppServer/logs/wbi/zSMPInstall.trace.
7. Create the log file in the run-time directory.

   Default file name is /WebSphere/V6R0M0/AppServer/logs/wbi/zSMPInstall.log
8. Verify that the prerequisite WebSphere Application Server for z/OS is at required level.

Create symbolic links

The *Create symbolic links*, encompasses the following actions related to installing product and enabling the product for use:

1. Create a symbolic link file in the run-time directory for every file in the install SMP/E read-only HFS directory.

   For Example: A file named /WebSphere/V6R0M0/AppServer/profileTemplates/default.bfm/actionRegistry.xml (of type "Syml") is created that is a symbolic link to the SMP/E HFS file named /usr/lpp/zWPS/V6R0/profileTemplates/default.bfm/actionRegistry.xml .
2. Create a file in WAS_HOME/bin/zWESBConfig.sh that is a symbolic link to SMP/E HFS file named /usr/lpp/zWPS/V6R0/zos.config/bin/zWESBConfig.sh
3. Create file in WAS_HOME/bin/zWPSConfig.sh that is a symbolic line to SMP/E HFS file named /usr/lpp/zWPS/V6R0/zos.config/bin/zWPSConfig.sh

Create post installer file

The *Create post installer file*, encompasses the following actions to installing product and enabling the product for use:

1. Create the service product directory, if it does not already exits.

   Default name is /WebSphere/V6R0M0/AppServer/properties/service/product/WBI
2. Create the service product backup directory, if it does not already exits.

Default name is/WebSphere/V6R0M0/AppServer/properties/service/product/WBI/backup

3. Create the service log directory, if it does not already exits.

   Default name is /WebSphere/V6R0M0/AppServer/properties/service/logs/WBI

4. Copy service level properties from the install SMP/E read-only HFS directory - default name is /usr/lpp/zWPS/V6R0/productversion/properties/service/product/WBI/service/service-level.properties, to the service product directory - default name is: /WebSphere/V6R0M0/AppServer/properties/service/product/WBI/service-level.properties

5. Create a symbolic link file in the run-time directory - default name is /WebSphere/V6R0M0/AppServer/properties/service/product/WBI/service, to the read-only HFS directory service directory - default name is /usr/lpp/zWPS/V6R0/properties/service/product/WPS/service/.

6. Create a Installer properties file in the run-time directory - default name is /WebSphere /V6R0M0/AppServer/properties/service/product/WBI/zWPSPostInstaller.properties.

   The **zWPSPostInstaller.properties** file written contains the following lines.

   Note: Some lines, such as WBI_TIME.created and WBI_BUILD_LEVEL.current_level, will contain variable data.

```
#--------------------------------------------------------------------------------
# WPS Post Installer Properties File
#
# WebSphere Process Server 6.0
#
# Licensed Materials - Property of IBM
# 5655-N53
# (C) Copyright IBM Corporation 2006. All Rights Reserved.
# US Government Users Restricted Rights - Use, duplication or disclosure
# restricted by GSA ADP Schedule Contract with IBM Corp.
#
#--------------------------------------------------------------------------------
WBI_TIME.created=Apr 5, 2006 3:53:47 PM EDT
WBI_TIME.modified=
WBI_PATH.smpe_root=/usr/lpp/wbi/zWPS/V6R0
WBI_PATH.config_root=/WebSphere/V6R0M0/AppServer
WBI_PATH.service_root=/WebSphere/V6R0M0/AppServer/properties/service/product/WBI
WBI_PATH.backup_root=/WebSphere/V6R0M0/AppServer/properties/service/product/WBI/backup
WBI_BUILD_LEVEL.target_rollback_level=
WBI_BUILD_LEVEL.initial_install=o0611.10
WBI_BUILD_LEVEL.current_level=o0611.10
WBI_BUILD_LEVEL.last_component_install=o0611.10
```

Update code base permissions

The *Update code base permissions*, encompasses the following actions related to installing product and enabling the product for use:

1. Adds the SMP/E read-only HFS directory to the JVM options by inserting the following string: -Dwps.smpe.install.root=WPS SMP/E HFS in value clause for symbolic name WAS_SERVER_ONLY_default_jvm_options in the variables.xml file. Default location for variables.xml file is: /WebSphere/V6R0M0/AppServer/profiles/default/config/cells/SY1/ nodes/SY1/variables.xml

   Example:

   ```
   <entries xmi:id="VariableSubstitutionEntry_1122997753573"
     symbolicName="WAS_SERVER_ONLY_default_jvm_options"
     value="-Dwps.smpe.install.root=/usr/lpp/zWPS/V6R0  ..."
   ```

2. Saves the original **variables.xml** file (i.e. without the above change) to file **variables.xml.wps** in the same directory.

3. Updates the **server.policy** by inserting the following lines in the **server.policy** file. Default server.policy location is: /WebSphere/V6R0M0/AppServer/profiles/default/properties/server.policy.

```
// Added the following for WPS for z/OS
   grant codeBase "file:${wps.smpe.install.root}/-" {
      permission java.security.AllPermission;
    };
```

4. Saves the original server.policy file (i.e. without the above change) to file server.policy.wps in the same directory.

Update Configuration Manager

The *Update Configuration Manager*, encompasses the following actions related to installing product and enabling the product for use:

1. Rebuilds the administrative console on the WebSphere Config HFS by invoking the Config Manager with the 6.0 full install directory. The full install directory (a symbolic link to the WAS SMP/E HFS) contains a number of Ant files to be executed. Default name for the install directory is: /WebSphere/V6R0M0/AppServer/properties/version/install.wbi/6.0.0.0/config/full/install

   Ant scripts to be run include the following:

   ```
   90SConfigNoProfileFirstStepsESB.ant
   90SConfigNoProfileFirstStepsWBI.ant
   90SConfigureWSProfileForWBI.ant
   90SConfigWBIMigrationScript.ant
   90SInstallCEI.ant
   90SUpdateJavaOptions.ant
   91SConfigNoProfileFirstStepsCharset.ant
   98SDeployBPCAdminConsolePlugins.ant
   98SDeployServerAdminConsolePlugins.ant
   99SDeployCoreAdminConsolePlugins.ant
   ```

About the uninstall command

You run the uninstall command from the same script that you use to install WebSphere Process Server for z/OS.

The uninstall actions are implemented by running zSMPInstall.sh with the **-uninstall** keyword.

Uninstall command line arguments and parameters

The following table describes the command arguments used to uninstall WebSphere Process Server for z/OS.

Table 3. Command line arguments for un-installing WebSphere Process Server for z/OS.

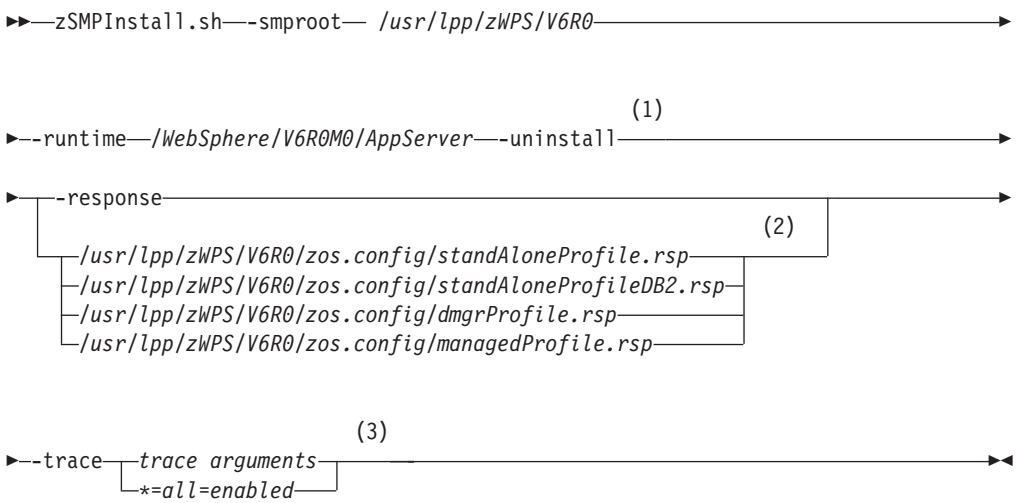
Description	Implementation method	Result
Remove the product definitions from the administrative console and un-augment the default profile.	Run **zSMPInstall.sh** from command line, incorporating the sequence of command arguments necessary to uninstall the product. To uninstall WebSphere Process Server for z/OS, run **zSMPInstall.sh** with the following command arguments and parameters: • **-smproot** This keyword and its associated keyword parameter are required. The parameter value is the Hierarchical File System (HFS) directory that resulted from the SMP/E install. The standard path to this directory is:*/usr/lpp/zWPS/V6R0* • **-runtime** This keyword and its associated keyword parameter are required. The parameter value is the configuration root of WebSphere Application Server for z/OS. The standard path to this directory */WebSphere/V6R0M0/AppServer*. • **-uninstall** The uninstall keyword used to drive the uninstall actions. • **-response** Because the uninstall actions include un-augmenting the default profile, you must include the -response keyword. • **-trace** This is an optional argument. If you incorporate this argument into the script command, you must enter a parameter as qualifier to this argument.	Removes the product features from the WebSphere Process Server for z/OS product directory. **-uninstall** results are: • Disables WebSphere Process Server features by running Configuration Manager scripted actions. This will remove any administrative console plug-in extensions. • Remove profile augmentation using WSPROFILE scripted actions. This will un-augment the default profile. • delete post install file • remove code base permissions **-trace** results are: If you choose not to use the -trace argument, then "*=all=disabled" is used as the trace specification. Valid -trace parameters are as follows: • *=all=enabled To record trace information on all trace-enabled Java classes • Installer=all=enabled To record trace information on only the Installer group of Java classes

Command syntax

The following syntax diagram illustrates the allowed syntax for the uninstall command.

Note: For information on how to read syntax diagrams, see Reading the syntax diagrams.

Uninstall script syntax diagram



Notes:

1 The **-uninstall** argument implements actions that remove the product definitions from the administrative console and un-augment the default profile.

2 Type the absolute path of the response file that is associated with the profile that you are un-augmenting.

3 By including **-trace** the command writes to the appropriate log file.

Uninstall actions

Revert Configuration Manager

The **Revert Configuration Manager**, encompasses the following actions related to un-installing the product and un-augmenting the default profile:

1. Reverts the adminconsole program to pre-install state by invoking the Config Manager with the 6.0 full uninstall directory.

   The full uninstall directory (a symbolic link to the WAS SMP/E HFS) contains a number of Ant files to be executed.

   Default name for the install directory is: /WebSphere/V6R0M0/AppServer/properties/version/install.wbi/6.0.0.0/config/full/uninstall

   Ant scripts to be run include the following:

   ```
   90SDeleteFirstStepsFilesESB.ant
   90SDeleteFirstStepsFilesWBI.ant
   90SRemoveJavaOptions.ant
   ```

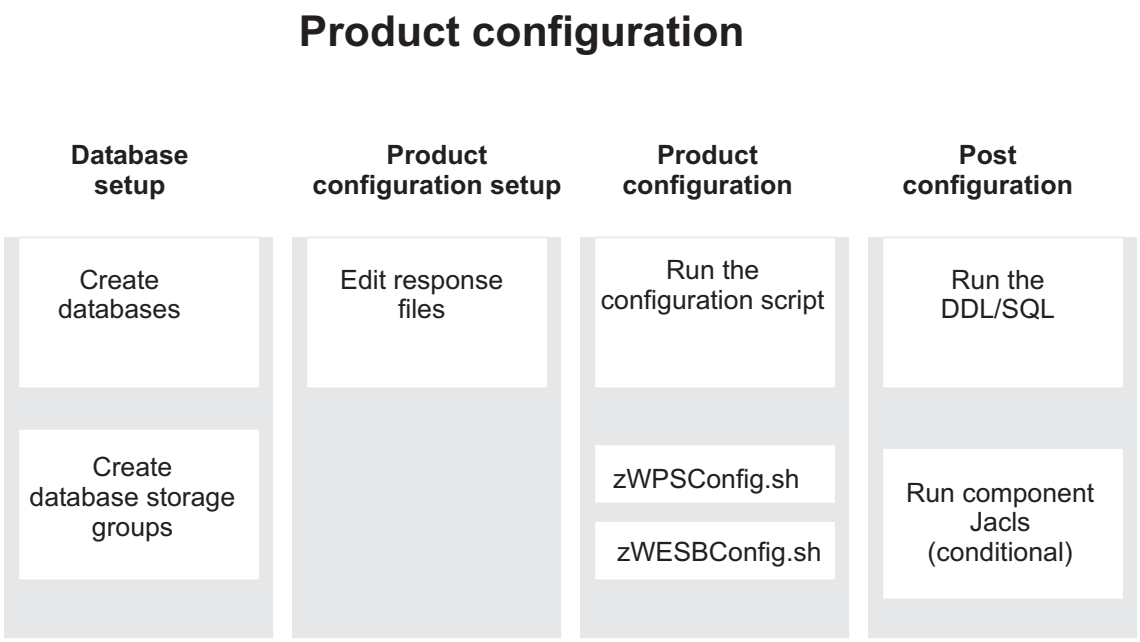
```
90SUninstallCEI.ant
98SUndeployBPCAdminConsolePlugins.ant
98SUndeployServerAdminConsolePlugins.ant
99SUndeployCoreAdminConsolePlugins.ant
```

About product configuration on z/OS

Configuring WebSphere Process Server for z/OS involves running the configuration script from a command line to enable profile augmentation using scripted actions.

Before you configure WebSphere Process Server for z/OS, you need to create the appropriate database and storage groups for some database types. For information on how to create the databases and storage groups, see Planning your configuration.

Also, depending on your environment, you may be required to perform additional tasks after you have run the configuration script. For information on these tasks, see Post configuration.



The product configuration script is run from a command line.

There are several configuration options, which on z/OS are driven by the WebSphere Process Server for z/OS configuration script's use of response files.

The content in the response file is used to augment the WebSphere Application Server for z/OS profile with WebSphere Process Server for z/OS configuration data.

Profile augmentation

Profile augmentation is the process by which a product administrator adds WebSphere Process Server for z/OS configuration data, (such as the data that describes the characteristics of a user, group, resource, program, device, or remote location) to the WebSphere Application Server for z/OS default profile.

Note: On z/OS, there is always one and only one profile and that profile is named **default** in each of the configurations.

The objective of profile augmentation is to extend the capabilities of your WebSphere Application Server for z/OS product configuration by adding to it the features and functions of WebSphere Process Server for z/OS.

Before running the scripts that perform profile augmentation actions, a product administrator must edit the response file that is specific to the WebSphere Process Server for z/OS configuration. The augment actions are Ant scripts that get called with the properties in the response files.

To configure WebSphere Process Server for z/OS the product administrator runs one of the following configuration scripts:

- zWPSConfig.sh
- zWESBConfig.sh

The two configuration scripts allow for flexibility when configuring WebSphere Process Server for z/OS. Run the script **zWESBConfig.sh** if your process server will run Mediations only. A configuration that runs Mediations only, reduces the footprint of the server by excluding unused features and enabling the server to start quicker and to consume less memory. Run **zWPSConfig.sh** if your process server will be used for business processes. The script **zWPSConfig.sh** is required if the application servers are configured for processes.

The manner by which the product administrator performs profile augmentation varies with the WebSphere Application Server for z/OS configuration type. The directory structure (which will vary by product configuration) is passed into the configuration manager portion of the installer. The configuration manager runs the Ant scripts in the /actions subdirectory, which results in the administrative console being updated with WebSphere Process Server definitions.

About the configuration script

Use the configuration script to augment the default profile.

WebSphere Process Server comes with two configuration scripts named **zWPSConfig.sh** and **zWESBConfig.sh** respectively.

The commands described in this topic are contained in, and should be run from the /WebSphere/V6R0M0/AppServer/bin directory.

The two configuration scripts allow for flexibility when configuring WebSphere Process Server for z/OS. The script **zWESBConfig.sh** allows the product administrator to configure the server as an "ESB-only" configuration, reducing the footprint of the server by excluding unused features and enabling the server to start quicker and to consume less memory. The script **zWPSConfig.sh** configures your server as a process server, enabling the server to handle processes. The decision on which script to run is based on business objectives and server performance considerations. For information on enterprise service bus functionality, see *Overview of enterprise service bus* in the Product overview. For information on WebSphere Process Server functionality, see *Architectural overview of WebSphere Process Server* in the Product overview.

Depending on how you will use WebSphere Process Server for z/OS, you run one of the two configuration scripts, **zWPSConfig.sh** or **zWESBConfig.sh** from command line prompt. The scripts support the same command arguments.

As a result of running either of the configuration scripts, the default profile is updated with WebSphere Process Server for z/OS configuration data.

If you choose to create an ESB-only configuration, you have the capability later on to change to a Process Server configuration. For more information, see About the reconfiguration process.

The command for running the configuration script includes the path to the response file. The property values that you set in the response file are used to create your product configuration. The choices you make when setting certain parameters in the response file may result in the need to perform additional post configuration tasks. For information on which parameter settings will result in having to perform post configuration tasks, see Working with response files.

Command line structure

The following illustration demonstrates the command line structure of the configuration scripts by labeling the various portions of the command line and by providing an example of the command arguments.

Command line structure
Augmenting the default profile

Script name	Procedure	Configuration Response file	Property overrides	Trace option
zWPSConfig.sh zWESBConfig.sh	-augment	-response <response file path>	-Z <property name>	-trace

Command syntax

The following syntax diagram illustrates the syntax of the configuration script.

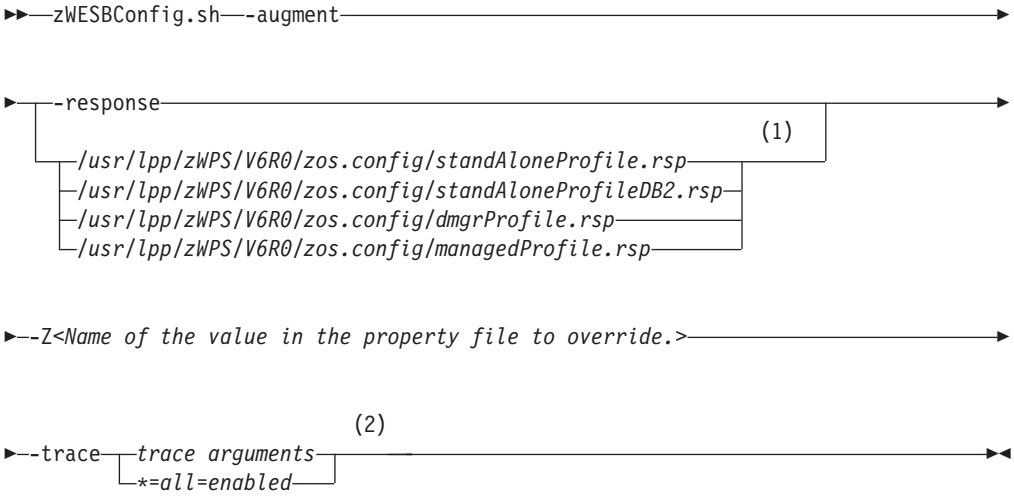
Syntax diagram for configuring WebSphere Process Server for z/OS

Notes:

1 Type the absolute path of the response file that is associated with the configuration onto which you are installing WebSphere Process Server for z/OS. This example shows the default path to the response files packaged with the product. When you run the command, the **-response** parameter will represent the absolute path to the response file that you have modified with the values specific to your environment. The information required to create the DB2 for z/OS database in the response file named standAloneProfileDB2.rsp contains properties that can be set to support DB2 for z/OS version 7 or DB2 for z/OS version 8. The default values are set for DB2 for z/OS version 8.

2 By including **-trace** the command writes to the appropriate trace file.

Syntax diagram for configuring WebSphere Process Server for z/OS as an ESB-only server



Notes:

1 Type the absolute path of the response file that is associated with the configuration onto which you are installing WebSphere Process Server for z/OS. This example shows the default path to the response files packaged with the product. When you run the command, the **-response** keyword and its parameter represent the absolute path to the response file that you have modified with the values specific to your environment. The information required to create the DB2 for z/OS database in the response file named standAloneProfileDB2.rsp contains properties that can be set to support DB2 for z/OS version 7 or DB2 for z/OS version 8. The default values are set for DB2 for z/OS version 8.

2 By including **-trace** the command writes to the appropriate trace file.

Command line arguments and parameters

The following table describes the product configuration command arguments and the result of running the script with the command arguments.

Table 4. Configuring WebSphere Process Server for z/OS - Running the server as a Process Server

Description	Implementation method	Result
Augment the default profile with WebSphere Process Server for z/OS configuration data.	Run **zWPSConfig.sh** from command line, incorporating the sequence of keywords and keyword parameters to achieve the desired configuration. You can run **zWPSConfig.sh** with the following keyword and keyword parameters: **Note:** The actions **–prereqonly** and **–augment** are mutually exclusive (you run one or the other – not both!) • -prereqonly • -augment • -response You must enter a keyword parameter as qualifier to the -response keyword. • **-Z** This is an optional keyword. If you incorporate this keyword into the script command, you must enter a parameter as qualifier to this argument. • -trace You must enter a keyword parameter as qualifier to the -trace keyword. The default **-trace** specification is ″*=all=disabled″. To record trace information on all trace-enabled Java classes, specify ″*=all=enabled″. To record trace information on only the Installer group of Java classes, specify ″Installer=all=enabled″.	Results, based on the keywords and keyword parameters used when running **zWPSConfig.sh** are as follows: • **-prereqonly**, results are: Validates that **zSMPInstall.sh** has successfully created the product definitions for WebSphere Process Server. • **-augment** results are: – Validates prerequisites – Verifies arguments – Enable profile augmentation using scripted actions. Augments the WebSphere Application Server for z/OS default profile with WebSphere Process Server for z/OS configuration data, as specified by property values in the response file. • **-response <absolute path of response file>** results are: – The response file containing the properties for configuring WebSphere Process Server for z/OS. – Response files will be provided for the following WebSphere Application Server for z/OS configurations: - Stand-alone using Cloudscape - Stand-alone using DB2 for z/OS version 7 - Stand-alone using DB2 for z/OS version 8 - Managed node - Deployment Manager **Note:** There is a single DB2 for z/OS response file for the stand-alone configuration. This response file contains values for DB2 for z/OS version 7 and DB2 for z/OS version 8. The default values for DB2 for z/OS-related properties are set to DB2 for z/OS version 8. • **-Z**<propertyOverride> results are:

Table 5. Configuring WebSphere Process Server for z/OS - Running the server as an ESB-only server

Phase	Implementation method	Result
Augment the default profile with configuration data for enterprise service bus functionality.	Run **zWESBConfig.sh** from command line, incorporating the sequence of command keywords and keyword parameters necessary to achieve the desired configuration. You can run **zWESBConfig.sh** with the following keywords: **Note:** The actions **–prereqonly** and **–augment** are mutually exclusive (you run one or the other – not both!) • -prereqonly • -augment • -response You must enter a parameter as qualifier to this argument. • -Z You must enter a parameter as qualifier to this argument. • -trace You must enter a parameter as qualifier to this argument. The default **-trace** specification is ″*=all=disabled″. To record trace information on all trace-enabled Java classes, specify ″*=all=enabled″. To record trace information on only the Installer group of Java classes, specify ″Installer=all=enabled″.	Results, based on the keywords used when running **zWESBConfig.sh** are as follows: • -**prereqonly**, results are: Validates that **zSMPInstall.sh** has successfully created the product definitions for WebSphere Process Server. • **-augment** results are: – Validates prerequisites – Verifies arguments – Enable profile augmentation using scripted actions. Augments the WebSphere Application Server for z/OS default profile with the configuration data for WebSphere Process Server for z/OS configured as an ESB-only server. • **-response** <absolute path of response file> results are: – The response file containing the properties for configuring WebSphere Process Server for z/OS. – Response files will be provided for the following WebSphere Application Server for z/OS configurations: - Stand-alone using Cloudscape - Stand-alone using DB2 for z/OS version 7 - Stand-alone using DB2 for z/OS version 8 - Managed node - Deployment Manager **Note:** There is a single DB2 for z/OS response file for the stand-alone configuration. This response file contains values for DB2 for z/OS version 7 and DB2 for z/OS version 8. The default values for DB2 for z/OS-related properties are set to DB2 for z/OS version 8. • **-Z**<propertyOverride> results are: – This option will allow the user to override any individual property specified in the provided

Augmentation actions

Verify arguments and environment

The *Verifying the arguments and the environment*, encompasses the following augmentation actions:

1. Verify that one and only one of the argumentation procedure arguments (-prereqonly or -augment) is specified.
2. Verify that the -response argument and it's parameter value are specified and that the response file that is specified in the value exists.
3. Verify that any **-Z** arguments contain an equal sign (=).
4. If the -trace argument is specified, verify a value is also specified.

   If the -trace argument is not specified, set to default ″*=all=disabled″.
5. Verify that any unrecognized keywords are not specified.
6. Create the trace file in the run-time directory if it does not exist.

   For a WebSphere Process Server configuration, the default file name is /WebSphere/V6R0M0/AppServer/logs/wbi/zWPSConfig.trace

   For an ESB-only server configuration, the default file name is /WebSphere/V6R0M0/AppServer/logs/wbi/zWESBConfig.trace.
7. Create the log file in the run-time directory.

   For a WebSphere Process Server configuration, the default file name is /WebSphere/V6R0M0/AppServer/logs/wbi/zWPSConfig.log

   For an ESB-only server configuration, the default file name is /WebSphere/V6R0M0/AppServer/logs/wbi/zWESBConfig.log.
8. Read the Response File and load it's values to the response properties table.
9. Add or replace any -Z keyword command line values in the response properties table.
10. Verify that the profileName, profilePath, nodeName, cellName, and serverName properties are specified in the response properties table and that the properties specified are valid.

    Note: The serverName property is validated only if specified. The severName property is not required in the managed node response file.
11. Verify that the prerequisite WebSphere Application Server is at the supported level.

Augment profile(s)

The *Augment profile(s)*, encompasses the following actions:

1. Enables Profile Augmentation using (WSPROFILE) scripted actions.

   This is accomplished by invoking the Profile Updater and passing it a profileTemplate directory.

   Which profileTemplate is passed is based on the templatePath value in the response file, which is referenced in the **zWPSConfig.sh** or **zWESBConfig.sh** command line.

   As with all response file properties, the templatePath value may be overridden using the -Z argument on the command line.

   The default response file templatePath looks like this: templatePath=/WebSphere/V6R0M0/AppServer/profileTemplates/default.*, where **default.***

represents a wildcard that invokes the Profile Updater for all the WebSphere Process Server for z/OS profileTemplates. These templates include the following:

```
WebSphere/V6R0M0/AppServer/profileTemplates/default.wbicore
/WebSphere/V6R0M0/AppServer/profileTemplates/default.bfm
/WebSphere/V6R0M0/AppServer/profileTemplates/default.wbiserver
```

For more detailed information on the **profileTemplate** property, see the Response file samples.

About the reconfiguration process

You can re-configure an ESB-only server to a process server.

Running **zWPSConfig.sh** against an ESB-only configuration re-configures WebSphere Process Server for z/OS as a full functioning process server.

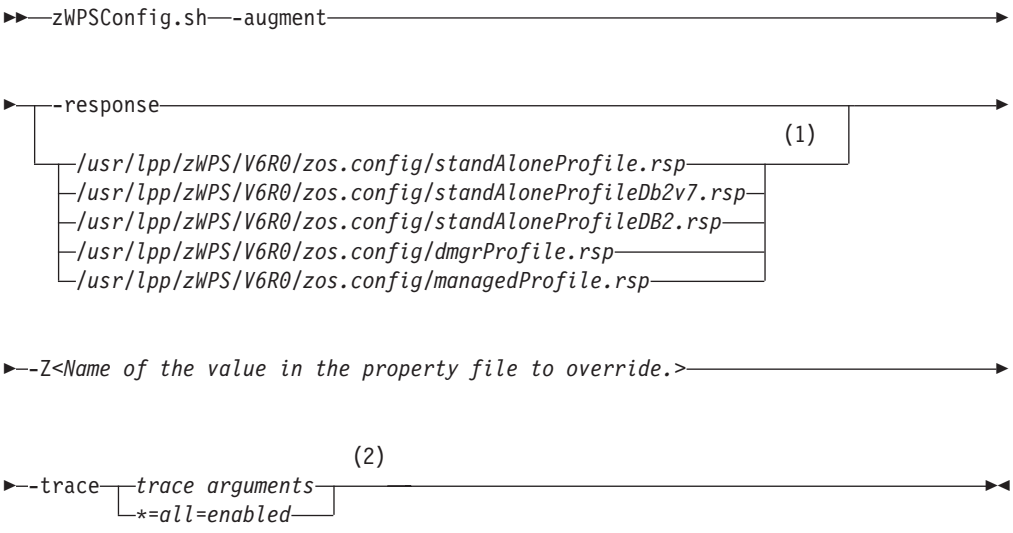
The reason for re-configuring an ESB-only server to a full process server might be driven by a change in the business model. For example, there may be a new business requirement for using and executing process flows. Because an ESB-only configuration does not include Business Process Choreographer functionality, you would need to re-configure the server to include that functionality.

When you run **zWPSConfig.sh**, it validates that there is an existing ESB-only configuration and as part of its actions, it adds to the ESB-only configuration the components (Business Process Choreographer for example) required to configure the server a full Process server.

Command syntax

The following syntax diagram illustrates the syntax of the configuration script for re-configuring an ESB-only server to a full Process server.

Syntax diagram for re-configuring an ESB-only server to a full Process server



Notes:

1 Type the absolute path of the response file that is associated with the configuration onto which you are installing WebSphere Process Server for z/OS. This example shows the default path to the response files packaged with the product. When you run the command, the **-response** parameter will

represent the absolute path to the response file that you have modified with the values specific to your environment.

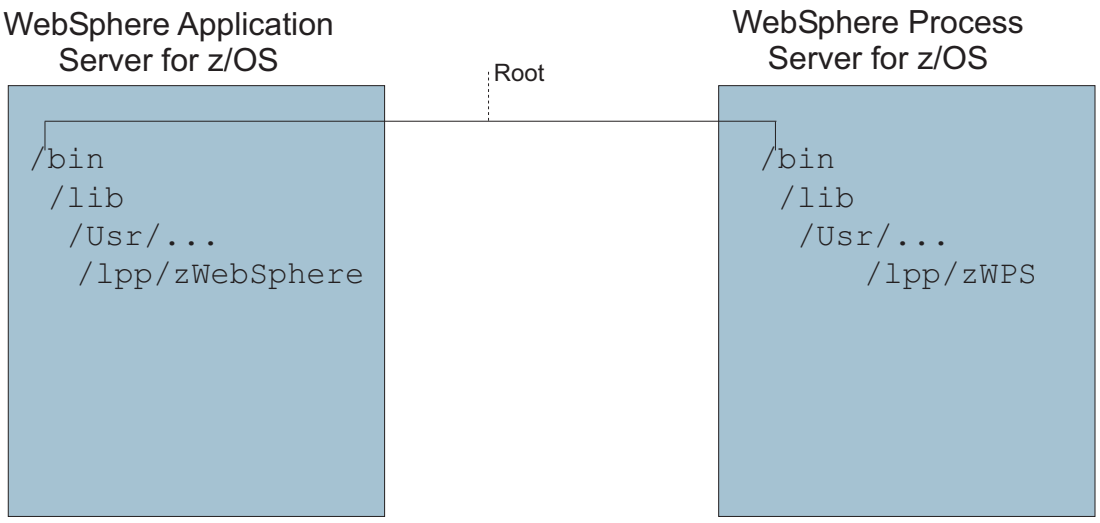
2 By including **-trace** the command writes to the appropriate log file.

For a detailed description of the keywords and keyword parameters in the configuration script, see the table, *Configuring WebSphere Process Server for z/OS - Running the server as a Process Server* in the topic, About the configuration script

Understanding the product file system

The files in the file system resulting from the WebSphere Process Server for z/OS install are organized into a hierarchical structure.

HFS structure for WebSphere Application Server and WebSphere Process Server



All of the files within a hierarchical file system (HFS) are members of a directory. Each directory is, in turn, a member of another directory at a higher level in the hierarchy.

The install HFS represents a file organization that stores data in a top-to-bottom organization structure. All access to the data starts at the top and proceeds downward throughout the levels of the hierarchy.

At the highest level or the starting point in the HFS is the *root directory*. The root directory is the directory that begins with a forward slash (/). All file system entries, including mounted partitions, hang off of the root directory.

Each directory contains information about itself as well as the files contained within, including the file names, their size, the date the files were created or modified, and other relevant information.

The entire HFS file structure, from the root directory on down, is stored as a single data set on an IBM mainframe, and z/OS manages access to specific directories and files. z/OS can manage the hierarchical files through its own HFS facility. As a result, HFS files can be used in both the UNIX System Services and z/OS

environments. This makes it possible for application programs that are designed to run in a UNIX environment to handle files as they normally would under z/OS.

Although most HFS files store data, some can be executable modules and still others can consist of UNIX shell scripts. Executable modules, or programs, are similar to the compiled and linked programs found in z/OS load libraries. UNIX shell scripts are similar to procedures. They consist of a series of UNIX commands (and, optionally, scripting commands) that are executed in order whenever the file is invoked.

The default directory conventions in use for WebSphere Application Server for z/OS are as follows:

- **app_server_root - the install_root for WebSphere Application Server**

  The default installation root directory for WebSphere Application Server for z/OS is the `WAS_HOME/WebSphere/AppServer` directory.
- **profile_root**

  The default profile root directory for WebSphere Application Server for z/OS is the `WAS_HOME/WebSphere/AppServer/profiles/default` directory.

The default directory convention in use for WebSphere Process Server for z/OS is as follows:

- **The install_root for WebSphere Process Server**

  The default profile root directory for WebSphere Process Server is `-PathPrefix-/usr/lpp/zWPS/V6R0`

Environment configuration types

WebSphere Process Server for z/OS installs on top of WebSphere Application Server for z/OS and provides enhanced business process capabilities. WebSphere Process Server for z/OS can be configured to run in a stand-alone or network deployment installation.

WebSphere Process Server for z/OS provides a single, integrated platform that extends the benefits of the WebSphere Application Server for z/OS. It provides business flexibility on a highly scalable, reliable platform. The product offers robust process automation, advanced human workflow, business rules with rich business to business capabilities all on a common, native SOA platform with full enterprise service bus (ESB) connectivity.

WebSphere Process Server for z/OS can be installed and configured to run in the following WebSphere Application Server for z/OS environments:

- WebSphere Process Server for z/OS, Standalone installation
- WebSphere Process Server for z/OS, Network Deployment installation
- WebSphere Process Server for z/OS, Network Deployment Managed Node installation

To specify the environment configuration type during configuration, you can select and customize the response file you will use when you execute the **zWPSConfig.sh** or **zWESBConfig.sh** script.

Stand-alone configuration

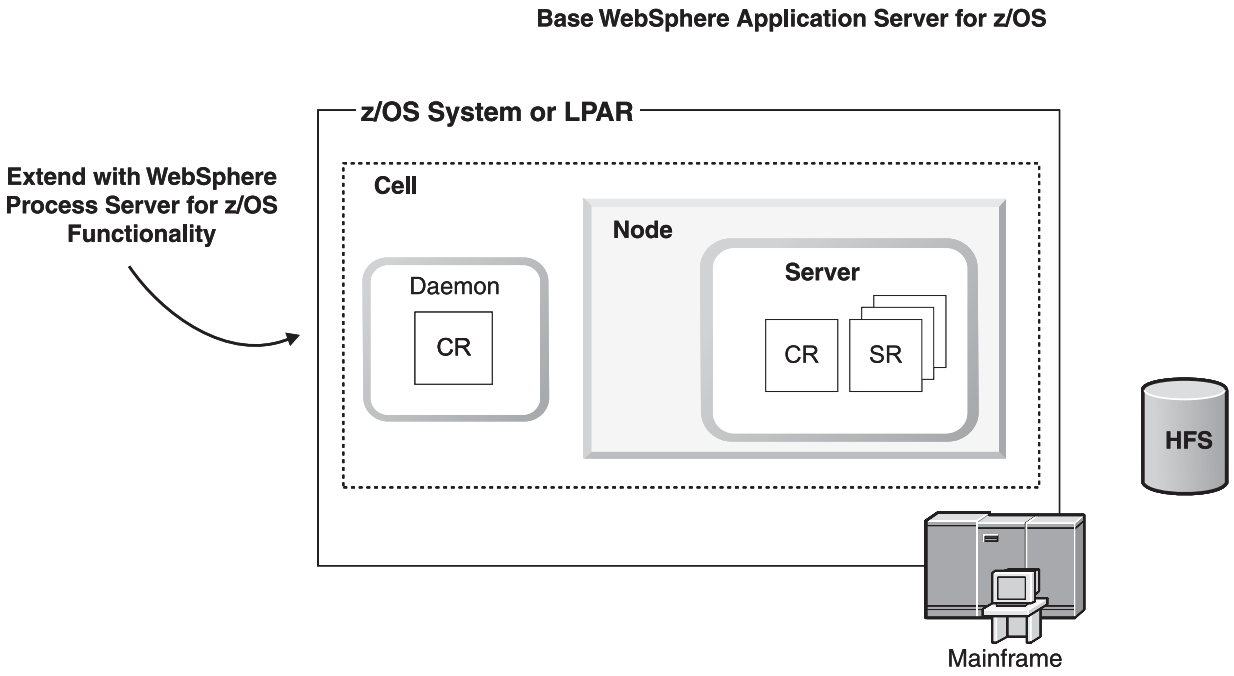
WebSphere Process Server for z/OS can be installed and configured to run in a WebSphere Application Server for z/OS *stand-alone* configuration. This configuration is also called a *base* WebSphere Application Server for z/OS configuration.

Stand-alone Application Server

The stand-alone application server is the easiest and simplest operating structure in WebSphere Application Server for z/OS. It consists of one or more server instances, a daemon server, one node and one cell running in a single z/OS system or LPAR.

- A *cell* represents the boundary of an administrative domain.
- A *node* is a collection of servers grouped together on a single system for the purposes of administration.
- A *daemon* manages requests and owns the interaction with workload management.

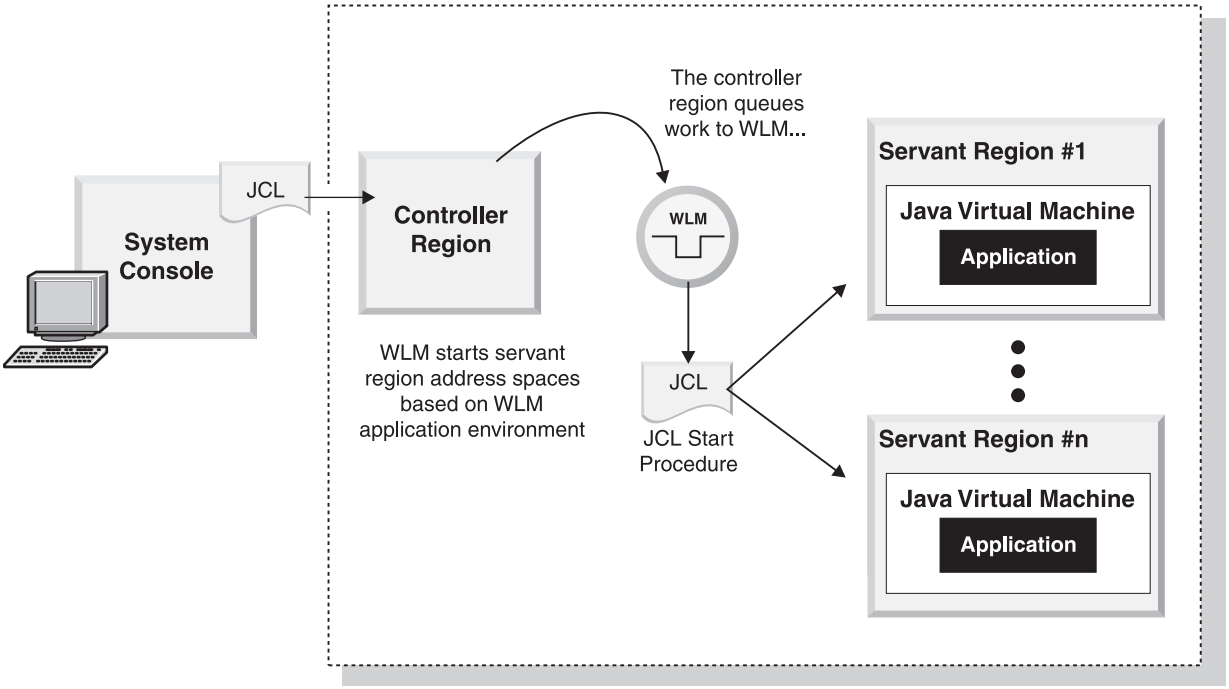
The definitions and configuration files are all kept in the HFS directory structure for the stand-alone application server. The application server relies on a set of XML/XMI files for its configuration repository. The application server administration is based on two tools: the Web browser-based graphical user interface called the administrative console, and the non-graphical command line scripting client.



Application Server

The server instance consists of a *controller region* and a *servant region*. In WebSphere Application Server, the Java Virtual Machines (JVM) reside in the controller and servant address spaces. A controller region and one or more servant regions are

called a *server*. The controller and servant structure design makes it possible to start multiple servants by the workload manager (WLM), based on the workload queued by the controller region.



Daemon Server

The *daemon* server is a special server with only one controller region. It is a supporting daemon that is responsible for accepting and initiating application requests. To perform its work, the daemon must know which servers are active and know all the applications in them. There is only one daemon per cell per system.

Node

A *node* is a collection of servers on a given system or LPAR. The node name in a cell has to be unique. The purpose of a node becomes clearer when the configuration includes several systems or LPARs, and the whole environment is managed from a central administrative console. This would assume that we also have a *Node Agent*, which is a specialized server that receives commands from the central administrator and issues those commands against the servers in its node. The stand-alone application server does not have such a Node Agent, because the administrative Web interface is running within one of its own servers.

A stand-alone application server installation is a single node that runs one application server process independent of a Network Deployment cell.

Cell

A *cell* is a collection of nodes that makes up an administrative domain or boundary. A cell name must be unique and cannot be extended beyond the sysplex. The simplest example of this is the stand-alone application node. The administrative domain extends only to itself, which makes it very small. The cell

can be extended to exist across multiple systems or LPARs in a sysplex and consist of many nodes, which makes the administrative domain very large. Even a stand-alone Application Server node may contain multiple servers, which also expands the administrative domain. Whether it is a very simple or a very large and more complicated configuration, an administrative domain is always required.

Benefits of a standalone configuration

For many reasons, complete isolation between test and production systems is always recommended. The nature of testing implies brand new changes are injected into an application the company depends on. If the changes cause a major failure during test and there is no isolation, the production system could be slowed by bad application code in the test environment.

There are three typical reasons to have WebSphere focused on one LPAR.

- The company is very large and gives each of their test groups an LPAR to exercise the application.
- The company is smaller and needs to run production and test on one zSeries system. Maybe they give one LPAR to test and all others to production.
- The company is trying to decide whether to use WebSphere and gives up a small bit of resources for feasibility testing.

Multiple stand-alone application server nodes can be configured to run in a single LPAR. However, each stand-alone application server node is isolated from the others and contains its own server root (HFS), cell, node, TCP ports and admin interface.

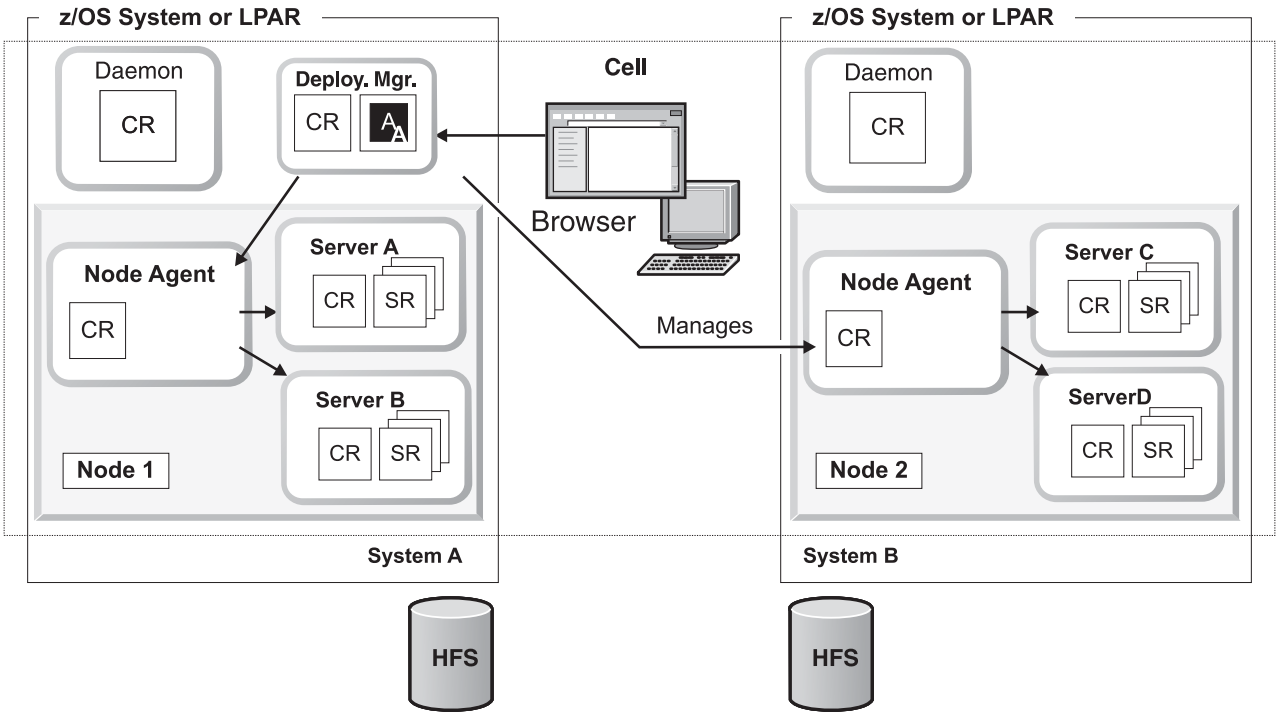
Network deployment configuration

WebSphere Process Server for z/OS can be installed and configured to run in a WebSphere Application Server *network deployment* configuration. This configuration is also referred to as a WebSphere Application Server for z/OS *deployment manager* configuration.

Network deployment configuration

A network deployment configuration facilitates administration of application servers that may be dispersed among multiple sysplexes in a network environment. This feature allows you to share applications and servers.

- A *deployment manager* is a special kind of application server that provides administrative services in a network.
- A *node agent* is a special single controller region server that the deployment manager communicates with to manage the application servers in the node.



Sysplex

A *sysplex* is a set of z/OS systems communicating and cooperating with each other, through certain multisystem hardware components and software services, in order to process customer workloads. Notice that in the example, we have a single cell with multiple nodes that spans systems in a sysplex. The cell is under the management of the deployment manager, and only one deployment manager is permitted per cell, even when the cell spans z/OS systems.

Deployment Manager

The *deployment manager* node is a central point of administrative control. An administrative application (see A in diagram) runs in the deployment manager. The deployment manager does not interact with the servers directly, it communicates with node agents, which in turn control the servers in the node. There is only one deployment manager per cell, no matter how many systems the cell spans in the sysplex.

Base application nodes can be federated (joined) into a deployment manager cell using the addNode shell script or through the administrative console.

Node Agent

The *node agent* is responsible for administration at the node level. Before the deployment manager can manage application servers, those applications servers need to be grouped into nodes. Nodes must stay with a system or LPAR. There is one node agent per node.

The node agent is also a special server in that it has no servant regions. The node agent communicates all configuration changes to the deployment manager at scheduled intervals. Changes can also be forced up to the deployment manager for faster change notification at any time.

Benefits of a network deployment configuration

In a network deployment configuration, the applications and servers are now truly shared. In this configuration, anyone who is logged on to the administrative console can start and stop servers and applications. In a stand-alone configuration, starting and stopping is only allowed from the administrative application contained in the individual server. Taking administration out of the individual server allows for expanded administrative functionality.

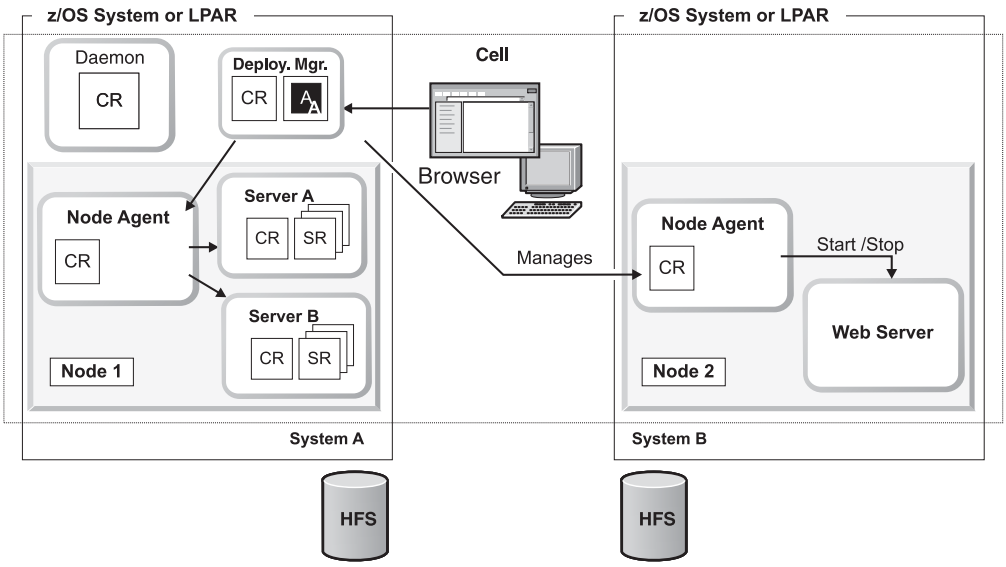
Managed node configuration

WebSphere Process Server for z/OS can be installed and configured to run in a WebSphere Application Server for z/OS *managed node* configuration.

Managed node

A *managed node* is a Web server that is being managed by the deployment manager. A managed node configuration provides the ability to start and stop the Web server and to push the plug-in configuration file to the Web server automatically. The Web server managed node requires an application server node to be created on the Web server system. It is commonly used when Web servers are installed behind a firewall where an application server node can be installed.

You can create an empty managed node and add it to an existing network deployment cell. The managed node will contain a node agent but no application servers. You can add an application server to the node with the administrative console since it can be federated it into an existing network deployment cell.



Unmanaged node

A Web server *unmanaged node* is a Web server in a configuration that is not being managed by any deployment manager. It is commonly used when Web servers are installed outside of a firewall where no application server node can be installed.

Before you begin

To successfully install WebSphere Process Server for z/OS you need to ensure that you are adequately prepared. Before you begin, evaluate your skills and educations needs, install the documentation, determine hardware and software requirements, create your implementation plan, and prepare the base operating system.

Installation: Resources for learning

Use the following links to find relevant supplemental information about installation and customization of WebSphere Process Server. The information resides on IBM and non-IBM Internet sites, whose sponsors control the technical accuracy of the information.

These links are provided for convenience. Often, the information is not specific to the IBM WebSphere Process Server product, but is useful all or in part for understanding the product. When possible, links are provided to technical papers and Redbooks(TM) that supplement the broad coverage of the release documentation with in-depth examinations of particular product areas.

View links to additional information about:

- "Planning, business scenarios, and IT architecture"
- "Programming instructions and examples" on page 39
- "Programming specifications" on page 39
- "Administration" on page 39
- "Support" on page 39

Planning, business scenarios, and IT architecture

- WebSphere Process Server detailed system requirements

  From this page, select the link to your version of WebSphere Process Server. The official site for determining product prerequisites for hardware and software for WebSphere Process Server.
- IBM developerWorks WebSphere

  The home of technical information for developers working with WebSphere products. You can download WebSphere software, take a fast path to developerWorks zones, such as WebSphere Business Integration (of which WebSphere Process Server is a part), learn about WebSphere products through a newcomers page, tutorials, technology previews, training, and Redbooks, get answers to questions about WebSphere products, and join the WebSphere community, where you can keep up with the latest developments and technical papers.
- IBM developerWorks: New to WebSphere Business Integration

  Resource center for those getting started with WebSphere Business Integration. It provides an overview that explains how Eclipse-based WebSphere Business Integration version 6 products work together to take you from high-level business models to deployed applications integrated within a Service-Oriented Architecture (SOA).
- IBM WebSphere Process Server home page

  The IBM WebSphere Process Server home page contains useful information, including support links and downloads for fixes, APARs, tools, and trials.
- IBM WebSphere Process Server library and information center Web site

  The IBM WebSphere Process Server Library Web site contains a link to the WebSphere Process Server information center.

- IBM WebSphere software platform home page

  The IBM WebSphere software platform home page introduces WebSphere products and describes how companies can easily transform to an e-business, with software that can grow as fast as the business it supports.
- IBM developerWorks: IBM Patterns for e-business

  The IBM developerWorks site is the source for IBM patterns for e-business, a set of tested, reusable intellectual assets that you can use to design and implement your e-business network and architecture.
- The User centered design (UCD) for different project types, part 2

  This Web page is the latest of two articles that describes design activities that IBM scientists have found most useful in various types of projects. This article defines user interface design elements, including the design prototype, use case model, and design specification document.

Programming instructions and examples

- IBM developerWorks

  IBM developerWorks contains many excellent resources for developers, including tutorials on Web development-related topics. There is an excellent tutorial on the JDBC API.
- IBM Redbooks

  The IBM Redbooks site contains many documents that are related to WebSphere Application Server, the product which WebSphere Process Server extends.

Programming specifications

- J2EE information

  For more information about J2EE specifications, visit the Sun site.

z/OS

- z/OS Internet Library

  The z/OS Internet Library contains documentation for z/OS elements, features, and software products.
- IBM Education Assistant

  This site integrates narrated presentations, Show Me Demonstrations, tutorials, and resource links to help you successfully use the IBM software products. The materials include information on WebSphere Process Server for z/OS and WebSphere Application Server for z/OS.
- SMP/E Internet Library

  This SMP/E Internet Library contains SMP/E technical documentation and white papers.

Administration

- The IBM Terminology Web site

  This glossary consolidates and defines the terminology from many IBM products in one convenient location. In addition to base computer terminology, terms and definitions from many different IBM brands and product families are included. It is not a comprehensive resource of all IBM computing terms. This resource is provided for information purposes only and is updated periodically. IBM takes no responsibility for the accuracy of the information it contains.

Support

- WebSphere Process Server Support page

Take advantage of the Web-based Support and Service resources from IBM to quickly find answers to your technical questions. You can easily access this extensive Web-based support through the IBM Software Support portal at URL `http://www.ibm.com/software/support/` and search by product category, or by product name. For example, if you are experiencing problems specific to WebSphere Process Server, select **WebSphere Process Server** in the product list. The WebSphere Process Server Support page is displayed.

- z/OS Support page

  This support page provides Web-based Support and Service resources for the z/OS platform.

- IBM e-server Support: Fix Central

  A Web facility for downloading fixes for the zSeries family.

Determining your skill needs

In assembling your project team, consider the skills you need to implement WebSphere Process Server for z/OS.

This article discusses the recommended skill set required to support the WebSphere Process Server for z/OS configurations (standalone, network deployment and managed node on a network deployment configuration).

Documentation to support the z/OS skills described in this article can be found at this Web site: z/OS Internet Library

Documentation describing the skill needs required for WebSphere Application Server for z/OS can be found in the: WebSphere Application Server for z/OS information center.

Creating your implementation plan

Create an implementation plan to provide a roadmap of the sequence of activities to perform for a particular implementation of WebSphere Process Server for z/OS.

The information in this topic assumes that you have already installed and customized WebSphere Application Server for z/OS. For information on installing and customizing WebSphere Application Server for z/OS, see the installation and customization documentation on the WebSphere Application Server for z/OS information center

This procedure guides you through initial planning and implementation of WebSphere Process Server for z/OS. Then it guides you through the process of installing the WebSphere Process Server for z/OS product code and creating the WebSphere definitions that enable the product for use.

Create your implementation plan prior to installing WebSphere Process Server for z/OS. The implementation plan can help you address the necessary requirements for installing WebSphere Process Server for z/OS by listing the tasks that you need to perform and by providing references to the documentation resources that can help you install and configure WebSphere Process Server for z/OS.

WebSphere Application Server for z/OS is installed as a component of WebSphere Process Server for z/OS. You will configure WebSphere Process Server for z/OS in a stand-alone server node or install directly into a network deployment cell.

Note: You can, optionally, federate the base application server node into a network deployment cell.

Perform the following steps to implement your plan, checking off each item as you complete it.

1. Plan for your WebSphere Process Server for z/OS system.

Table 6. Planning the system

Check off	Task description	Information center resource
	Determine the skills that you need.	See *Determining your skill needs* for information.
	Determine the system requirements.	See the system requirements web site for WebSphere Process Server and select *Version 6.0.1 Process Server for z/OS*. WebSphere Process Server for z/OS installs on top of WebSphere Application Server for z/OS . For a list of system requirements see, the *Hardware and software requirements* as documented in the WebSphere Application Server for z/OS Information Center.
	Understand security options and prepare for securing your system.	Security options are set in WebSphere Application Server for z/OS. See Security planning overview information in the WebSphere Application Server for z/OS Information Center.
	Implement Workload Management in goal mode on each z/OS system if necessary.	See *Workload management (WLM) plan strategy* in the WebSphere Application Server for z/OS Information Center.
	Implement Resource Recovery Services (if not already implemented) on each z/OS system.	See *Preparing Resource Recovery Services* in the WebSphere Application Server for z/OS Information Center.
	Plan for your performance and monitoring systems.	See *Monitoring end user response time* in the WebSphere Application Server for z/OS Information Center.
	Plan and define your problem diagnosis procedures.	See *Problem diagnostic plan strategy* in the WebSphere Application Server for z/OS Information Center.
	Consider automatic restart management before you install WebSphere Application Server for z/OS.	See *Automatic restart management* in the WebSphere Application Server for z/OS Information Center.

Table 6. Planning the system (continued)

Check off	Task description	Information center resource
	Perform planning tasks in preparation for unloading the program materials from the installation media.	For information on planning tasks associated with unloading the WebSphere Application Server for z/OS install media, see *Planning for installation* in the WebSphere Application Server for z/OS Information Center. For information on planning tasks associated with unloading the WebSphere Process Server for z/OS install media, see *Planning to unload the product code*.

2. Unload the contents of the WebSphere Process Server for z/OS installation media.

   Before you unload the contents of the WebSphere Process Server for z/OS installation media onto the system, you must first perform the following tasks for WebSphere Application Server for z/OS:

   - Unload the contents of the WebSphere Application Server for z/OS install media.

     For information on how to do this, see *Installing the product and additional software* in the WebSphere Application Server for z/OS information center.
   - Prepare the base operating system.

     For information on how to do this, see *Preparing the base operating system* in the WebSphere Application Server for z/OS information center.
   - Plan for customizing WebSphere Application Server for z/OS

     For information on how to do this, see *Planning for product configuration* in the WebSphere Application Server for z/OS information center.
   - Customize WebSphere Application Server for z/OS

     For information on how to do this, see *Configuring the product after installation* in the WebSphere Application Server for z/OS information center.

Table 7. Install WebSphere Process Server for z/OS

Check off	Task description	Information center resource
	Print the Program Directory for WebSphere Process Server for z/OS.	The Program Directory is packaged with the product install media and is also available as a PDF from the WebSphere Process Server for z/OS library page.
	Unload the WebSphere Process Server for z/OS product code from the install media.	See *Unload the product code from the install media*.

Table 7. Install WebSphere Process Server for z/OS (continued)

Check off	Task description	Information center resource
	Perform the planning activities associated with running the install script.	Before you run the install script (**zSMPInstall.sh**) you should understand the various keyword and keyword values supported by the command line install. To obtain an understanding of the keyword and keyword parameter's that drive the install, see "About the install script" on page 14.
	Run the install script to create the product definitions and update the administrative console with WebSphere Process Server for z/OS product plug-ins.	See *Running the install script*.

3. Configure the product after running the install script.

   After running **zSMPInstall.sh**, you must run another script from the command line to create the product configuration for WebSphere Process Server for z/OS. Depending on how you plan to use WebSphere Process Server for z/OS, you can run one of two product configuration scripts as follows:

   - zWESBConfig.sh

     Run this configuration script if the server you are configuring will run Mediation flows only.

   - zWPSConfig.sh

     Run this configuration script if the server you are configuring will run Process flows.

Table 8. Product configuration tasks for WebSphere Process Server

Check off	Task description	Information center resource
	Understand the product configuration script	See "About product configuration on z/OS" on page 22. To understand the various keyword and keyword values supported by the command line configuration script, see "About the configuration script" on page 23.
	Understand database considerations and perform the necessary setup procedures prior to running the WebSphere Process Server for z/OS configuration script.	As part of your WebSphere Process Server for z/OS configuration planning you should make sure that DB2 for z/OS is setup to support the features that utilize it. See *Considerations for creating the database*.
	Create the databases and storage groups	Create the database and storage group into which you will configure WebSphere Process Server for z/OS See *Creating databases and storage groups*.

Table 8. Product configuration tasks for WebSphere Process Server (continued)

Check off	Task description	Information center resource
	Understand how to set property values in the response file.	The WebSphere Process Server for z/OSconfiguration script includes a default response file that holds property values used to configure the product. You will need to edit some of the property values in the default response file. See *Working with response files*.
	Create the product configuration by running the configuration script.	See "About running the configuration script" on page 128.

4. (Conditional) Perform various post-configuration tasks.

   Depending on environment configuration variables, such as the version of DB2 for z/OS you are using, you may need to run additional Jacl scripts to complete the WebSphere Process Server for z/OS configuration.

Table 9. Post-configuration tasks for WebSphere Process Server

Check off	Task description	Information center resource
	(Conditional) - If the product configuration script did not create the database definitions automatically, run the SQL to create the database definitions.	See *Configuring the database manually*.

You are done when you have checked the applicable tables.

Hardware and software requirements

This topic includes links to additional information on the hardware requirements and software corequisites and prerequisites needed for installing WebSphere Process Server .

See WebSphere Process Server detailed system requirements at http://www.ibm.com/support/docview.wss?uid=swg27006205 and select the link to your version of WebSphere Process Server. If there is a conflict between the information provided in this topic and the information on the detailed system requirements Web pages, the information on the Web pages takes precedence. The information in this topic is provided as a convenience only.

Supported operating systems

See WebSphere Process Server detailed system requirements at http://www.ibm.com/support/docview.wss?uid=swg27006205 and select the link to your version of WebSphere Process Server.

The installation program for WebSphere Process Server verifies that a supported operating system is installed. The verification includes checking for required patches.

Always consult the List of supported hardware and software for WebSphere Process Server Web site to determine whether your operating system is supported. The Web site lists all supported operating systems and the operating system fixes and patches that you must install to have a compliant operating system.

Refer to the documentation for non-IBM prerequisite and corequisite products to learn how to migrate to their supported versions.

Installing the documentation

You can install and run a server copy of the WebSphere Process Server for z/OS information center on z/OS.

Requirements and considerations for installing and running an information center on z/OS

The documentation is packaged as Eclipse document plug-ins and must be viewed using the IBM WebSphere help system. The IBM WebSphere help system (or viewer) and document plug-in format are based on an open source approach developed by the Eclipse Project. IBM product document plug-ins are contained in folders that follow a consistent naming convention (com.ibm.*xxx*.doc).

The IBM WebSphere help system is packaged on the WebSphere Process Server for z/OS product tape. The information in this topic assumes that the IBM WebSphere help system has been loaded on to the z/OS server as part of the WebSphere Process Server for z/OS product install.

You should be aware of the following requirements and considerations if you want to install and run a server copy of the WebSphere Process Server for z/OS information center:

- Support for the information center on z/OS is only offered in server mode for remote access using a browser.
- The information center uses a JRE. If you want to run an information center on z/OS, you need to use the JRE provided with the operating system.
- The information center runs in UNIX ® System Services (USS) on z/OS 1.4 or later.
- To get the best results when viewing the information center, it is recommended that you use one of the following browsers:
  - Microsoft Internet Explorer 6.0
  - Mozilla 1.7
- To view PDF documents within the information center, you must have Adobe Acrobat Reader 4.05 or higher installed, and the Acrobat Web plug-in installed in your browser.
- Make sure that you have proper authority / permissions set by the systems administrator to work with files on the z/OS system.
- The first time you perform a search in the information center, there will be a delay where the pregenerated search indexes are unzipped and any additional files are indexed. In a server environment, the indexing takes place once for every browser locale that performs a search in the information center. For example, if a user has a browser locale of **en_gb** and another user has a locale of **en_us**, both users will create a search index.
- You can choose to download PDF versions of the documentation from the WebSphere Process Server for z/OS library page as another option for working with the information offline.

Note: If you already have an Eclipse help system, or an Eclipse-based IDE such as WebSphere Integration Developer, you can opt to install only the WebSphere Process Server document plug-in and view the documentation within that tool.

Installing the document plug-ins

To view IBM product documentation in the viewer, you must install new or updated document plug-ins into the `eclipse/plugins` folder of the IBM WebSphere Help System. The help system works with any information that has been packaged as an Eclipse document plug-in, including IBM product document plug-ins. IBM product plug-in folders are easily identified because they use a common naming convention (com.ibm.*xxx*.doc).

To install document plug-ins, complete the following steps:

1. Install the Eclipse help system by performing the following steps:
   a. Change to the following directory: `cd install_root/zos.iehs/lib`.
   b. Locate the file **IBM-Help-zOS.tar**
   c. Create the directory into which you will untar **IBM-Help-zOS.tar**. For example:

      `mkdir<dirname>`

      Where <*dirname*> is a user-defined location.
   d. Untar the file into a directory of your choosing by using the command:

      `tar -xvf IBM-Help-zOS.tar -C <directory name>`

      This extracts the help system to the named directory.
2. Obtain the document plug-ins from the IBM product Web page http://www.ibm.com/software/integration/wps/library/infocenter/.
3. Download the document plug-ins that you require.

   You can download zip file to a local directory on your machine and then use FTP to transfer the file to the following directory on the z/OS system.`<directory name>/eclipse/plugins`

   This is the directory that was created when you un-tarred the help system.

   - If you tarred the plug-ins before using FTP, you can use the tar command above to extract them.
   - If you zipped the plug-ins before using FTP, you can use the jar command to unzip them to the right directory. For example, you could enter jar -xvf <*plugins.zip*>, where plugins.zip is a user-defined name.

   These commands maintain the directory structure. Ensure that the plug-ins, when extracted, are placed in the**eclipse/plugins** directory.

   Note: Keep in mind that this must be the `eclipse/plugins` folder of the IBM WebSphere Help system; you might have additional `eclipse/plugins` folders on your system.
4. Change the permissions on the **IC_start.sh** and **IC_end.sh** scripts so that you have permission to edit and execute the files that start and stop the information center.
5. Edit the start up file **IC_start.sh**, that is provided in the `ibm_help` directory, to point to the location of Java.

   For example, you could add the following to the file: export PATH=/usr/lpp/java142/J1.4/bin:$PATH, where `/java142/J1.4/bin:$PATH` is the location of Java.

Note: Eclipse requires Java™ 1.4.2 to run. This is provided with the information center on all platforms except z/OS.

6. The port number is also specified in the file IC_start.sh - the default is 8888. You can change this by editing the -port parameter.

Starting the viewer

Batch files are provided to run the information center, either locally on a single workstation or on a server for multiple users.

You can start the information center from ISPF or using telnet.

To start the information center on a server, perform the following steps:

1. Open the directory where you have installed the help system.
2. Run the file `IC_start.sh`.

   You can run the information center as a background task. This means that even if you log off the workstation or server, the information center will continue to run. Use the following command: ./IC_start.sh &.
3. To verify that the information center has started, open a browser and enter a URL using the name of the server and the port number that is specified in the file `IC_start`.

   For example, you could specify: `http://winmvs26.site.company.com:8888`. The default port is 8888, but you can change this value by editing the startup file `IC_start` and setting the -port parameter to a suitable value.

   Note: The first time you perform a search in the information center, there will be a delay where the pregenerated search indexes are unzipped and any additional files are indexed. This action is only performed once in the local information center. In a server environment, the indexing takes place once for every browser locale that performs a search in the information center. For example, if a user has a browser locale of **`en_gb`** and another user has a locale of **`en_us`**, both users will create a search index.

Shutting down the viewer

You can start the information center from ISPF or using telnet.

To shutdown the information center on a server, perform the following steps:

1. Open the directory where you have installed the help system.
2. Run the file `IC_end.sh`.

   You can run the information center as a background task. This means that even if you log off the workstation or server, the information center will continue to run. Use the following command: ./IC_end.sh &.

Installing the documentation using the help system packaged on the supplemental CD

In addition to packaging a server version of the IBM WebSphere help system on the WebSphere Process Server for z/OS product tape, there is a version of the help system included on the supplemental Message Service Client CD (LCD8-0507 for Linux operating system or LCD8-0506 for Windows operating system), which is included with theWebSphere Process Server for z/OS product.

To install the documentation plug-ins perform the following steps:

1. Install the IBM WebSphere Help System, which can be found on the Messaging Clients supplemental CD.

   You can find the help system component on the \IEHS directory.
2. Obtain the document plug-ins from the IBM product Web page at http://www.ibm.com/software/integration/wps/infocenter/
3. Copy the document plug-in folders to the eclipse\plugins folder of the help system.

   For example, if you installed the help system to C:\WebSphere Help System, you copy the document plug-in folders to: C:\WebSphere Help System\eclipse\plugins

   Keep in mind that this must be the eclipse\plugins folder of the IBM WebSphere Help system; you might have additional eclipse\plugins folders on your system.
4. To see the newly added document plug-in, start the viewer by following the instructions in the Starting the viewer section below (or shut it down by following the instructions in the Shutting down the viewer section below, and then restart it if it was running).

Starting the viewer for help system installed from CD

To start the viewer:

1. Open the WebSphere Help System folder.
2. Double-click the help_start.bat file.

Note: It might take a few minutes for the system to start and the document plug-ins to be displayed the first time you start it.

Shutting down the viewer from help system installed from CD

When you close the viewer by simply closing its window, its processes are still running in the background. This enables a much faster launch during subsequent sessions. However, you must shut down these processes when you install document plug-ins or update the help system with new plug-ins. Shutting down the viewer when not in use also frees up system memory. When you shut down your machine, all help system processes shut down.

To shut down the viewer:

1. Open the WebSphere Help System folder.
2. Double-click the help_end.bat file.

Viewing information in different languages

To view information in a different language, change the regional settings of your operating system to the locale and language of your choice.

Important: The system will display the translated documentation that has been provided by the product owner. If the product owner has not provided the documentation in your chosen language but has provided the English version, the system will display the English version by default, even if the browser language option is set to another language.

Preparing the base operating system

Prepare the base z/OS environment and all of the subsystems associated with z/OS to be able to support your WebSphere Process Server for z/OS installation.

1. Ensure you have an adequate skill level to support the base z/OS environment.
2. Complete the steps in the WebSphere Application Server for z/OS information center that describe how to install the product and additional software.

   For information see *Installing the product and additional software*
3. Identify the z/OS systems on which you plan to run WebSphere Application Server for z/OS.

This topic provides information on prerequisite products and z/OS subsystem prerequisites with which you must comply before you plan for and customize WebSphere Application Server for z/OS.

You prepare the base operating system after you have installed the WebSphere Application Server for z/OS product code.

By performing these tasks you prepare your z/OS target systems to run WebSphere Application Server for z/OS.

By preparing the base operating system to run WebSphere Application Server for z/OS, you are by default preparing it to run WebSphere Process Server for z/OS.

For a description of how to prepare the base operating system, see *Preparing the base operating system* in the WebSphere Application Server for z/OS information center.

.

Once you complete the steps to prepare the base operating system, you need to complete the WebSphere Application Server for z/OS product configuration before moving onto installing the WebSphere Process Server for z/OS product code.

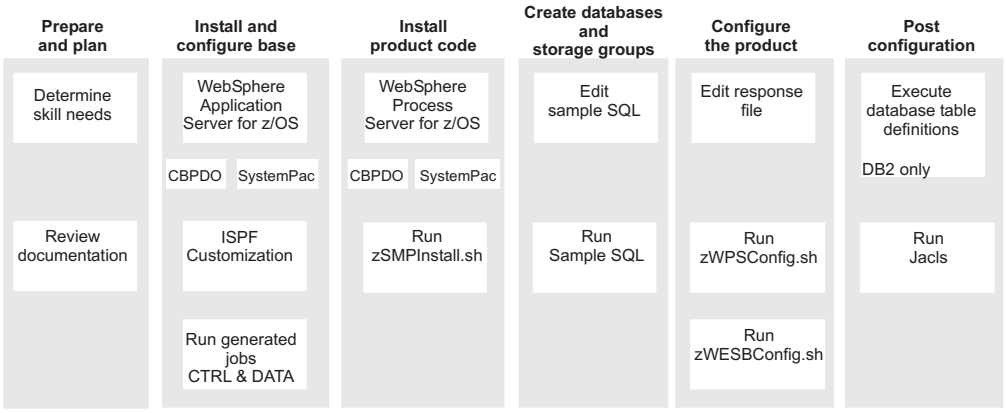
Task overview: installing

This article describes the process of installing and configuring WebSphere Process Server.

Before you begin

This article introduces the context of installing and customizing IBM WebSphere Process Server, including the tasks you need to perform before and after installing.

The following diagram illustrates a high-level task flow for installing and configuring the product.



To create a complete, customized WebSphere Process Server application serving environment, you need to install the product binaries, create WebSphere Process Server definitions, augment your user profile as needed and bring up your server.

The following sections list the sequence of tasks that users perform to install and configureWebSphere Process Server.

The sections below assume that users have already unloaded the WebSphere Process Server product code and unloaded the WebSphere Application Server for z/OS product code using SMP/E.

Use the program directory for WebSphere Application Server for z/OS and for WebSphere Process Server to guide you through the SMP/E process of unloading the modules from the media onto your z/OS system. The result of completing the SMP/E install is an install HFS.

Task flow for installing WebSphere Process Server into a stand-alone configuration

Installing WebSphere Process Server into a stand-alone configuration involves a series of steps that span the WebSphere Application Server for z/OS install and customization procedures and WebSphere Process Server install and configuration procedures.

The list of steps below represent a sequential task flow for installing WebSphere Process Server into a stand-alone configuration.

The details of how to perform these tasks are described in this information center and, where appropriate, in the WebSphere Application Server for z/OS information center.

1. Prepare and Plan your installation.

   Make sure you have access to the required documentation for both WebSphere Process Server for z/OS and WebSphere Application Server for z/OS.

   Prepare your z/OS target systems to run WebSphere Process Server, as described in .
2. Create a stand-alone application server cell.

   Use the ISPF customization dialog to create the stand-alone application server cell, as described in the WebSphere Application Server for z/OS information center.

As a result of completing the ISPF customization dialogs and running the generated jobs, you will have a default profile associated with the stand-alone configuration.

Several values specified on ISPF customization will need to be replicated in the response file that is used by the WebSphere Process Server for z/OS configuration script. These values include:

- Cell name
- Node name
- Server name
- Template path
- Profile path

3. Run the WebSphere Process Server for z/OS install script against the install HFS to create the symbolic links and product definitions.

   This step involves running the WebSphere Process Server for z/OS install script (zSMPInstall.sh) from a command line.

   The install command contains keyword parameters for the WebSphere Process Server for z/OS SMP/E install root, as well as the WebSphere Application Server for z/OS configuration root ($WAS_Home).

   When the install script completes, it extends the WebSphere Process Server for z/OS features into the WebSphere Process Server for z/OS product directory. For example, $WAS_Home /WebSphere/V6R0/ AppServer

4. Create the databases and storage groups for the WebSphere Process Server for z/OS components that use databases.

   Before you run the WebSphere Process Server for z/OS configuration script you need to create the databases and storage groups utilized by WebSphere Process Server for z/OS.

5. Edit the properties in the stand-alone response file.

   WebSphere Process Server for z/OS includes sample response files for each of the supported configurations. These response files are populated with default property values. For a WebSphere Process Server for z/OS stand-alone configuration, you need to edit some of the property values in the response file to tailor it to your environment.

   This response file is used as a keyword parameter in the configuration command. It contains the WebSphere Process Server for z/OS profile information that will be used to augment the default profile that was created when you filled out the customization dialogs and ran the resulting jobs to create the stand-alone application server cell

6. Configure WebSphere Process Server for z/OS by running the configuration script.

   As a result of running the configuration script, configuration actions augment the stand-alone default profile with WebSphere Process Server for z/OS configuration data from the stand-alone response file. Additionally, the DDL and SQL for the database tables is generated.

   Note: You can configure the response file to invoke actions that run the generated DDL and SQL, thereby creating the database definitions automatically.

7. Create the database definitions manually by running the DDL and SQL that was generated by the WebSphere Process Server for z/OS configuration script.

(DB2 for z/OS only) If you did not configure the response file to create the database definitions automatically, you need to run the generated DDL and SQL manually to create the database definitions.

8. Tune for performance. For best performance on any platform, see Tuning the application serving environment.

Task flow for installing WebSphere Process Server into a network deployment configuration

Installing WebSphere Process Server into a network deployment configuration involves a series of steps that span the WebSphere Application Server for z/OS install and customization procedures and WebSphere Process Server install and configuration procedures.

The list of steps below represent a sequential task flow for installing WebSphere Process Server into a network deployment configuration.

The details of how to perform these tasks are described in this information center and, where appropriate, in the WebSphere Application Server for z/OS information center.

1. Prepare and Plan your installation.

   Make sure you have access to the required documentation for both WebSphere Process Server for z/OS and WebSphere Application Server for z/OS.

   Prepare your z/OS target systems to run WebSphere Process Server, as described in .

2. Create a network deployment cell.

   Use the ISPF customization dialog to create the network deployment cell, as described in the WebSphere Application Server for z/OS information center.

   As a result of completing the ISPF customization dialogs and running the generated jobs, you will have a default profile associated with the network deployment configuration.

   Several values specified on ISPF customization will need to be replicated in the response file that is used by the WebSphere Process Server for z/OS configuration script. These values include:

   - Cell name
   - Node name
   - Server name
   - Template path
   - Profile path

3. Run the WebSphere Process Server for z/OS install script against the install HFS to create the symbolic links and product definitions.

   This step involves running the WebSphere Process Server for z/OS install script (zSMPInstall.sh) from a command line.

   The install command contains keyword parameters for the WebSphere Process Server for z/OS SMP/E install root, as well as the WebSphere Application Server for z/OS configuration root ($WAS_Home), and when it completes, it extends the WebSphere Process Server for z/OS features into the WebSphere Process Server for z/OS product directory. For example: $WAS_Home/WebSphere/V6R0/DeploymentManager

4. Create the databases and storage groups for the WebSphere Process Server for z/OS components that use databases.

Before you run the WebSphere Process Server for z/OS configuration script you need to create the databases and storage groups utilized by WebSphere Process Server for z/OS.

5. Edit the properties in the deployment manager response file.

   WebSphere Process Server for z/OS includes sample response files for each of the supported configurations. These response files are populated with default property values. For a WebSphere Process Server for z/OS network deployment configuration, you need to edit some of the property values in the response file to tailor it to your environment.

   This response file is used as a keyword parameter in the configuration command. It contains the WebSphere Process Server for z/OS profile information that will be used to augment the default profile that was created when you filled out the customization dialogs and ran the resulting jobs to create the network deployment cell.

6. Configure WebSphere Process Server for z/OS by running the configuration script.

   As a result of running the configuration script, configuration actions augment the network deployment default profile with WebSphere Process Server for z/OS configuration data from the deployment manager response file. Additionally, the DDL and SQL for the database tables is generated.

   Note: You can configure the response file to invoke actions that run the generated DDL and SQL, thereby creating the database definitions automatically. This can be done only for the WebSphere Process Server for z/OS database.

7. Create the database definitions manually by running the DDL and SQL that was generated by the WebSphere Process Server for z/OS configuration script.

   If you did not configure the response file to create the database definitions automatically, you need to run the generated DDL and SQL manually to create the database definitions.

8. Tune for performance. For best performance on any platform, see Tuning the application serving environment.

Task flow for installing WebSphere Process Server managed node into a network deployment configuration

Installing WebSphere Process Server into a managed node in a network deployment configuration involves a series of steps that span the WebSphere Application Server for z/OS install and customization procedures and WebSphere Process Server install and configuration procedures.

The list of steps below represent a sequential task flow for installing WebSphere Process Server into a managed node in a network deployment configuration.

The details of how to perform these tasks are described in this information center and, where appropriate, in the WebSphere Application Server for z/OS information center.

1. Prepare and Plan your installation.

   Make sure you have access to the required documentation for both WebSphere Process Server for z/OS and WebSphere Application Server for z/OS.

   Prepare your z/OS target systems to run WebSphere Process Server, as described in .

2. Create an empty managed node and add it to an existing Network Deployment cell.

   Use the ISPF customization dialog to create an empty managed node, as described in the WebSphere Application Server for z/OS information center.

   Note: Do not run the job BBOWMNAN to federate the node into the Deployment manager at this time. You federate the managed node into the Deployment Manager later on in this sequence of steps!

   Several values specified on ISPF customization will need to be replicated in the response file that is used by the WebSphere Process Server for z/OS configuration script. These values include:

   - Cell name
   - Node name
   - Server name
   - Template path
   - Profile path

3. Run the WebSphere Process Server for z/OS install script against the install HFS to create the symbolic links and product definitions.

   This step involves running the WebSphere Process Server for z/OS install script (zSMPInstall.sh) from a command line.

   The install command contains keyword parameters for the WebSphere Process Server for z/OS SMP/E install root, as well as the WebSphere Application Server for z/OS configuration root, ($WAS_Home), and when it completes, it extends the WebSphere Process Server for z/OS features into the WebSphere Process Server for z/OS product directory. For example: $WAS_Home/WebSphere/V6R0/AppServer

4. Create the databases and storage groups for the WebSphere Process Server for z/OS components that use databases.

   Before you run the WebSphere Process Server for z/OS configuration script you need to create the databases and storage groups utilized by WebSphere Process Server for z/OS.

5. Edit the properties in the deployment manager response file.

   WebSphere Process Server for z/OS includes sample response files for each of the supported configurations. These response files are populated with default property values. For a WebSphere Process Server for z/OS network deployment configuration, you need to edit some of the property values in the response file to tailor it to your environment.

   This response file is used as a keyword parameter in the configuration command. It contains the WebSphere Process Server for z/OS profile information that will be used to augment the default profile that was created when you filled out the customization dialogs and ran the resulting jobs to create the network deployment cell.

6. Configure WebSphere Process Server for z/OS by running the configuration script.

   As a result of running the configuration script, configuration actions augment the network deployment default profile with WebSphere Process Server for z/OS configuration data from the deployment manager response file. Additionally, the DDL and SQL for the database tables is generated.

   Note: You can configure the response file to invoke actions that run the generated DDL and SQL, thereby creating the database definitions automatically.

7. Create the database definitions manually by running the DDL and SQL that was generated by the WebSphere Process Server for z/OS configuration script.

   If you did not configure the response file to create the database definitions automatically, you need to run the generated DDL and SQL manually to create the database definitions.

8. Run the install script.

   You need to run the install script a second time for the managed node.

9. Edit the properties in the managed node response file.

   WebSphere Process Server for z/OS includes sample response files for each of the supported configurations. These response files are populated with default property values. For a WebSphere Process Server for z/OS managed node configuration, you need to edit some of the property values in the response file to tailor it to your environment.

   This response file is used as a keyword parameter in the configuration command. It contains the WebSphere Process Server for z/OS profile information that will be used to augment the default profile that was created when you filled out the customization dialogs and ran the resulting jobs to create a WebSphere Application Server for z/OS managed node configuration.

10. Configure WebSphere Process Server for z/OS by running the configurations script.

    As a result of running the configuration script, configuration actions augment the managed node default profile with WebSphere Process Server for z/OS configuration data from the deployment manager response file. Additionally, the DDL and SQL for the database tables is generated.

    Note: You can configure the response file to invoke actions that run the generated DDL and SQL, thereby creating the database definitions automatically.

11. Federate the managed node into the Deployment Manager.

    This is accomplished by running the job BBOWMNAN, that you did not run when creating an empty managed node.

12. Administer the managed node configuration as described in the topic *Creating a clustered environment* in the section Administering WebSphere Process Server for z/OS.

13. Tune for performance. For best performance on any platform, see Tuning the application serving environment.

Packaging

WebSphere Process Server for z/OS V6.0.x is packaged as a hierarchical file system (HFS) via SMP/E and will include the HFS of the appropriate release of WebSphere Application Server for z/OS on the same tape.

The WebSphere Process Server for z/OS packaging consists of a tape with the following content:

- WebSphere Application Server for z/OS
- WebSphere Process Server for z/OS

Note: The WebSphere Process Server for z/OS can be configured as a WebSphere ESB server for Mediation flows only.

How to acquire WebSphere Process Server for z/OS

You can obtain the product code in any of the following ways:

- IBM Custom-Built Product Delivery Option (CBPDO) – the system programmer uses SMP/E to unload the product code onto the z/OS system.
- IBM SystemPac / ServerPac – the system programmer copies SMP/E data sets that correspond to the CustomPac service level onto the z/OS system.

To buy the software, contact your IBM representative or IBM reseller, or visit the WebSphere Process Server home page at http://www.ibm.com/software/integration/wps/ and select the *How to buy* link in the left column. For additional information refer to "Install phases" on page 9.

Software supplied with WebSphere Process Server for z/OS

The WebSphere Process Server for z/OS package is delivered in the tape install media. Within the package is the software that you need to install WebSphere Process Server for z/OS, to set up your WebSphere Process Server environment, and to assemble and deploy applications. The tape install media includes WebSphere Application Server for z/OS which needs to be installed and configured prior to installing WebSphere Process Server for z/OS. Also included in each set are optional supplemental software programs that provide additional value and tool support for your production and development environments.

Table 10 lists the software that is provided with the WebSphere Process Server for z/OS product.

Table 10. Software supplied with WebSphere Process Server for z/OS

Software	Description
WebSphere Process Server for z/OS	Based on service-oriented architecture (SOA) and as a single, simplified programming model, WebSphere Process Server is the next-generation business process server that delivers and supports all styles of integration based on open standards to automate business processes that span people, workflows, applications, systems, platforms, and architectures. New in WebSphere Process Server: • Service component architecture - One simplified integration framework that leverages existing IT • Support for all styles of integration - Including human tasks, roles based task assignments, and multilevel escalation. Visual editors for component assembly • Change business processes on the fly with relatively minimal skills • Business rules, business state machines, and selectors to dynamically choose interface based on business scenarios • Broadest reach in integration - Built on Enterprise Service Bus (ESB) technologies and support for IBM WebSphere Adapters • Support for business-to-business (B2B) through a restricted use license of IBM WebSphere Partner Gateway

Table 10. Software supplied with WebSphere Process Server for z/OS (continued)

Software	Description
WebSphere Application Server for z/OS	The industry's premier Java-based application platform, integrating enterprise data and transactions for the dynamic e-business world. WebSphere Application Server, upon which WebSphere Process Server is built, delivers a rich application deployment environment with application services that provide enhanced capabilities for transaction management, as well as the security, performance, availability, connectivity, and scalability expected from the WebSphere family of products. WebSphere Application Server in a network deployment configuration also enables clustering, edge-of-network services, Web services enhancements, and high availability for distributed configurations. See more details about WebSphere Application Server for z/OS packaging.
IBM HTTP Server	The foundation of any e-business application is the Web server. IBM HTTP Server features include: • Easy installation • Support for SSL secure connections • Fast Response Cache Accelerator • IBM support as part of the WebSphere bundle • Hardware crypto support • Administration Server that helps to administer and configure IHS servers. • Help information that uses the easy-to-navigate design that is common to all WebSphere products
Web server plug-ins	WebSphere Process Server supplies a unique binary plug-in module and an associated plug-in configuration file for each supported Web server. The Plug-ins Installation Wizard installs required files and configures the Web server and the underlying application server of WebSphere Process Server to allow communication between the servers.
WebSphere Application Server Application Clients	An application client module is a Java Archive (JAR) file that contains a client for accessing a Java application. Running J2EE and Thin application clients that communicate with the underlying WebSphere Application Server product requires that elements of the Application Server are installed on the workstation on which the client runs. However, if the system does not have the Application Server installed, you can install Application Clients, which provide a stand-alone client run-time environment for your client applications.
IBM Eclipse Help System	Downloadable versions of the WebSphere Process Server documentation are packaged as Eclipse document plug-ins and must be viewed using the IBM Eclipse Help System. The help system (or viewer) and document plug-in format are based on an open source approach developed by the Eclipse Project.

Table 10. Software supplied with WebSphere Process Server for z/OS (continued)

Software	Description
IBM Message Service Clients	Software that provides messaging and Web services capabilities in non-Java environments. Extend interaction between applications and WebSphere Process Server by using the provided clients: • IBM Message Service Client for C/C++ extends the JMS model for messaging to C and C++ applications. • IBM Message Service Client for .NET enables .NET applications to participate in JMS-based information flows. • IBM Web Services Client for C/C++ is a JAX-RPC-like Web services client for C++ that enables C and C++ applications to connect to Web services hosted on WebSphere.
DataDirect Java(TM) Database Connectivity (JDBC) drivers in the `Data Direct` directory.	Two JDBC drivers produced by DataDirect Technologies for enabling connectivity to Microsoft(R) SQL Server. These drivers are the SequeLink and Connect JDBC drivers.
WebSphere Application Server Toolkit	Provides basic assembly and deployment tooling for publishing to an application server, such as WebSphere Application Server Network Deployment. You can also use the tool to perform basic unit testing, debugging, and profiling functions.
WebSphere Process Server Edge Components	Address the needs of highly available, high-volume environments with the Edge components. The Edge components include sophisticated load balancing, caching, and centralized security capabilities. See the WebSphere Application Server Network Deployment Edge Components Web page for more information.
DB2(R) Universal Database(TM) Enterprise Server Edition for WebSphere Process Server	DB2 Universal Database integrates with WebSphere Process Server, key development tools, and platforms, to help make programmers more efficient than ever. A broad array of autonomic or self-managing capabilities can free more administrator time to focus on driving business value. The ease of use in DB2 and the self-managing characteristics might even eliminate the need for dedicated administrators in smaller implementations.
Tivoli(R) Directory Server for WebSphere Process Server	The IBM Tivoli Directory Server product is a powerful Lightweight Directory Access Protocol (LDAP) infrastructure. Tivoli Directory Server provides a foundation for deploying comprehensive identity management applications and advanced software architectures. See IBM Tivoli Directory Server for more information.
IBM Tivoli Access Manager Servers for WebSphere Process Server	IBM Tivoli Access Manager for e-business integrates with e-business applications right out of the box, to deliver a secure, unified, and personalized e-business experience. By providing authentication and authorization APIs and integration, Tivoli Access Manager for e-business helps you secure access to business-critical applications and data that might be spread across the extended enterprise. See IBM Tivoli Access Manager for e-business for more information.

Table 10. Software supplied with WebSphere Process Server for z/OS (continued)

Software	Description
WebSphere Partner Gateway Advanced Edition	WebSphere Partner Gateway offers a consolidated gateway solution to support EDI and Internet standards that can extend enterprise processes to external trading partners. It provides consolidated partner services for process integration with the WebSphere software platform. Business-to-business (B2B) gateway consolidation centralizes a company's B2B communications with trading partner communities, providing a central point of control for interactions among partners, and providing a security-rich environment at the edge of the enterprise.
WebSphere MQ Server	WebSphere MQ is a prerequisite for use of WebSphere Partner Gateway. It provides standards-based reliable and secure connectivity between applications and systems.
WebSphere MQ Clients	Clients for WebSphere MQ Server.

Product version information

The properties/version directory in the *install_root* contains important data about the product and its installed components, such as the build version and build date.

Product information files

This information is included in `WBI.product` and `[component].component` files.

Run the historyInfocommand to create a report about installed maintenance packages. The historyInfo command creates a report on the console and also creates tracking files in the `config_root/properties/version/history` directory.

Time-stamped, detailed logs record each update process in the `properties/version/log` directory of the *config_root*.

This article describes the XML data files that store product version information for WebSphere Process Server for z/OS Version 6.0.x. By default, the document type declarations (DTDs) for these files are in the `properties/version/dtd` folder of the *install_root*, or the server root directory. See the **Directory locations** section of this topic for more information.

XML files in the properties/version directory that store version information:

platform.websphere

One file whose existence indicates that a WebSphere Application Server product is installed. An example of the file follows:

```
<?xml version="1.0" encoding="UTF-8">
<!DOCTYPE websphere PUBLIC "websphereId" "websphere.dtd"
<websphere name="IBM WebSphere Application Server" version="6.0"/>
```

The following XML files in the properties/version directory represent installed items and installation events such as product edition, version, component, and build information.

WAS.product

One file whose existence indicates the particular WebSphere

Application Server product that is installed. The type of product installed is indicated by the <id> tag. Data in the file indicates the version, build date, and build level

For example, <id>ND</id> product indicates that the installed product is WebSphere Application Server Network Deployment. An example of the file follows:

```
<?xml version="1.0" encoding="UTF-8">
<!DOCTYPE websphere PUBLIC "productId" "product.dtd"
<product name name="IBM WebSphere Application Server - ND">
<id>ND</id>
<version>6.0.0version>6.0.0>
<build-info date="02/03/05" level="s0461.18"/>
</product
```

WBI.product

One file whose existence indicates the particular WebSphere Process Server product that is installed. The type of product installed is indicated by the <id> tag. Data in the file indicates the version, build date, and build level

For example, <id>WBI</id> product indicates that the installed product is WebSphere Process Server for z/OS on an Network Deployment configuration. An example of the file follows:

```
<?xml version="1.0" encoding="UTF-8">
<!DOCTYPE websphere PUBLIC "productId" "product.dtd">
<product name name="IBM WebSphere Process Server">
<id>WBI</id>
<version>6.0.0version>6.0.0>
<build-info date="02/03/05" level="s0461.18"/>
</product
```

Reports

After completing the installation, the WebSphere Application Server for z/OS reports will reflect the installed WebSphere Process Server for z/OS product. Refer to the product version information topic in the WebSphere Application Server for z/OS information center for details.

Preparing to install on z/OS

Use this task to prepare to install WebSphere Process Server for z/OS product code.

Prepare z/OS subsystems and do the other tasks in this section before you start installation and configuration activities.

This task assumes that you have installed and customized WebSphere Application Server for z/OS successfully and that you have prepared base operating system.

Print and review “Creating your implementation plan” on page 40.

Assemble a team of people to install and configure the product. Be sure that the team has the skills needed to plan, install, and configure WebSphere Process Server for z/OS. See “Determining your skill needs” on page 40 for more information.

Perform the tasks listed in this topic before you unload the contents of the WebSphere Process Server for z/OS media onto your system and before you run the install and configuration scripts.

Proper preparation can help you avoid problems later on in the install and configuration process.

1. Perform the necessary planning for an SMP/E install of WebSphere Process Server for z/OS. See “Planning to unload the product code.”
2. Know the name of the read-only install HFS that is created as a result of unloading the contents of the WebSphere Process Server for z/OS install media.

   Normally, the read-only HFS would look something like this: /usr/lpp/zWPS/V6R0 .

   It is in the /zos.config/bin subdirectory of this read-only HFS that you should run the install script (zSMPInstall.sh).

   This is also the value of the -smproot keyword that you must specify when running the product install shell script (zSMPInstall.sh). For example, -smproot /usr/lpp/zWPS/V6R0.
3. Know the name of the WebSphere Application Server for z/OS directory structure.

   Normally, this directory structure is /WebSphere/V6R0M0/AppServer.

   When you run the install script, the product configuration scripts (zWPSConfig.sh and zWESBConfig.sh) are copied into the /bin subdirectory of the WebSphere Application Server for z/OS directory structure. It is from this directory that you run the configuration scripts.

   This is also the value of the -runtime keyword that must be specified when running the install shell script (zSMPInstall.sh).

   It is in the /zos.config/bin subdirectory of this read-only HFS that you should run the install script (zSMPInstall.sh). or example, -runtime /WebSphere/V6R0M0/AppServer.
4. Make sure you have authority to run the install and configuration scripts.
5. Understand the use of the task keywords on both the install script (zSMPInstall.sh) and the configuration scripts (zWPSConfig.sh and zWESBConfig.sh). For a description of the keywords and keyword parameters, see “About the install script” on page 14 and “About the configuration script” on page 23.
6. Make sure you have created the necessary databases and the database storage groups, as some of the WebSphere Process Server for z/OS features use databases. You must create the databases and the database storage groups before you run the product configuration script. See Planning your configuration for information on how to create the required databases and database storage groups.
7. Understand how the values in the response file apply to your environment configuration. You will need to edit values in the default response files to suit your environment. In the product configuration command, you include the path name of the modified response file. See Working with response files for information on the contents of the response files.

When you have successfully finished these preparations you are ready to begin the process of installing and configuringWebSphere Process Server for z/OS.

Planning to unload the product code

Use this task to prepare to install WebSphere Process Server for z/OS program materials from the install media onto the z/OS system.

1. Review the pertinent information in *Creating your implementation plan*
2. Print a copy of the Program Directory for WebSphere Process Server for z/OS.

The Program Directory provides specific *program information* that is needed to install, maintain, service, and use the product.

3. Review *Installation: Resources for learning* for reference information you might need during installation.

You must unload the product code from the install media onto the system order for the product administrator to be able to run the WebSphere Process Server for z/OS installation script. Obtain product code from IBM in one of the following formats:

Obtain WebSphere Process Server for z/OS product code from IBM in one of the following formats:

- An **IBM ServerPac or SystemPac**, which consists of loadable product libraries and corresponding SMP/e data sets. A ServerPac or SystemPac contains program libraries with integrated maintenance for one or more products, which include a base operating system, such as z/OS, if desired. Program library data sets are loaded to disk as part of the ServerPac or SystemPac installation. After installation, perform maintenance with SMP/e.
- An **IBM Custom-Build Product Delivery Option (CBPDO)**, which consists of SMP/e relfiles. A CBPDO contains SMP/e relfiles and maintenance for one or more products. Install each product using SMP/e commands (APPLY / ACCEPT) or the corresponding panels. After installation, perform maintenance with SMP/e.

Perform the tasks listed in this section before you begin the SMP/e installation of WebSphere Process Server for z/OS. By planning your product code placement and naming, you can ease future product maintenance and migration tasks.

1. Make sure that the z/OS system on which you will install WebSphere Process Server for z/OS meets the hardware and software requirements. See *Hardware and software requirements* for more information.
2. Identify the software delivery option you will use. To review your product delivery options, see the z/OS and OS/390 delivery options. See *Using and IBM Custom-Build Product Delivery Offering* or *Using an IBM SystemPac or ServerPac* for more information.
3. Learn about WebSphere Process Server for z/OS product data sets. See *Planning for product data sets* for more information.
4. Learn about WebSphere Process Server for z/OS product directories, and plan a mount point convention. See *Planning for product HFS directories* for more information.

When you have finished the planning process, you are ready to install theWebSphere Process Server for z/OS program materials from the install media onto the system. See *Unloading the product code from the install media*

Planning for product HFS directories

WebSphere Process Server for z/OS product code resides in MVS partitioned data sets (the product data sets) and MVS hierarchical file system (HFS) directories (the product directory and its subdirectories).

This article describes the WebSphere Process Server for z/OS product directory.

Product directory

All WebSphere Process Server for z/OS product HFS files reside in the product directory and its subdirectories. Throughout the product and documentation, `install_root` is used to represent the fully-qualified path name of the WebSphere Process Server for z/OS product directory. The default product directory is `/usr/lpp/zWPS/V6R0`.

The product directory and all of its subdirectories should reside in the same hierarchical file system (HFS) or zSeries file system (ZFS) data set. This data set can be the same as the z/OS root or version data set, which is not recommended, or a separate data set used just for WebSphere Process Server for z/OS. The install jobs and program directory assume that such a separate data set is allocated; and we refer to it as wps_hlq.SBPZHFS, where *wps_hlq* represents the product data set name high-level qualifiers.

Product directory and configuration directory

Each WebSphere Process Server for z/OS application serving environment (stand-alone application server node or Network Deployment cell) has configuration files in one or more WebSphere configuration directories. These configuration directories contain symbolic links to files in the product directory.

Planning for product data sets

This article describes the WebSphere Process Server for z/OS product data sets and recommends a product data set naming convention.

WebSphere Process Server for z/OS product code resides in MVS partitioned data sets (which contain the product data sets) and MVS hierarchical file system directories (which contain the product directory and its subdirectories).

The default high-level qualifier for the product data sets is BPZ.

Note: In this and subsequent articles, *wps_hlq* is used to represent the high-level data set name qualifier for a particular set of WebSphere Process Server for z/OS product data sets.

Product data set contents

The WebSphere Process Server for z/OS product data sets are divided into target data sets (used during product customization and execution) and distribution libraries (used to ″back off″ maintenance if necessary).

The WebSphere Process Server for z/OS target libraries are as follows:

wps_hlq.**SBPZEXEC**
: CLIST scripts

wps_hlq.**SBPZJCL**
: JCL for installation jobs

The WebSphere Process Server for z/OS distribution libraries are as follows:

wps_hlq.**ABPZANT**
: HFS files

wps_hlq.**ABPZEBCD**
HFS files (EBCDIC)

wps_hlq.**ABPZEXEC**
CLISTs used by the Customization Dialog

wps_hlq.**ABPZJCL**
JCL for installation jobs

See WebSphere Process Server for z/OS Program Directory (GI10-0781) for allocation information about each target library and distribution library. Updates to this information are included in the Preventive Service Planning (PSP) bucket for each release of WebSphere Process Server for z/OS.

Product data set naming convention

As noted above, certain WebSphere Process Server for z/OS data sets must have the same high-level data set name qualifier in order for the product to function correctly. Product maintenance and migration is easier if all product data sets have the same high-level qualifier.

On the other hand, in order to continue to run WebSphere Process Server for z/OS while applying maintenance, you must have at least two copies of the product data sets: one for the running application execution environment and one to which service is applied.

We recommend you choose a middle level qualifier for each separate release and maintenance level of WebSphere Process Server for z/OS. This middle level qualifier can reflect a very simple test/production distinction, such as with ″BPZ.V6PROD.*″ or ″BPZ.V6TEST.*″, or can include specific service level information, such as with ″WPS.W600102.*″ or ″WPS.W600103.*″.

There are many places where you must specify the product data set names, so, to avoid undue confusion, use the simplest data set naming scheme that accomplishes your maintenance goals.

Unload the product code from the install media

The product code for WebSphere Process Server for z/OS is installed using either an IBM ServerPac/SystemPac or an IBM Custom-Built Product Delivery Option (CBPDO). In a z/OS environment, unloading product media is usually the responsibility of a system programmer.

1. Make sure that the supported version of WebSphere Application Server for z/OS has been installed and customized.

   For information on how to install and customize WebSphere Application Server for z/OS, see Installing your application serving environment.
2. Complete the steps in *Planning to unload the product code*.
3. Have access to the Program Directory for WebSphere Process Server for z/OS .

   The program directory is the primary user assistance for performing the SMP/E installation.

   You can download the program directory in PDF format from the WebSphere Process Server for z/OS download page, at http://www-306.ibm.com/software/integration/wps/library/infocenter/.

This section of the documentation provides guidance on using either CBPDO or an IBM ServerPac/SystemPac to install the WebSphere Process Server for z/OS program materials onto the z/OS system.

1. Order an IBM ServerPac/SystemPac or IBM CBPDO that contains the appropriate WebSphere Process Server for z/OS product.
2. Follow the instructions for the delivery vehicle that you choose:
   a. *Using an IBM SystemPac or ServerPac*

      An IBM SystemPac or ServerPac consists of loadable product libraries and corresponding SMP/E data sets.
   b. *Using an IBM Custom-Build Product Delivery Offering .*

      A CBPDO contains SMP/E relfiles and maintenance for one or more products.
3. File the installation materials for later use during product maintenance.

When you are finished unloading the product code from the install media onto the system, notify the product administrator so that he or she can run the installation script (**zSMPInstall.sh**) to create the WebSphere definitions that enable the product for use. See *Running the install script* for more information.

Using an IBM SystemPac or ServerPac

An IBM CustomPac (SystemPac, ServerPac or ProductPac) is a set of preloaded product data sets bundled with an IBM dialog that is used to load the data sets to disk and perform initial customization.

In general, SMP/E work is not required during installation of a CustomPac offering. Instead, SMP/E data sets that correspond to the CustomPac service level are loaded onto the disk along with the product data sets. You can still use SMP/E to install preventive and corrective service after CustomPac installation.

If you use an IBM SystemPac or ServerPac, follow the instructions in the copy of *ServerPac: Installing your Order* that ships with your SystemPac or ServerPac.

See *ServerPac: Using the Installation Dialog (SA22-7815)* for information about the ISPF dialog used to install a SystemPac or ServerPac.

Notes:

- Be sure to choose a product data set naming convention that allows you to keep and maintain at least two copies of product libraries for maintenance purposes.

  See Planning for product data sets for more information.
- If you are installing from a driving system, make sure the maintenance level of the target system meets requirements for WebSphere Process Server for z/OS.
- When installation is complete, make sure the product data sets are available to your z/OS target system(s) and the product code HFS is mounted at `/usr/lpp/zWPS/V6R0` or a similar mount point of your choice on each target system.

For further information, see the following:

- eSupport web site at http://www.ibm.com/software/webservers/appserv/zos_os390/support
- PSP buckets
- IBM Software Support Center web site at http://www-306.ibm.com/software/support/.

Using an IBM Custom-Build Product Delivery Offering

An IBM Custom-Build Product Delivery Offering (CBPDO) is a set of product tapes for one or more IBM software products that is bundled with cumulative service. Install the products and service on your system using SMP/E.

If you use CBPDO, follow the instructions in the copy of *WebSphere Process Server for z/OS: Program Directory* that ships with your order.

Notes:

- Be sure to choose a product data set naming convention that allows you to keep and maintain at least two copies of product libraries for maintenance purposes.

  See Planning for product data sets for more information.
- If you are installing from a driving system, make sure the maintenance level of the target system meets requirements for WebSphere Process Server for z/OS.
- When installation is complete, make sure the product data sets are available to your z/OS target system(s) and the product code HFS is mounted at `/usr/lpp/zWPS/V6R0` or a similar mount point of your choice on each target system.

For further information, see the following:

- eSupport Web site at http://www.ibm.com/software/webservers/appserv/zos_os390/support
- PSP buckets
- IBM Software Support Center web site at http://www-306.ibm.com/software/support/.

Verify the product code was unloaded

Verify that the SMP/E install WebSphere Process Server for z/OS was successful.

- **Verify that the code was successfully unloaded onto the system**.

  The systems administrator relies on the *Program Directory for WebSphere Process Server for z/OS* for information concerning the material and procedures associated with the loading the contents of the product tapes onto the z/OS operating system.

  The verification of the SMP/E install is an iterative process, where return codes indicate the success or failure of the various jobs run to load the contents of the product tape onto the system.

  There is no Installation Verification Test (IVT) for the SMP/E install. It is assumed that the systems administrator has SMP/E expertise and access to SMP/E documentation to facilitate this portion of the install.

  The following SMP/E documentation is recommended:

  – z/OS: SMP/E Commands, SA22-7771
  – z/OS: SMP/E Messages, Codes, and Diagnosis, GA22-7770
  – z/OS: SMP/E Reference, SA22-7772
  – z/OS: SMP/E User's Guide, SA22-7773
  – z/OS: MVS Initialization and Tuning Reference, SA22-7592

Now that the product code has been loaded onto the system successfully, you can begin to plan to install the WebSphere definitions that enable the product for use.

Planning to run the install script

After the system programmer unloads the WebSphere Process Server for z/OS code onto the system, you can begin the planning process for creating and installing the product definitions.

Generally speaking, the person running the install script should have a sound understanding of how to work with WebSphere products on z/OS.

1. Review Installation: Resources for learning .
2. Print and review the topic *Creating your implementation plan*.
3. Assemble a team of people to enable the product for use. Be sure that the team has the skills needed to plan and enable WebSphere Process Server for z/OS for use.

   See *Determining your skill needs* for more information.

Begin planning to run the install script for WebSphere Process Server for z/OS for use after the system programmer has successfully installed WebSphere Process Server for z/OS code using SMP/E.

The topics listed in this section describe supported scenarios for running the WebSphere Process Server for z/OS install script, as well as topics that describe choosing response files for use in the product install script.

You should read through the information in this section to understand the various topologies into which WebSphere Process Server for z/OS can be installed.

1. Review the supported install scenarios
2. Understand how response files are used in the install script
3. Plan for the Common Event Infrastructure
4. Plan for Business Process Choreographer

When you have finished planning, you are ready to run the WebSphere Process Server for z/OS install script.

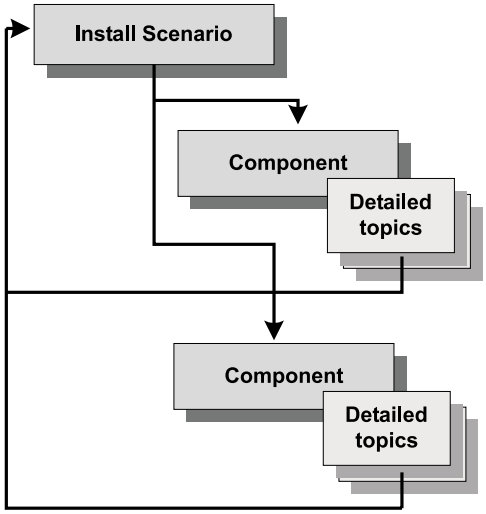
Scenarios for installing

WebSphere Process Server includes many products and components that you assemble to suit your specific business needs. The topics listed in this section of the information center provide detailed planning, installing, configuring, and verifying information to help you set up components within your environment. The following information does not prescribe an installation but allows you to build an environment that is right for you.

Before you begin

Plan your WebSphere Process Server environment. Although individual components have specific planning considerations, it is recommended that you plan your environment before starting installation.

How to use these scenarios



1. Choose an installation scenario. Within the scenario, follow the links for each component.
2. Within the component, follow the ordered links to access detailed topics that describe how to plan, install, configure, and verify the component. Each step includes a link to more detailed instructions.
3. After you are finished with a component, return to the scenario to proceed to the next component.

Installation scenarios

The following scenarios provide the recommended order to install and configure WebSphere Process Server and its components:

- Performing a complete installation in a stand-alone configuration
- Performing a complete installation in a network deployment configuration
- Performing a complete installation in a managed node configuration
- Performing a manual custom installation
- Performing a automated custom installation
- Performing an uninstall

Topology planning considerations

One of the first steps in planning any large software installation is determining the topology in which it will reside. Since WebSphere is middleware, the topologies can become quite complex and require some forethought.

The following list provides the considerations that need to be made while planning an appropriate WebSphere topology. Typically, you will address these issues when you plan your WebSphere Process Server for z/OS installation.

- Security
- Cost
- Administration (maintainability)
- Performance (throughput, response times)
- Availability
- Scalability
- Session state

For detailed information about these topics, refer to the WebSphere Application Server for z/OS documentation.

A stand-alone (single server) topology providing a framework for a quick start or development environment. Also, smaller corporations may find that a single server topology is all they need to meet their requirements.

A network deployment (multi-server) topology facilitates administration of application servers that may be dispersed among multiple sysplexes in a network environment.

See ″Choosing between a stand-alone or network deployment configuration″ topic in the WebSphere Process Server for z/OS documentation infocenter for specifics on the differences between a WebSphere Process Server for z/OS stand-alone cell and network deployment cell.

Planning for a stand-alone application server cell

A stand-alone application server cell is the simplest WebSphere Process Server for z/OS configuration on which you can deploy and run WebSphere Process Server applications.

A stand-alone application server cell includes the following:

- A basic cell and node configuration
- A location service daemon
- An application server that runs the administrative console application. You can deploy and run additional applications on this server.

You cannot add additional servants to an application server running the stand-alone version of the administrative console application. You can define additional application servers in the stand-alone cell, but you cannot control them using the administrative console. For more complicated or robust WebSphere Process Server application-serving environments, the network deployment cell configuration is recommended.

Planning for a network deployment cell

A network deployment cell is a full-function WebSphere Process Server for z/OS configuration on which you can deploy and run applications.

A network deployment cell includes the following:

- A cell configuration.
- A deployment manager that runs the administrative console application.
- One or more application server nodes (one is recommended) on each z/OS target system hosting portions of the cell. Each node consists of a node agent and some number of application servers.
- A single location service daemon on each z/OS system.

This part of the configuration process creates the initial cell configuration, the deployment manager, and a location service daemon for the z/OS system on which the deployment manager runs. Once the network deployment cell is created, add application server nodes by creating and federating new managed nodes, or by federating stand-alone application server nodes into the network deployment cell.

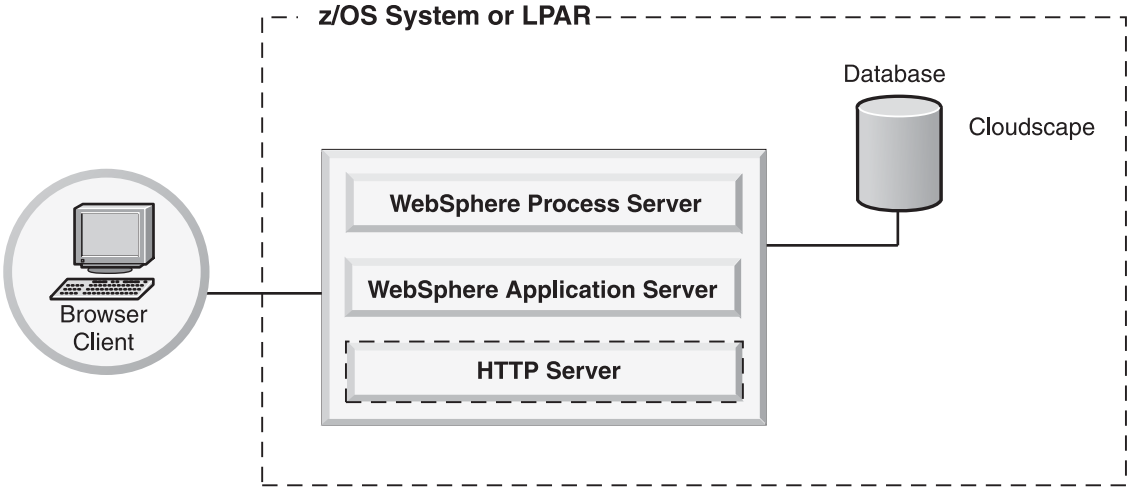
When configuring your deployment manager node, keep the following in mind:

- When allocating target data sets for this option, it is possible, though not recommended, to use the same target data sets that you used for the stand-alone application server node. The jobnames for each configuration are very close to one another; and if you use the same target data sets, you might find it difficult to keep the two sets of jobs separate. Therefore, it is better to create a new set of target data sets and keep the two sets of jobs separate from one another.
- If possible, set up your HFSs such that the root HFS is shared among all processors and the deployment manager's configuration is in a configuration HFS on a system-generic mount point.

Scenario: Complete install into a stand-alone configuration

This scenario outlines how to perform a complete installation in a stand-alone configuration.

Refer to the following outline and choose the options for a WebSphere Process Server installation in a stand-alone configuration:



1. WebSphere Application Server for z/OS
   - Installing your application serving environment
2. WebSphere Process Server for z/OS
   - Preparing to install on z/OS
   - Preparing the base z/OS environment

     Note: If the version of WebSphere Process Server for z/OS that resides on the system does not meet the prerequisite level required by the product you are installing, SMP/E will halt the install. WebSphere Process Server for z/OS service may need to be applied to ensure that the base product is at the level to support the product being installed.
   - Planning to unload the product code
   - Unload the product code from the install mediaWebSphere Process Server for z/OS
   - Verifying the product code was unloaded
   - Running the install script
     - Log onto system with the appropriate credentials to install the product.
     - Run the installation script, **zSMPInstall.sh**, with the `-install` argument

- Considerations for creating the database
- Create the database and storage groups
- Work with response files
- Configuring
- Verifying the install
- Post configuration

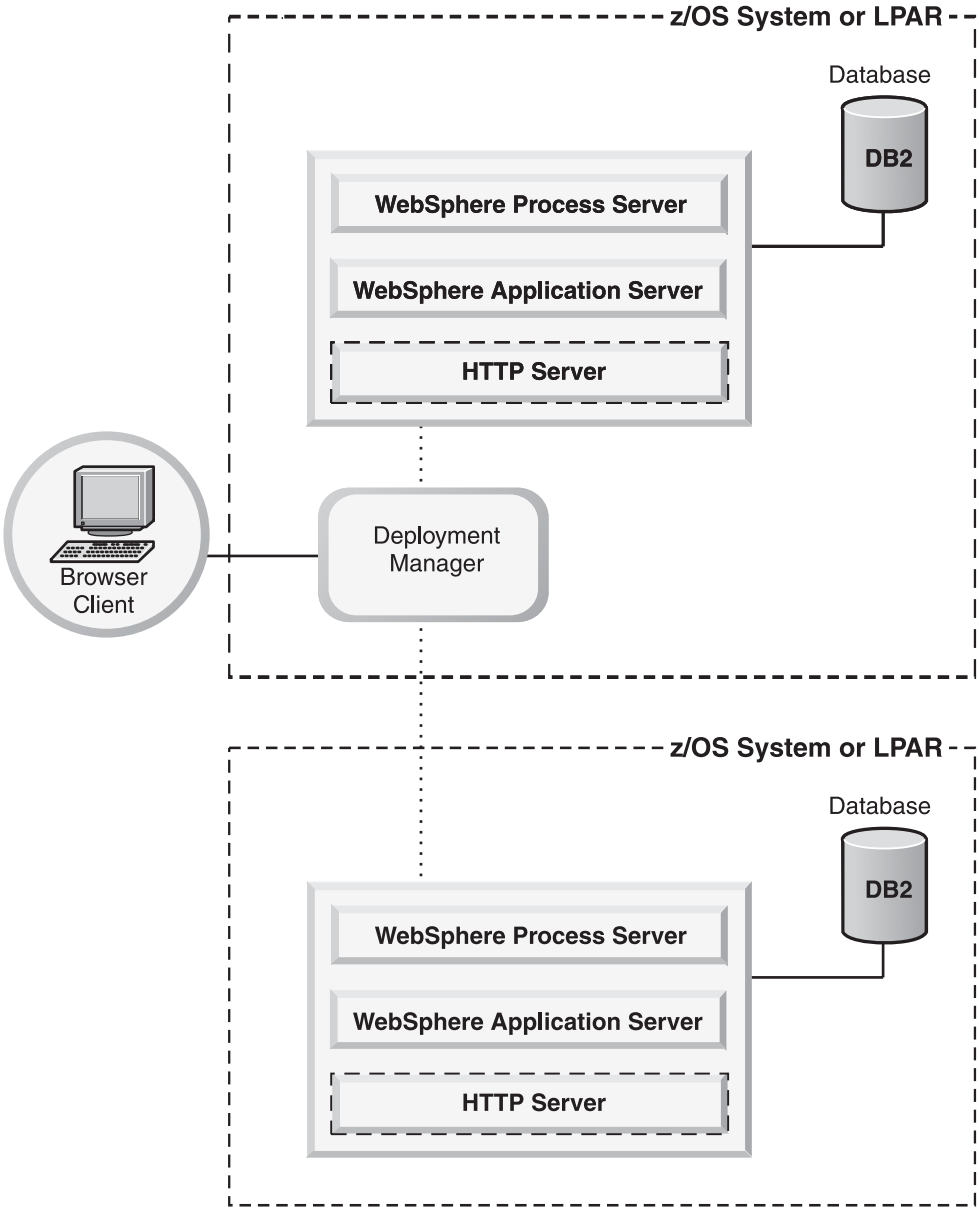
3. Business Process Choreographer
   - Planning to use Business Process Choreographer
   - Configuring Business Process Choreographer
4. Common Event Infrastructure
   - Planning to use Common Event Infrastructure
   - Configuring the Common Event Infrastructure

You can now begin using your installation.

Scenario: Complete install into a network deployment configuration

This scenario outlines how to perform a complete installation in a network deployment configuration.

Refer to the following outline and choose the options for a WebSphere Process Server for z/OS installation in a network deployment configuration:



1. WebSphere Application Server for z/OS
   - Installing your application serving environment
2. WebSphere Process Server for z/OS
   - Preparing to install on z/OS
   - Preparing the base z/OS environment

     Note: If the version of WebSphere Process Server for z/OS that resides on the system does not meet the prerequisite level required by the product you are installing, SMP/E will halt the install. WebSphere Process Server for z/OS service may need to be applied to ensure that the base product is at the level to support the product being installed.
   - Planning to unload the product code
   - Unload the product code from the install mediaWebSphere Process Server for z/OS
   - Verifying the product code was unloaded

- Running the install script
  - Log onto system with the appropriate credentials to install the product.
  - Run the installation script, **zSMPInstall.sh**, with the `-install` argument
- Considerations for creating the database
- Create the database and storage groups
- Work with response files
- Configuring
- Verifying the install
- Post configuration

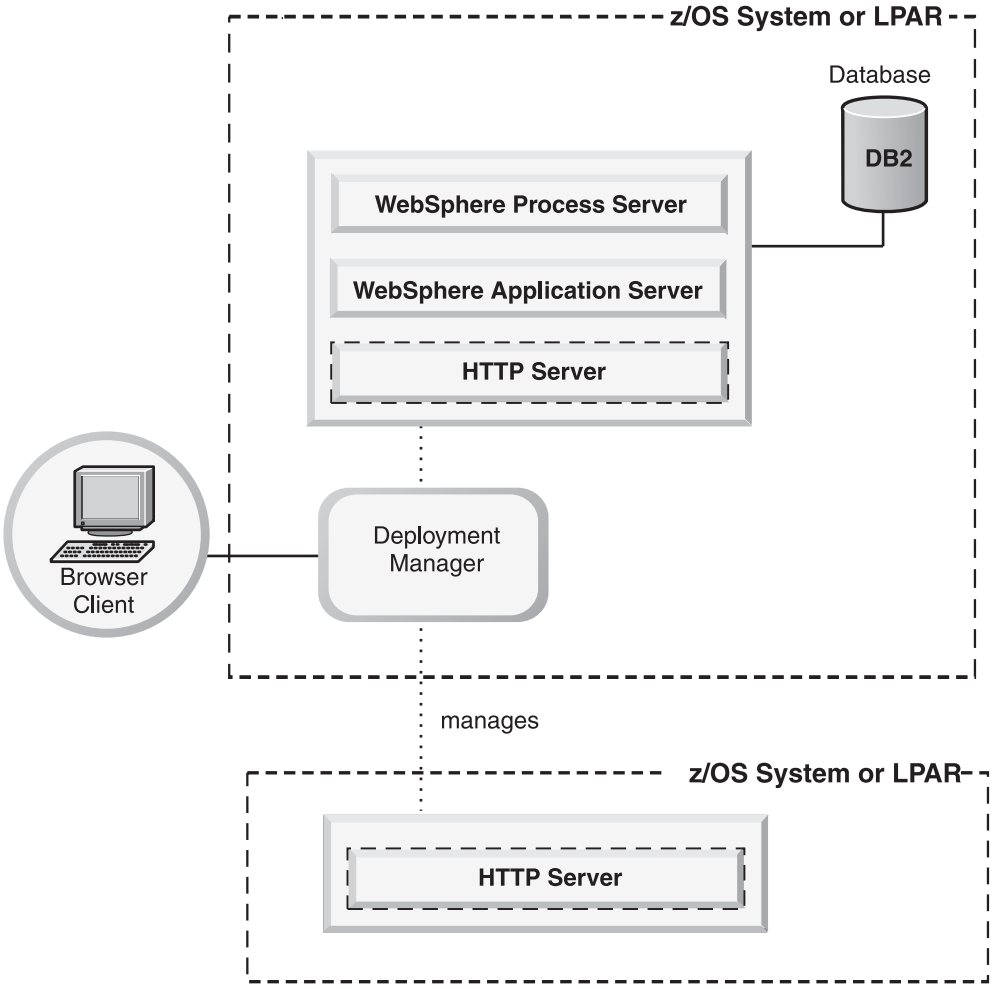
3. Business Process Choreographer
   - Planning to use Business Process Choreographer
   - Configuring Business Process Choreographer
4. Common Event Infrastructure
   - Planning to use Common Event Infrastructure
   - Configuring the Common Event Infrastructure

You can now begin using your installation.

Scenario: Complete install into a managed node configuration

This scenario outlines how to perform a complete installation in a managed node configuration.

Refer to the following outline and choose the options for a WebSphere Process Server for z/OS installation in a managed node configuration:



1. WebSphere Application Server for z/OS
   - Installing your application serving environment
2. WebSphere Process Server for z/OS
   - Preparing to install on z/OS
   - Preparing the base z/OS environment

     Note: If the version of WebSphere Process Server for z/OS that resides on the system does not meet the prerequisite level required by the product you are installing, SMP/E will halt the install. WebSphere Process Server for z/OS service may need to be applied to ensure that the base product is at the level to support the product being installed.
   - Planning to unload the product code
   - Unload the product code from the install mediaWebSphere Process Server for z/OS
   - Verifying the product code was unloaded
   - Running the install script
     - Log onto system with the appropriate credentials to install the product.
     - Run the installation script, **zSMPInstall.sh**, with the `-install` argument
   - Considerations for creating the database
   - Create the database and storage groups
   - Work with response files

- Configuring
- Verifying the install
- Post configuration

3. Business Process Choreographer
   - Planning to use Business Process Choreographer
   - Configuring Business Process Choreographer
4. Common Event Infrastructure
   - Planning to use Common Event Infrastructure
   - Configuring the Common Event Infrastructure

You can now begin using your installation.

Scenario: Uninstalling

This scenario outlines how to uninstall WebSphere Process Server for z/OS.

The following scenario describes how to perform a complete uninstall of WebSphere Process Server for z/OS.

Note: There is no support for a partial or incremental uninstall.

WebSphere Process Server for z/OS

- Uninstalling WebSphere Process Server for z/OS
  - Log onto system with the appropriate credentials to install the product.
  - Run the installation script, **zSMPInstall.sh**, with the `-uninstall` argument. The script performs the following tasks:
    - Disables the WebSphere Process Server for z/OS by running Configuration Manager scripted actions. This removes any Administrative Console plug-in extensions.
    - Removes Profile Augmentation using the WSPROFILE scripted actions. This results in the unaugmentation of the WebSphere Process Server for z/OS default profile.
    - Delete the post install file.
    - Removes code base permissions.

  Note: If any shared common components are being shared by other applications the command line prompting warns that un-installation of the product may cause other applications to no longer function correctly.

  Uninstall is blocked at this point unless the user specifically selects to continue the un-installation by affirming the wish to continue with the uninstall process.

  The user is warned either that augmented default profile will be deleted (if underlying WebSphere Application Server for z/OS or WebSphere Application Server Network Deployment for z/OS is being uninstalled) or that augmented profiles will no longer be usable.

The WebSphere Process Server for z/OS product has been removed from your system.

Planning to use Business Process Choreographer

For each application server or cluster that runs business processes or human tasks, you will have to configure the business process container and the human task container before installing any enterprise applications that contain business processes or human tasks.

1. If you intend to use Business Process Choreographer on a cluster plan your cluster.
2. Decide which database system to use:
   - Cloudscape

     Note:

     – Because Cloudscape™ Network Server has no XA support, Business Process Choreographer can only use the embedded Cloudscape version that cannot be accessed remotely. This restriction is why Cloudscape cannot be used as database system for Business Process Choreographer in a clustered environment.

     – Cloudscape serializes database access. Activities are therefore always performed sequentially, even in flows that are modeled to support the parallel execution of activities.
   - DB2® for z/OS®
3. Check the requirements for the DB2 for z/OS Universal JDBC Driver provider and data source.
4. Decide which server you want to host the database. If the database server is remote, you need a suitable database client or a type-4 JDBC driver that has XA-support.
5. Decide which Java™ Message Service (JMS) provider you will use:
   - WebSphere default messaging
   - WebSphere MQ
6. Plan the settings that are described in "Business process container installation wizard settings" on page 156.
7. Plan the settings that are described in "Human task container installation wizard settings" on page 171.
8. Decide if you will configure the business process container manually (recommended) or if you will use the installation wizard to configure the business process container.
   - If you are going to configure the business process container manually, plan the settings as described in "Business process container settings" on page 166.
   - If you are going to use the installation wizard, plan the settings described in "Business process container installation wizard settings" on page 156.

After installing WebSphere Process Server, you are ready to perform "Configuring Business Process Choreographer" on page 136.

About Business Process Choreographer

This describes the facilities provided by the business process container and the human task container.

Business Process Choreographer is an enterprise workflow engine that supports both business processes and human tasks in a WebSphere Application Server environment. These constructs can be used to integrate J2EE resources, services, and activities that require human interaction. Business Process Choreographer

manages the life cycle of business processes and human tasks, navigates through the associated model, and invokes the appropriate services.

Business Process Choreographer provides the following facilities:

- Support for business processes and human tasks. Business processes offer the standard way to model your business process using the Web Services Business Process Execution Language (WS-BPEL, abbreviated to BPEL). With human tasks, you can use the Task Execution Language (TEL) to model the interactions that involve humans, such as human-to-human, human-to-computer, computer-to-human. Both business processes and human tasks are exposed as services in a Service Oriented Architecture (SOA) or Service Component Architecture (SCA); they also support both simple data objects and business objects.
- Application programming interfaces for developing customized applications for interacting with business processes and human tasks.
- Business Process Choreographer Explorer. This Web application offers functions for managing and administering business process and human tasks.

About business processes:

A process is a set of business-related activities that are invoked in a specific sequence to achieve a business goal.

A process that is defined in the Web Services Business Process Execution Language (WS-BPEL) comprises:

- The activities that are the individual business steps within the process. An activity can be one of several different types. Also, an activity can be categorized as either a basic activity or a structured activity.
  - Basic activities are activities that have no structure and do not contain other activities.
  - Structured activities are activities that contain other activities.
- The partner links, also known interface partners or reference partners, that specify external entities and partners that interact with the process or vice versa using WSDL interfaces.
- The variables that store messages that are passed between activities. They represent the state of a business process instance.
- Correlation sets that are used to correlate multiple service requests or response messages with the same business process instance. Correlation sets are based on application data that is contained in messages that are exchanged with the process.
- Fault handlers that deal with exceptional situations that can occur when a business process runs.
- Event handlers that receive and process unsolicited messages in parallel to the normal execution process.
- Compensation handlers that specify the compensation logic for a single activity or a group of activities.

For more information on these constructs, refer to the BPEL specification.

Business Process Choreographer also supports the IBM® extensions to the BPEL language, such as:

- Human task activities for human interaction. These inline participating tasks can be almost any step in the business process that involves a person, for example, completing a form, approving a document or drawing, writing a letter, and so on.
- Script activities for running inline Java code. The Java code can access all of the BPEL variables, correlation properties, and partner links, as well as process and activity contexts.
- Valid-from timestamps for process model versioning.
- Common Event Infrastructure (CEI) logging.
- Explicit checkpointing to support multiple activities in one transaction.
- Timeouts for activities.

Business process types:

Business processes can be either long-running or microflows.

Long-running processes

A long-running business process is interruptible, and each step of the process can run in its own physical transaction. Long-running business processes can wait for external stimuli. Examples of external stimuli are events that are sent by another business process in a business-to-business interaction, responses to asynchronous invocations, or the completion of a human task.

A long-running process has the following characteristics:
- Runs as several transactions
- Consists of synchronous and asynchronous services
- Stores each intermediate process state, which makes the process forward-recoverable

Microflows

A microflow runs in one physical thread from start to finish without interruption. Microflows are sometimes referred to as non-interruptible business processes. Microflows can have different transactional capabilities. A microflow can run within a global transaction or as part of an activity session.

A microflow has the following characteristics:
- Runs in one transaction
- Normally runs for a short time
- Does not store run-time values in the database
- Consists of only synchronous services and non-interruptible subprocesses, which means that a microflow cannot contain:
  - Human tasks
  - Wait activities
  - Multiple receive activities
  - Should not invoke long-running services
  - Should not invoke activities bound to asynchronous protocols

About human tasks:

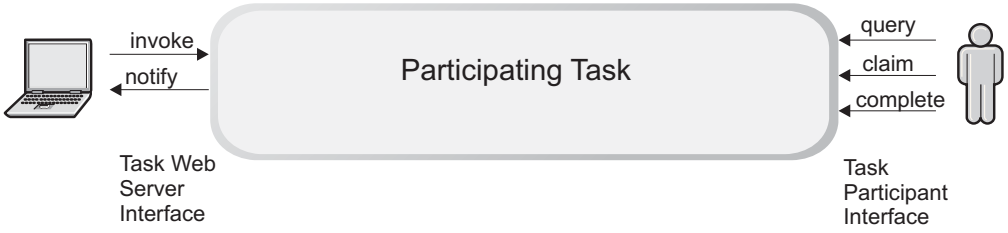
A human task is a component that involves a person interacting with a service or another person.

The interaction can be initiated either by a person or by an automated service. A service that is initiated by a person can be either an automated implementation or a service that is provided by another person. A human task that is invoked by an automated service can be replaced easily by an automated implementation.

Tasks can be used to implement staff activities in business processes that require human interactions, such as manual exception handling and approvals. All other exception handling is modeled natively in Web Services Business Process Execution Language (WS-BPEL, abbreviated to BPEL), by using faults and fault handlers, or compensation.

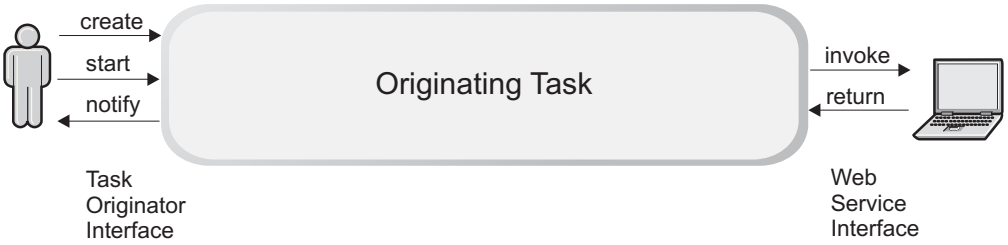
The types of human tasks are as follows:

Participating tasks
Support Web-service-to-person interactions, which enable a person to implement a service. For example, a participating task can be a staff activity in a business process.



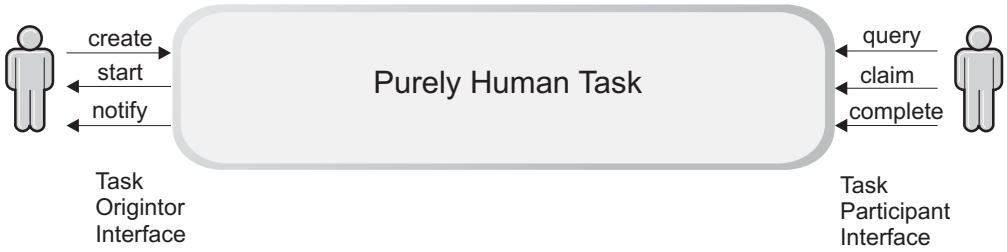
Administrative tasks
Administrative tasks are similar to participating tasks, except that they are used by administrators to solve technical problems that occur in processes. Administrative tasks support authorization and user interface settings for starting or administering business processes and human activities. Currently, administration tasks are only created and managed by Business Flow Manager.

Originating tasks
Support person-to-computer interactions, which enables people to create, initiate, and start services through a graphical user interface. For example, a user can start a business process, or send it an event by means of an originating task.



Purely human tasks
Support person-to-person interactions, which enable a person to invoke a task as though it were an originating task. This invoked task is then performed by another person, who interacts with it as though it were a

participating task. Purely human tasks do not interact with business processes or other Web services.



Who can interact with a task can be determined using one of the supported staff directories. Work items are created for users who have a reason to view or interact with the task.

The human task manager supports the following types of user registry:

- Lightweight Directory Access Protocol (LDAP) user registry
- WebSphere user registry

Escalations

An escalation is a course of action that is executed when a task is not completed satisfactorily within a specific period of time. For example, if tasks are not claimed or are not completed within a defined time limit. You can specify one, or more, escalations for a task. These escalations can be started either in parallel or as a chain of escalations.

Escalations are initialized when the associated task reaches a certain state in its lifecycle. After a well defined duration, the task state is verified, and if it does not meet the modeled expectation, the escalation action is invoked. The following escalations actions are supported:

- Work items are created for a set of users
- E-mails are sent to the designated recipients
- Notification events are sent to registered consumers

Business Process Choreographer scenarios for clustering

Describes different configuration options and considerations for Business Process Choreographer scenarios that use clusters.

The main advantages of using WebSphere Process Server clusters for Business Process Choreographer instances are:

- High service availability due to failover
- Increased workload capacity
- Improved resource utilization
- Workload sharing
- Easier administration

Configuration options

You can configure Business Process Choreographer in many different ways, so cluster configurations are usually very complex. Some of the main options to consider before you start creating application servers are outlined in the following descriptions:

Number of nodes in the WebSphere Process Server cell
One or more. All nodes are administered from a single deployment manager.

Number of nodes in each WebSphere Process Server cluster
One or more. Horizontal WebSphere clustering can increase service availability and the total workload capacity.

Number of application servers in each node
One or more. Vertical WebSphere Process Server clustering can increase resource utilization.

Database host

- Remote, on a dedicated server
- Local to one of the application servers in the cluster

It is recommended to host the database on a dedicated server, preferably one with a hot standby.

Application messaging queues

- Local queues
- Remote queues

Connection (default messaging)
When default messaging is used, you can configure the message engines in the same cluster or in a remote cluster. For Business Process Choreographer, you should use the same approach used for the other WebSphere Process Server components. For more details about possible scenarios, see ″Preparing a server or cluster to support service applications″ in the PDF for Administration. The recommended configuration is to have the messaging engines run in a different cluster than the cluster in which Business Process Choreographer is installed. This is similar to the central queue manager configuration that can be used with WebSphere MQ.

Connection (WebSphere MQ queue managers)

- One central (remote) queue manager hosting the queues for the application servers within one cluster. This configuration is generally recommended.
- One local queue manager per application server. This provides no failover and no intraprocess workload sharing.
- Two local queue managers per node, and WebSphere MQ clustering used to balance workload across several application servers.

  Workload distribution between different Business Process Choreographer instances requires that the queue managers that are used by the business process container of each application server are members of the same WebSphere MQ cluster. This configuration provides no failover and is not generally recommended.

WebSphere MQ is not recommended as JMS provider if a clustered scenario is used.

Database system
You can use any of the supported databases except Cloudscape.

Hot standby servers
You have the following options for hot standby servers:

- None
- For the database
- For a central queue manager

Cluster types: This topic refers to two different types of *cluster*. A *WebSphere cluster* groups application servers together to share workload and increase service availability. A *WebSphere MQ cluster*, previously known as an MQSeries® cluster, groups together WebSphere MQ queue managers and can be used to achieve intraprocess workload balancing.

High availability

To achieve high availability of Business Process Choreographer services, consider the following items:

- By creating cloned application servers in a WebSphere cluster, the services provided by the application servers become highly available.
- The Business Process Choreographer database is a single point of failure that can be protected using a hot standby system.
- A central queue manager can be protected by hot standby hardware.

Vertical clustering to maximize resource utilization

To improve performance, you might have to create multiple application server instances on the same node so that Business Process Choreographer can use the available system resources.

Workload sharing

If you want different instances of Business Process Choreographer to share the same workloads, they must use one of the following queue manager configurations:

Central queue manager
A central queue manager hosts the queues that are needed by Business Process Choreographer. All Business Process Choreographer instances in the WebSphere cluster read from the same queues.

WebSphere MQ cluster
Each application server has two queue managers. One queue manager hosts local queues and is used for getting messages, the other queue manager hosts no queues and is used only for putting messages. All the queue managers of all the Business Process Choreographer instances in the WebSphere cluster are made members of a WebSphere MQ cluster.

The result of only putting to queue managers that host no queues is that the messages are distributed evenly across all the get queue managers in the cluster. After using the installation wizard to install and configure the business process container on the cluster, you must manually change the two connection factories per application server to point to the local get and put queue managers.

Business Process Choreographer database

Hosting the database on a dedicated server, preferably one with a hot standby is recommended. The database can be on a server that is outside the WebSphere cell, however the deployment manager must have access to it.

When planning the database, consider the following points:

- All business process containers in the same WebSphere cluster access the same database. By contrast, any business process container that is not in a WebSphere cluster must have its own database.
- To enable access to a remote Business Process Choreographer database, you must install the appropriate database client, or a type-4 Java Database Connectivity (JDBC) driver on all application servers that do not have a local database.
- The deployment manager requires access to all databases for Business Process Choreographer instances in the WebSphere cell, regardless of whether they are in a cluster, or not. You must enable this access before you can use the deployment manager to install a business process.
- Your database can be any of the supported databases except Cloudscape, because Cloudscape Network Server has no XA support and embedded Cloudscape cannot be accessed remotely.
- Each database that is used by Business Process Choreographer instances in the same WebSphere cell must be accessible using a unique name. For DB2, the same database name must be used on the deployment manager and on the application server.
- The database is a single point of failure. This problem can be solved only by using a high-availability hot standby solution, such as High Availability Cluster Multiprocessing (HACMP™) on AIX®.

WebSphere default messaging JMS provider

Business Process Choreographer can use WebSphere default messaging, which supports clustering, workload management, and failover.

Two topologies are supported:

- The messaging resources are hosted by a different cluster than the applications. This is the recommended topology since it provides failover capabilities together with low administration overheads. This topology is similar to the central queue manager approach in the WebSphere MQ case.
- The messaging resources and the applications are hosted by the same cluster. This topology is ideal for high performance, though it requires more administration effort, especially when changes are applied.

For more information about considerations that apply when default messaging is used, refer to ″Creating a clustered environment″ in the PDF for Administering.

WebSphere MQ

Business Process Choreographer can use WebSphere MQ queues for receiving requests and sending replies. WebSphere MQ is not recommended as the JMS provider if a clustered scenario is used. If you use WebSphere MQ, you must still configure the default messaging for the Service Component Architecture (SCA),

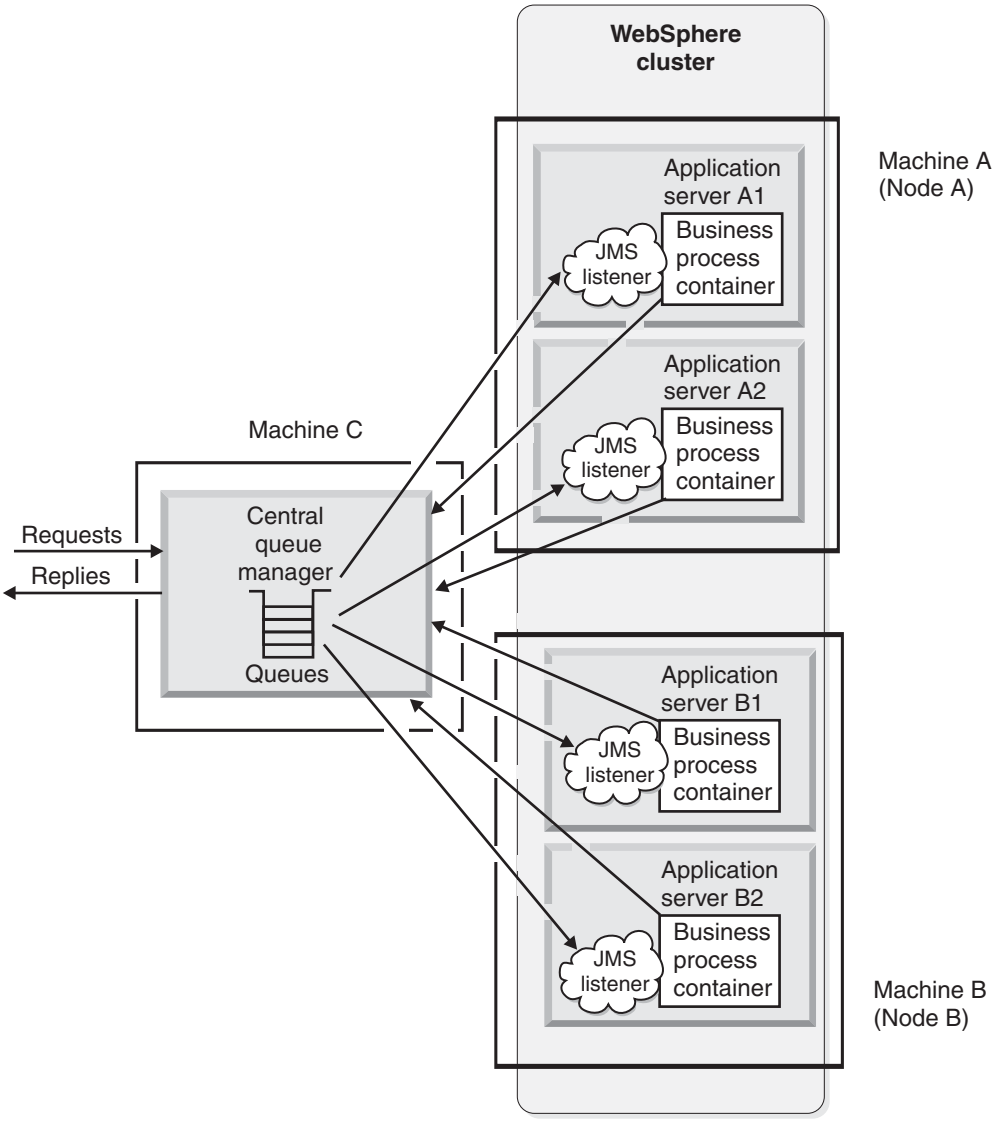
which Business Process Choreographer uses for inbound and outbound service invocation. Each application server that hosts Business Process Choreographer requires one of the following options:

- Access to a central queue manager that hosts all queues
- A local queue manager that is not a member of a WebSphere MQ cluster
- Two local queue managers that are members of a WebSphere MQ cluster

Central queue manager

By using a central queue manager for all queues, administration becomes easier. One queue manager is used by all cloned containers for human tasks and business processes. However, using a central queue manager creates a single point of failure that needs to be hosted on a high availability system.

The following figure shows all the application servers in a WebSphere cluster using a single central queue manager on another server. Every application server shown with a business process container, can also have a human task container.



Local queue manager without WebSphere MQ clustering

This example presents the standard, stand-alone Business Process Choreographer configuration. Each business process container has one local queue manager. This approach does not offer intraprocess workload sharing nor failover service availability.

WebSphere MQ clustering

This complex technique is not recommended. It supports intraprocess workload sharing for Business Process Choreographer services in a WebSphere cluster. The business processes in the cluster must all run on UNIX® only, or Windows® workstations only; a combination of UNIX and Windows servers does not work.

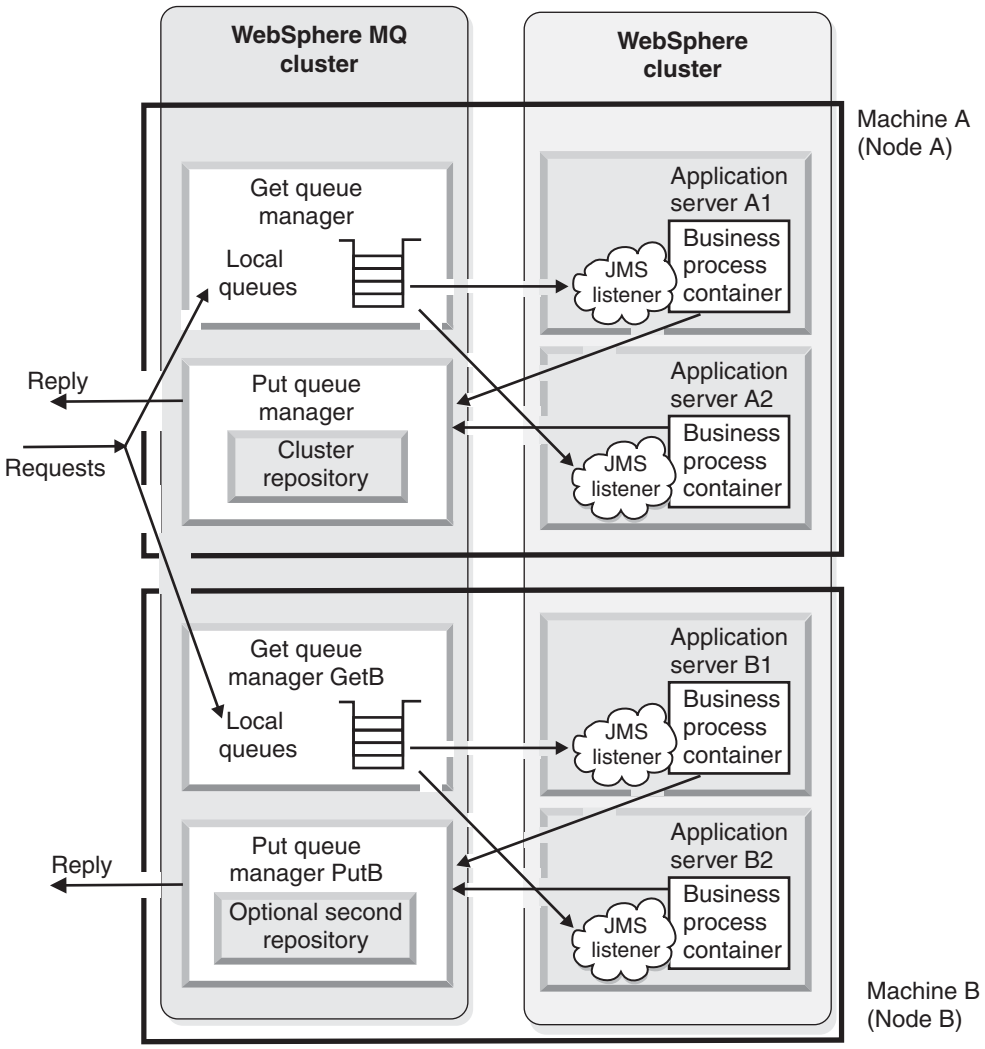
Each application server requires two local queue managers, one for putting and one for getting messages. All the queue managers become members of the same WebSphere MQ cluster. On Windows systems, all the queue managers must use the same binding protocol. On UNIX systems, the put and get queue managers must use different protocols. For example, you can modify the queue connection factories so that all the put queue managers use the binding protocol (interprocess communications) and all the get queue managers use the default, client (TCP/IP) protocol.

On Windows and UNIX systems, using the local bindings transport type is approximately 5% faster than using the client transport type, but has the effect that you must stop the entire application server to stop the local WebSphere MQ queue manager.

Each business process container in the WebSphere cluster must be customized to reflect its own queue managers.

It is recommended that more than one queue manager in the WebSphere MQ cluster is made a cluster repository.

The following figure shows how the queue managers that are used by the application servers are grouped together in a WebSphere MQ cluster. The WebSphere MQ cluster of queue managers is parallel to the WebSphere cluster of application servers. Requests are shared across the get queues in the cluster.



How the WebSphere cluster is created

Several different sequences are available for you to follow when creating a cluster for Business Process Choreographer. The following sequence is recommended:

1. Create a cluster using the defaultProcessServer template as the server template for the cluster members.
2. Add members to the cluster.
3. Enable the cluster for service applications.
4. Configure Business Process Choreographer on the cluster.
5. If you are using WebSphere MQ, and your WebSphere MQ configuration is a WebSphere MQ cluster of local queue managers, you must modify the connection factories. Because each queue manager has a different name, you must modify the connection factories in each of the cloned application servers to reflect its unique differences from the cluster-wide, standard Business Process Choreographer Install wizard configuration.

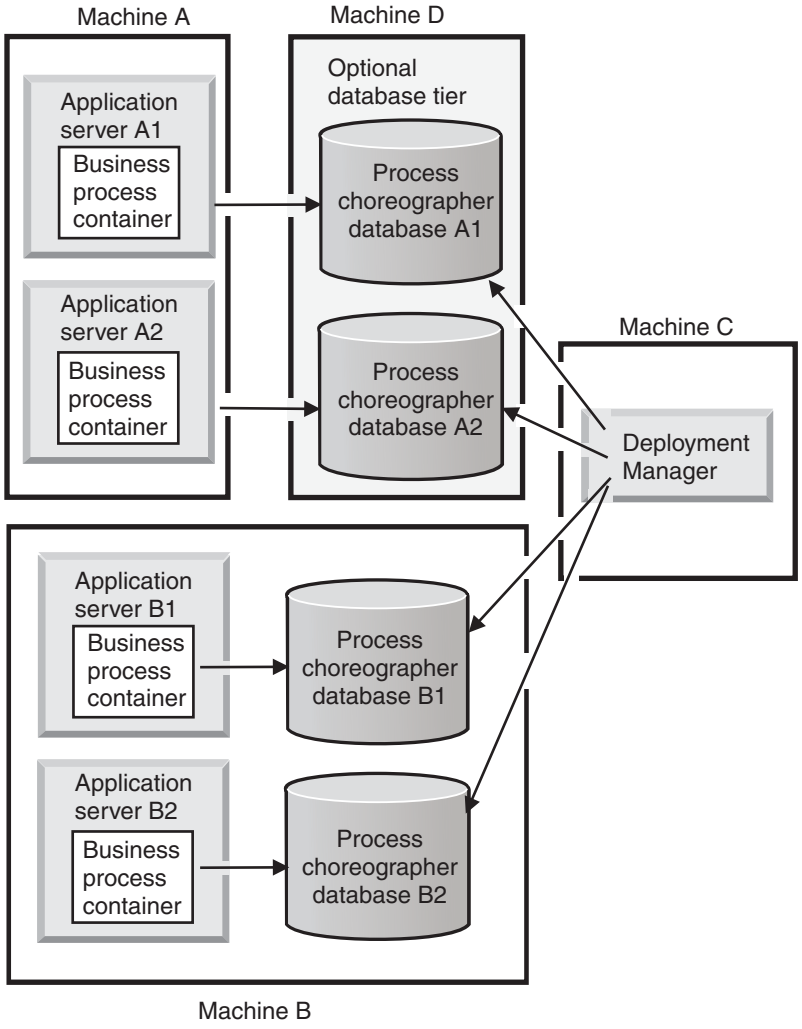
Business Process Choreographer and Network Deployment

Describes special considerations when using Business Process Choreographer in a Network Deployment environment.

If you use Network Deployment (ND), the following points must be considered.

Deployment manager must have access to the Business Process Choreographer databases

The deployment manager must have access to all the Business Process Choreographer databases that are used by business process containers and human task containers in the cell. You must install an appropriate database client or JDBC driver on the computer that hosts the deployment manager and add the database driver to the deployment manager class path. The following figure shows this configuration.



Before you install Business Process Choreographer on a cluster

Make sure that you have prepared your cluster as described in the PDF for Administering.

Customization required after installing and configuring Business Process Choreographer on a cluster

If you are creating a clustered setup that uses WebSphere MQ clusters of queue managers, you must perform some manual customization to make each Business Process Choreographer instance use its own queue managers. The necessary actions are described in Configuring the business process container.

For more information about using clustering with Business Process Choreographer, see Business Process Choreographer scenarios for clustering

Before installing an application that contains business processes, or human tasks, or both

Make sure that the servers where you want to install the application are running before installing the application. At least one server must be running for Java Naming and Directory Interface (JNDI) names to be resolved.

Planning to use the Common Event Infrastructure

The Common Event Infrastructure facilitates events.

The Common Event Infrastructure provides facilities for the generation, propagation, persistence, and consumption of events, but it does not define the actual events. When you plan how to use the event infrastructure in your system design, you need to understand the business concepts that are relevant, and map them to the appropriate components of your system design. You should provide the semantics of event management by defining event types and event groups, in the context of an architecture of event sources and event consumers.

1. Identify each *event source*. The event source is the application that creates the event. The event source passes the event object to the event infrastructure. The event infrastructure also stores the event object in a database for later retrieval. The role of the event infrastructure is to pass the event object onto any applications that express an interest in receiving it.
2. Identify each *event consumer*. An event consumer is an application that can use the information that is contained in the event object. Event consumers typically process events from a number of event sources.
3. Identify the hierarchy of the *event correlation spheres* and the identifiers for these spheres. Event consumers can use event correlation spheres to correlate events. The ECSEmitter class supports a hierarchy of correlation spheres by storing the current identifier and the parent identifier of the correlation spheres of an event in each event.

   Note: ECSEmitter and correlation sphere capabilities are provided through the Events service and not through the Common Event Infrastructure, itself.

   For example, a Business Process Execution Language (BPEL) activity opens a correlation sphere for the current activity that identifies the activity with the activity instance ID. The parent correlation sphere is the correlation sphere of the process instance on behalf of which the activity is run. The parent correlation sphere is identified by the process instance ID.
4. Identify each *event group*. An event group defines the characteristics (property values) that all events of interest to a particular type of consumer can contain. Policies, such as access controls and distribution rules are assigned to the event groups to customize the behavior of the event infrastructure for each user group.

WebSphere supplies a default event group that is defined to include all events. This event group is called *All events*.

The following figure shows the relationship between these objects:

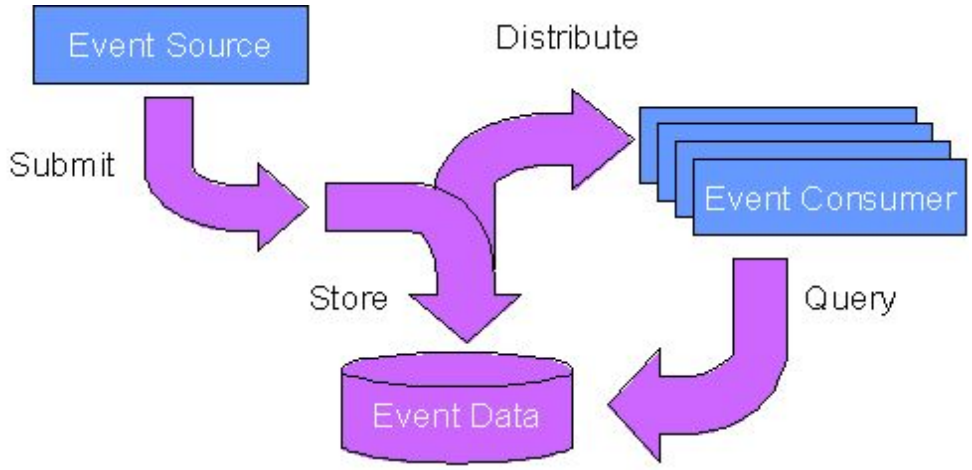


Figure 5. The architecture of an event source (which creates events), an event consumer (which makes use of the event data), and an event group (which defines the characteristics and associated policies for each type of event).

Running the install script

Run the install script to install and create the required WebSphere definitions for WebSphere Process Server for z/OS.

You run the WebSphere Process Server for z/OS install script from a command line.

You run the WebSphere Process Server for z/OS install script *after the system programmer unloads the product code successfully using SMP/E.*

Running the install script invokes Ant script actions that update the administrative console with theWebSphere Process Server for z/OS product definitions. For detailed description of install actions invoked when you run the install script, see About the install script.

The Ant script actions associated with the install script are "configuration neutral". In other words you do not run the install script for a specific configuration type.

After you run the install script, you determine how to configure WebSphere Process Server for z/OS.

Installing WebSphere Process Server for z/OS product definitions

Run **zSMPInstall.sh** to create and install the WebSphere Process Server for z/OS product definitions.

Running the installation script for WebSphere Process Server for z/OS requires a working knowledge of z/OS UNIX System Services. You may want to have access to the following documentation:

- z/OS V1R7.0 UNIX System Services User's Guide
- z/OS V1R7.0 UNIX System Services Command Reference

Before you run **zSMPInstall.sh** to enable WebSphere Process Server for z/OS for use, make sure of the following:

1. Review Task overview: installing for an understanding of the task flow of the installation process.
2. That the system programmer who is in charge of security and system access at your installation has set you up so that you can properly access the shell.

   This involves making modifications to your RACF (security) profile and creating a home directory within the UNIX shell. The home directory is where you always start off when you begin a UNIX session. Within this directory, you keep any environmental variable files you may need to execute programs. These files contain information that's required in certain languages, like the location of the Java classes that are used in Java programs. You can also use the home directory as the root directory for keeping your work data. See Resource Access Control Facility Tools in the WebSphere Application Server for z/OS for information on tools and techniques to help you manage the security definitions used for WebSphere Application Server for z/OS.
3. The WebSphere Application Server for z/OS is installed and customized.

   The system programmer has completed the SMP/E installation for WebSphere Application Server for z/OS and has performed the required customization tasks and has run the resulting configuration jobs to create a configuration type (stand-alone, network deployment or managed node).

   Note: The WebSphere Process Server for z/OS configuration process supports all of the configuration types (stand-alone, network deployment or managed node). See Planning your configuration for more information.
4. The WebSphere Process Server for z/OS product code has been loaded onto the system.

   The system programmer has completed the SMP/E installation by successfully loading the contents of the WebSphere Process Server for z/OS installation media onto the system.

Perform this task when your goal is to create the required WebSphere Process Server for z/OS definitions.

1. Create the WebSphere Application Server for z/OS configuration onto which you will install WebSphere Process Server for z/OS Follow the directions documented in the WebSphere Process Server for z/OS information center for the type of application serving environment you want to configure. Your choices are as follows:
   - Creating a stand-alone application server cell
   - Creating a Network Deployment cell
   - Creating an empty managed node

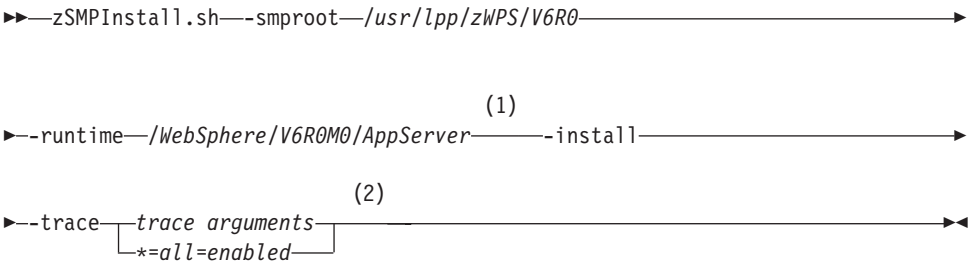
     If you are installing WebSphere Process Server for z/OS into a managed node, make sure that when create the empty managed node in WebSphere Application Server for z/OS, **that you do not run the job BBOWMNAN**. This job will federate your node into the specified Deployment Manager cell. You will federate the Deployment Manager cell as part of the WebSphere Process Server for z/OS configuration process.
2. Run the WebSphere Process Server for z/OS installation script. Enter the TSO command OMVS at the ISPF Command Shell or TSO OMVS from any other ISPF panel.

   Once you're in the UNIX shell, a command prompt (usually a dollar ($) or pound (#) sign) indicates that the system is ready to accept input.

From the command prompt, run the installation command.

Use the following syntax diagram as guide on how run the command:

Syntax diagram for installing WebSphere Process Server for z/OS product definitions



Notes:

1 The keyword parameters for -smproot and -runtime are user defined and may vary from the values shown in this example. The keyword parameter value for -smproot represents the Hierarchical File System (HFS) directory that resulted from the SMP/E installation. The keyword parameter value for -runtime represents configuration root ($WAS_Home) of WebSphere Application Server for z/OS. Type the keyword parameter that applies to your installation environment. The -runtime path must match what was set when you created your WebSphere Application Server for z/OS application serving environment.

2 By including **-trace** the command writes to the appropriate trace file.

Note: The instructions described above indicate how to run the installation script from a OS/390 UNIX command shell. Alternatively you can run the installation script from a prompt using Telnet. To run the installation script using Telnet, type the following from the system prompt: **telnet TCPIP-Address port number** and run the command as indicated in the instructions above. Here is an example of a Telnet command: `telnet 99.9.99.999 2023`, where 99.9.99.999 is the TCP IP-Address and 2023 is the port number.

Review the messages from Standard Out. These are either displayed on the screen from which you ran **zSMPInstall.sh**, or in the file specified if the redirect **(">")** symbol was used on the command line.

There should be no error messages displayed.

An example of a successful execution of **zSMPInstall.sh** is as follows:

```
parsing command arguments...
parsing arguments complete
setting up configuration...
runtimeRootDirName is: /WebSphere/V6R0M0/AppServer
WAS_HOME is: /WebSphere/V6R0M0/AppServer
WBI_HOME is: /WebSphere/V6R0M0/AppServer
set up configuration complete
creating the symbolic links...
invokeSymLink
creation of symbolic links complete
```

```
doing post install file updates...
post install updates complete
running Configuration Manager update...
Configuration Manager update complete
```

If the product did not install successfully, see “Troubleshooting the install” on page 245 for information on how to assess installation problems.

After you run the installation script successfully, you can configure the product. For more information, see Planning your configuration.

Running the installation verification test

Verify that the installation script for WebSphere Process Server for z/OS was successful.

Become familiar with the

Run the installation verification test

This is the same test used to verify the WebSphere Application Server for z/OS. There are two methods from which you can choose:

- Running the Installation Verification Test with a job
- Running the Installation Verification Test from a command line

For information on running the WebSphere Application Server for z/OS installation verification test, see Using the Installation Verification Test in the WebSphere Application Server for z/OS information center.

Perform visual confirmation of the WebSphere Process Server for z/OS installation

The WebSphere Process Server for z/OS install results in extending WebSphere Application Server for z/OS. You can confirm WebSphere Process Server for z/OS functionality via the administrative console.

Coexisting

Coexistence is the ability of two or more entities to function in the same system or network.

Coexistence does not imply the interoperability of these entities. As defined here, coexistence refers to a single environment in which a number of different WebSphere server configurations, which may involve different versions of the same product, coexist.

Coexistence support

Supporting coexistence means that you should be able to install, configure, run and manage coexisting products within your system.

Coexisting WebSphere server configurations can include the following:

- WebSphere Application Server for z/OS
- WebSphere Process Server for z/OS
- WebSphere Process Server configured as an ESB-only server.

When setting up your system for coexistence, you need to be aware of the following points:

- **Dependency on WebSphere Application Server for z/OS**

  Because WebSphere Process Server for z/OS installs on top of WebSphere Application Server for z/OS, it is a prerequisite that each server initially be configured for WebSphere Application Server for z/OS and be compliant with WebSphere Application Server for z/OS coexistence prior to installing and configuring the WebSphere Process Server for z/OS. For more information, see Coexistence support as described in the WebSphere Application Server for z/OS Information Center.
- **Version coexistence**

  You can have previous versions of the product on your system at the same, but they must exist in separate LPARs.

  WebSphere Process Server for z/OS can not be installed on top of a WebSphere Business Integration Server Foundation for z/OS server.
- **Installing for coexistence**

  The following installation scenarios for coexistence are supported:
  - WebSphere Application Server for z/OS, WebSphere Process Server and WebSphere ESB coexisting where none of the products are currently installed.
  - Coexisting WebSphere Process Server for z/OS and WebSphere Application Server for z/OS when WebSphere Application Server for z/OS already exists
  - Coexisting WebSphere Process Server for z/OS and WebSphere Process Server for z/OS configured as an ESB-only server in an existing WebSphere Application Server for z/OS

Avoiding port conflicts

A major consideration in coexistence is the avoidance of port conflicts.

Review the port number settings, especially when you are planning to coexist.

You must prevent port conflicts from occurring when you have WebSphere Process Server for z/OS coexisting on the same server with another installation of itself or of WebSphere Enterprise Service Bus, WebSphere Business Integration Server Foundation, andWebSphere Application Server for z/OS.

Use this procedure to avoid port conflicts when adding a WebSphere Process Server for z/OS node to a WebSphere Process Server for z/OS deployment manager cell when a WebSphere Enterprise Service Bus, WebSphere Business Integration Server Foundation for z/OS, and WebSphere Application Server for z/OS node exists on the same system.

Perform the steps in this task as part of your strategy to avoid port conflicts for coexisting products.

1. Determine which products are coexisting. For example, if you plan to use a single z/OS system to contain more than one WebSphere Process Server for z/OS instance, you must configure port allocations of the second and later WebSphere Process Server for z/OS instances to avoid port conflicts with the first instance of WebSphere Process Server for z/OS already installed.
2. Configure port allocations To configure the port allocation, on the Administrative console of WebSphere Application Server for z/OS, navigate to Servers > Application Servers > YourProcessServerName >End Points.

Port number settings in versions of WebSphere Process Server, WebSphere Enterprise Service Bus, WebSphere Application Server, WebSphere Application Server Network Deployment, and WebSphere Business Integration Server Foundation

You can prevent port conflicts from occurring when you want an installation of WebSphere Process Server to coexist with another installation of WebSphere Process Server, or with an installation of WebSphere Enterprise Service Bus, WebSphere Application Server, WebSphere Application Server Network Deployment, or WebSphere Business Integration Server Foundation. This topic provides reference information about identifying port numbers for these products.

Port numbers for WebSphere Process Server, WebSphere Enterprise Service Bus, and WebSphere Application Server and WebSphere Application Server Network Deployment, versions 6.0.0.x, 6.0.1.x, and 6.0.2.x

Table 11. Port definitions for WebSphere Process Server, WebSphere Enterprise Service Bus, and WebSphere Application Server and WebSphere Application Server Network Deployment, versions 6.0.0.x, 6.0.1.x, and 6.0.2.x

Port	**WebSphere Process Server or WebSphere Enterprise Service Bus (with underlying WebSphere Application Server Network Deployment)**	**WebSphere Application Server**	**File**
	Value	**Value**	
HTTP_TRANSPORT	9080	9080	serverindex.xml and virtualhosts.xml
HTTP Admin Console Port (HTTP_TRANSPORT_ADMIN)	9060	9060	serverindex.xml and virtualhosts.xml
HTTPS Transport Port (HTTPS_TRANSPORT)	9443	9443	serverindex.xml and virtualhosts.xml
HTTPS Admin Console Secure Port (HTTPS_TRANSPORT_ADMIN)	9043	9043	serverindex.xml and virtualhosts.xml
BOOTSTRAP_ADDRESS	9809	2809	serverindex.xml
SOAP_CONNECTOR_ADDRESS	8879	8880	serverindex.xml
SAS_SSL_SERVERAUTH_LISTENER_ ADDRESS A character space was added to this entry to enable it to fit in the table cell. The actual entry does not include a character space.	9401	9401	serverindex.xml
CSIV2_SSL_SERVERAUTH_ LISTENER_ADDRESS A character space was added to this entry to enable it to fit in the table cell. The actual entry does not include a character space.	9403	9403	serverindex.xml

Table 11. Port definitions for WebSphere Process Server, WebSphere Enterprise Service Bus, and WebSphere Application Server and WebSphere Application Server Network Deployment, versions 6.0.0.x, 6.0.1.x, and 6.0.2.x (continued)

Port	**WebSphere Process Server or WebSphere Enterprise Service Bus (with underlying WebSphere Application Server Network Deployment)**	**WebSphere Application Server**	**File**
	Value	**Value**	
CSIV2_SSL_MULTIAUTH_LISTENER_ADDRESS A character space was added to this entry to enable it to fit in the table cell. The actual entry does not include a character space.	9402	9402	serverindex.xml
ORB_LISTENER_ADDRESS	9100	9100	serverindex.xml
DCS_UNICAST_ADDRESS	9352	9353	serverindex.xml
SIB_ENDPOINT_ADDRESS	7276	7276	serverindex.xml
SIB_ENDPOINT_SECURE_ADDRESS	7286	7286	serverindex.xml
SIB_MQ_ENDPOINT_ADDRESS	5558	5558	serverindex.xml
SIB_MQ_ENDPOINT_SECURE_ADDRESS A character space was added to this entry to enable it to fit in the table cell. The actual entry does not include a character space.	5578	5578	serverindex.xml
Internal JMS Server (JMSSERVER_SECURITY_PORT)	Not applicable	5557	serverindex.xml
DRS_CLIENT_ADDRESS **Deprecation:** This port is deprecated and is no longer used in the current version of WebSphere Application Server.	7989	7873	serverindex.xml
IBM HTTP Server Port	80	80	virtualhosts.xml, plugin-cfg.xml, and *IHSinstall_root*/conf/httpd.conf
IBM HTTPS Server Admin Port	Not applicable	8008	*IHSinstall_root*/conf/admin.conf
CELL_DISCOVERY_ADDRESS	7277	Not applicable	serverindex.xml
CELL_MULTICAST_DISCOVERY_ADDRESS A character space was added to this entry to enable it to fit in the table cell. The actual entry does not include a character space.	7272	Not applicable	serverindex.xml

Table 11. Port definitions for WebSphere Process Server, WebSphere Enterprise Service Bus, and WebSphere Application Server and WebSphere Application Server Network Deployment, versions 6.0.0.x, 6.0.1.x, and 6.0.2.x (continued)

Port	WebSphere Process Server or WebSphere Enterprise Service Bus (with underlying WebSphere Application Server Network Deployment)	WebSphere Application Server	File
	Value	Value	
NODE_MULTICAST_IPV6_ DISCOVERY_ADDRESS A character space was added to this entry to enable it to fit in the table cell. The actual entry does not include a character space.	5001	5001	serverindex.xml

When you federate a node into a deployment manager cell, the deployment manager instantiates the nodeagent server process on the process server node. The nodeagent server uses these port assignments by default:

Table 12. Port definitions for the WebSphere Process Server nodeagent server process

Port	Value	File
BOOTSTRAP_ADDRESS	2089	serverindex.xml
ORB_LISTENER_ADDRESS	9100	serverindex.xml
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS	9901	serverindex.xml
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS	9202	serverindex.xml
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS	9201	serverindex.xml
NODE_DISCOVERY_ADDRESS	7272	serverindex.xml
NODE_MULTICAST_DISCOVERY_ADDRESS	5000	serverindex.xml
NODE_IPV6_MULTICAST_DISCOVERY_ADDRESS	5001	serverindex.xml
DCS_UNICAST_ADDRESS	9353	serverindex.xml
DRS_CLIENT_ADDRESS	7888	serverindex.xml
SOAP_CONNECTOR_ADDRESS	8878	serverindex.xml

Port numbers for WebSphere Application Server and WebSphere Application Server Network Deployment, version 5.1.x, and WebSphere Business Integration Server Foundation, version 5.1.x

WebSphere Business Integration Server Foundation uses the same ports as the product it extends. Therefore, if it extends WebSphere Application Server Network Deployment, use the values under that column in Table 13 on page 97. If it extends WebSphere Application Server, use the values under that column.

Table 13. Port definitions for WebSphere Application Server and WebSphere Application Server Network Deployment, version 5.1.x, and WebSphere Business Integration Server Foundation, version 5.1.x

Port	WebSphere Application Server Network Deployment	WebSphere Application Server	File
	Value	**Value**	
HTTP_TRANSPORT	Not applicable	9080	serverindex.xml and virtualhosts.xml
HTTPS Transport Port (HTTPS_TRANSPORT)	Not applicable	9443	serverindex.xml and virtualhosts.xml
HTTP Admin Console Port (HTTP_TRANSPORT_ADMIN)	9090	9090	serverindex.xml and virtualhosts.xml
HTTPS Admin Console Secure Port (HTTPS_TRANSPORT_ADMIN)	9043	9043	serverindex.xml and virtualhosts.xml
Internal JMS Server (JMSSERVER_SECURITY_PORT)	Not applicable	5557	server.xml
JMSSERVER_QUEUED_ADDRESS	Not applicable	5558	serverindex.xml
JMSSERVER_DIRECT_ADDRESS	Not applicable	5559	serverindex.xml
BOOTSTRAP_ADDRESS	9809	2809	serverindex.xml
SOAP_CONNECTOR_ADDRESS	8879	8880	serverindex.xml
DRS_CLIENT_ADDRESS	7989	7873	serverindex.xml
SAS_SSL_SERVERAUTH_LISTENER_ ADDRESS A character space was added to this entry to enable it to fit in the table cell. The actual entry does not include a character space.	9401	0	serverindex.xml
CSIV2_SSL_SERVERAUTH_ LISTENER_ADDRESS A character space was added to this entry to enable it to fit in the table cell. The actual entry does not include a character space.	9403	0	serverindex.xml
CSIV2_SSL_MULTIAUTH_LISTENER_ ADDRESS A character space was added to this entry to enable it to fit in the table cell. The actual entry does not include a character space.	9402	0	serverindex.xml
IBM HTTP Server Port	Not applicable	80	virtualhosts.xml, plugin-cfg.xml, and *IHSinstall_root*/conf/ httpd.conf
IBM HTTPS Server Admin Port	Not applicable	8008	*IHSinstall_root*/conf/ admin.conf
CELL_DISCOVERY_ADDRESS	7277	Not applicable	serverindex.xml
ORB_LISTENER_ADDRESS	9100	9100	serverindex.xml

Table 13. Port definitions for WebSphere Application Server and WebSphere Application Server Network Deployment, version 5.1.x, and WebSphere Business Integration Server Foundation, version 5.1.x (continued)

Port	WebSphere Application Server Network Deployment	WebSphere Application Server	File
	Value	**Value**	
CELL_MULTICAST_DISCOVERY_ ADDRESS A character space was added to this entry to enable it to fit in the table cell. The actual entry does not include a character space.	7272	Not applicable	serverindex.xml

When you federate a WebSphere Application Server or WebSphere Application Server Network Deployment, version 5.1.x application server node into a deployment manager cell, the deployment manager instantiates the nodeagent server process on the application server node. The nodeagent server uses these port assignments by default:

Table 14. Port definitions for the nodeagent server process of WebSphere Application Server and WebSphere Application Server Network Deployment, version 5.1.x, and WebSphere Business Integration Server Foundation, version 5.1.x

Port	Value	File
BOOTSTRAP_ADDRESS	2089	serverindex.xml
ORB_LISTENER_ADDRESS	9900	serverindex.xml
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS	9901	serverindex.xml
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS	9101	serverindex.xml
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS	9201	serverindex.xml
NODE_DISCOVERY_ADDRESS	7272	serverindex.xml
NODE_MULTICAST_DISCOVERY_ADDRESS	5000	serverindex.xml
DRS_CLIENT_ADDRESS	7888	serverindex.xml
SOAP_CONNECTOR_ADDRESS	8878	serverindex.xml

Planning your configuration

This task helps you plan your WebSphere Process Server for z/OS application serving environment for your z/OS target systems.

- Complete the steps in Installing the product and Preparing the base operating system.
- Read the section WebSphere Process Server for z/OS install terminology.
- If you already have any release of WebSphere Process Server for z/OS running on your z/OS target system, review Coexisting.
- Complete the steps for planning to run the install script and the steps for running the install script.

WebSphere Process Server for z/OS operates in a WebSphere Process Server for z/OS "application serving environment." Configuring an application serving environment after product installation requires a fair amount of planning and coordination.

If you have not previously configured WebSphere Process Server for z/OS, it is recommended that you configure a *practice* stand-alone process server using the default options from the stand-alone response file (StandAloneProfile.rsp) then proceed to configure the actual product configuration that you want.

Before you configure WebSphere Process Server for z/OS you will need to create the databases and database storage groups for the WebSphere Process Server for z/OS components that utilize database. See Creating databases and storage groups for details.

1. Decide whether to set up a stand-alone application server or a network deployment cell. See ″Choosing between a stand-alone or network deployment configuration″ topic in the WebSphere Application Server for z/OS documentation infocenter in the WebSphere Application Server for z/OS information center for more information.
2. Decide which functionality to include in the WebSphere Process Server for z/OS configuration. You can choose to configure WebSphere Process Server for z/OS as a Process server or as an ESB-only server.
3. Determine which response file to include in your product configuration command. WebSphere Process Server for z/OS includes default response files for the following product configurations:
   - Stand-alone configuration using Cloudscape

     This configuration is recommended for development/test environment only. It is likely not suitable for a production environment.
   - Stand-alone configuration using DB2 for z/OS version 7
   - Stand-alone configuration using DB2 for z/OS version 8
   - Network deployment configuration using DB2 for z/OS version 7
   - Network deployment configuration using DB2 for z/OS version 8
   - Managed node configuration using DB2 for z/OS version 7
   - Managed node configuration using DB2 for z/OS version 8
4. Determine where on your system to copy the default response file. The default response files reside in the install HFS, in a directory path like `/usr/lpp/zWPS/V6R0/zos.config/standAloneProfile.rsp`.

   You will need to copy the response file from the HFS directory to a directory on your system (you can transfer the file from the host to a local directory on your PC). Make sure that the response file is in EBCDIC when you return it to the z/OS system.
5. Determine which of the property values in the chosen response file need to be customized to suit your environment and edit the property values accordingly. The response file is keyword parameter in the configuration command.

   Note: Some of the database and storage group property values in the response file must match database and storage group property values that are set when you create the databases and storage groups. When this is the case, it is noted in the sample response files that are documented in Sample response files.
6. Determine if you will need to perform post-configuration tasks. Some of the property values that you set in the response file will be a determining factor with regard to the post configuration tasks that you will need to perform.

   The sample response files that are included in the Sample response files topic include comments that indicate which property value settings will result in post configuration tasks.

When you have completed the planning for the configuration that you have selected, you are ready to configure the application serving environment.

Considerations for creating the database

WebSphere Process Server for z/OS product configurations include features that utilize databases.

As part of your WebSphere Process Server for z/OS configuration planning, you create the database and the database storage groups required by WebSphere Process Server for z/OS components that utilize databases.

The WebSphere Process Server for z/OS component features that utilize databases include the following:

- Common Event Infrastructure

  The Common Event Infrastructure utilizes two databases. An event database and an event catalog database

  This means that you will be required to create the database and storage groups for both of these databases.
- Business Process Choreographer
- Components that share a common database. These common components include the following:
  - Relationship services
  - Mediation
  - Recovery
  - Business rules
  - Selector
  - Scheduler

The WebSphere Process Server for z/OS product package includes a sample file with default values for the database and database storage groups that you need to create prior to running the WebSphere Process Server for z/OS configuration script.

You can tailor values in the sample and use database creation utility to create the database and storage groups.

For detailed information, see Creating databases and database storage groups.

Creating databases and storage groups

Create the database and storage group into which you will configure WebSphere Process Server for z/OS

The WebSphere Process Server for z/OS includes a sample file that you can use to create the database and storage groups. You must have successfully installed WebSphere Process Server for z/OS in order to access the sample file.

You must create the databases and storage groups utilized by WebSphere Process Server for z/OS prior to running the configuration script. When you run the configuration script, the DDL and SQL for the database tables will be created. The databases to hold those tables must exist as a context for creating the DDL and SQL for the tables.

1. Access the sample file from the install HFS. Go to the directory containing the sample database configuration for WebSphere Process Server for z/OS. This file is located in the following directory:

   install_root/zWPS/V6R0/zos.config/wps_DB_StorGrp.sql

   For example, /usr/lpp/zWPS/V6R0/zos.config/wps_DB_StorGrp.sql
2. Review and, if necessary, edit the values in the file to suit your needs Change the database names and the storage group names to meet your naming requirements.

   Note: The sample file is delivered in ASCII format. Depending on the capabilities of the tool you use to view, edit and run this file, you may need to convert the file to a readable format, EBCDIC for example.

   See the example section at the end of this file for sample code for each of the databases as it is contained in the sample file.

   Note: The names that you specify in the sample file need to be carried forward as values in the response file that is referenced in the product configuration script.
3. Create the database and storage groups. Use your site's standard database definition tools and procedures to create the database and storage groups.

The following is an example of the default database and storage group names packaged in the sample file.

```
-- ################################################################################
-- # Licensed Materials - Property of IBM
-- # 5655-N53, 5655-R15
-- # (C) Copyright IBM Corporation 2006. All Rights Reserved.
-- # US Government Users Restricted Rights - Use, duplication or disclosure
-- # restricted by GSA ADP Schedule Contract with IBM Corp.
-- ################################################################################
SET CURRENT SQLID = 'WSADMIN';

DROP   DATABASE WPSDB ;
DROP   STOGROUP WPSDBSTO;
CREATE STOGROUP WPSDBSTO VOLUMES (WBIUS3,WBIUSR,WBIUS5) VCAT WSDB2;
CREATE DATABASE WPSDB
  STOGROUP WPSDBSTO
  BUFFERPOOL BP0
  INDEXBP BP0;
COMMIT;

DROP   DATABASE BPEDB ;
DROP   STOGROUP BPEDBSTO;
CREATE STOGROUP BPEDBSTO VOLUMES (WBIUS3,WBIUS4,WBIUS5) VCAT WSDB2;
CREATE DATABASE BPEDB
  STOGROUP BPEDBSTO
  BUFFERPOOL BP0
  INDEXBP BP0;
COMMIT;

DROP   DATABASE EVENT ;
DROP   DATABASE EVENTCAT ;
DROP   STOGROUP EVTSTO;
CREATE STOGROUP EVTSTO VOLUMES (WBIUS3,WBIUS4,WBIUS5) VCAT WSDB2;
CREATE DATABASE EVENT
  STOGROUP EVTSTO
  BUFFERPOOL BP0
  INDEXBP BP0;
CREATE DATABASE EVENTCAT
  STOGROUP EVTSTO
  BUFFERPOOL BP0
```

```
  INDEXBP BP0;
COMMIT;

DROP   DATABASE ESBDB ;
DROP   STOGROUP ESBDBSTO;
CREATE STOGROUP ESBDBSTO VOLUMES (WBIUS3,WBIUS4,WBIUS5) VCAT WSDB2;
CREATE DATABASE ESBDB
  STOGROUP ESBDBSTO
  BUFFERPOOL BP0
  INDEXBP BP0;
COMMIT;
```

Now you are ready to edit the response files. See Working with response files for more information.

Working with response files

Response files that contain property values for default product configurations that were loaded onto the z/OS system during the SMP/E portion of the install.

The information in the response file is used by the product configuration script to augment the WebSphere Application Server for z/OS default profile with WebSphere Process Server for z/OS configuration data.

You provide the path to a response file in the configuration command when you run the configuration script. You run the configuration script after you have run the product install script successfully.

For a listing of the response files that are included with the product, see Choosing a response file. The

See Sample response files for detailed descriptions of the sample response files.

The following syntax diagram indicates where in the command to specify the response file.

Configuring WebSphere Process Server for a stand-alone configuration using DB2 for z/OS

Notes:

1 Type the absolute path of the response file that contains the properties required for your product configuration. The path shown below is the path to the default response file packaged with the product. The path name that you would type must represent the path location of the response file that you modified with property values specific to your environment.

Things to know before you use response files

Before you run the product configuration script, you should know the following characteristics of response files:

- The response files shipped with the product contain default values that you will need to edit to suit your environment.
- Before you edit any of the response files, you will need to create databases and database storage groups.

  Some of the property values that you specify when creating the database and database storage groups, must match associated values in the response file. These instances are notated in the sample response files. See Sample response files for more information.
- The properties that you set in a response file may result in you having to perform additional configuration steps.

  Read the comment sections of the response file before setting the values. The comment sections contain important information on setting property values, including descriptions of what results from setting property values.
- Be careful when you add comments to a response file.

  If you put comments in the response file, and that comment resides on the same line as a property value, even if you precede the comment with a cross hatch character (#), the Ant script attempts to read the comment as part of the parameter value, causing unpredictable results. For example, do not enter comments as shown in this example:

  ```
  ##############################################################################
  # DB2 Properties
  ##############################################################################
  dbJDBCClasspath=/shared/db2810/jcc/classes    # DB2 ClassPath Location
  dbJDBCProperties=/u/hutch/wpswork/            # DB2JccConfiguration.properties
  ```

  The correct way to add comments shown in the sample above would be as follows:

  ```
  ##############################################################################
  # DB2 Properties
  ##############################################################################
  # DB2 ClassPath Location
  dbJDBCClasspath=/shared/db2810/jcc/classes
  # DB2JccConfiguration.properties
  dbJDBCProperties=/u/hutch/wpswork/
  ```
- Be careful of ″empty spaces″ at the end of a property value.

  Embedded spaces at the end of the property in your response file can cause problems. Make sure that you do not have any additional spaces at the end of a property value in your response file.

Reading a response file

The response file contains a lot of helpful information in the form of comments.

You should print out the response file and read the commented text prior to running the product configuration script. The response file comments help you to understand how set values for your particular environment.

The top of the response file contains a section named **GLOBAL Properties**. Here is an example:

```
##############################################################################
# GLOBAL Properties
##############################################################################
JMSUSER=ibmuser
JMSPASS=ibmuser
DBUSER=wsadmin
DBPASS=wsadmin
```

```
CONFIGSERVER=server1
DBLOCATION=LOC1

################################################################################
```

The **GLOBAL Properties** section provides a central location for ″common values″. Common values are values that display in multiple places (because they are used by multiple components) throughout the response file. Providing a central location for these common values facilitates the editing process. For example, if all server properties need to be set to **server1**, the use of a global property allows you to see and, if necessary, change all instances by changing the value of the global property only. You would not be required to search the response file for all instances of the property that you want to change.

You use the GLOBAL property by specifying the global keyword prefixed with the $ symbol as the value for a subsequent property. For example, serverName=$CONFIGSERVER.

Global properties can also simplify the use of the override argument (-Z) in the product configuration command by reducing command line clutter.

The comments for a property display above the property. The property and the default value display in bold text. As shown in the following example:

```
################################################################################
#
# Profile name
#
# On z/OS, there is always one and only one profile and that profile is named
# default in each of the configurations.
#
# The profile referred to here is the default profile installed and
# and configured during the WebSphere Application Server for z/OS install.
#
profileName=default
```

Choosing the response file

The response file contains properties used to create the WebSphere Process Server for z/OS definitions and used to augment the default profile with WebSphere Process Server for z/OS configuration data.

The configuration script includes the absolute path of the response file. Choosing a response file requires an understanding of the different configurations onto which one will install WebSphere Process Server for z/OS.

The WebSphere Process Server for z/OS product code includes sample response files for each of the supported configuration types.

The sample response files were loaded onto system and are part of the install HFS that was created during the SMP/E phase of the install.

The WebSphere Process Server for z/OS product includes the following response files:

Note: The directory specified assumes user defined HFS install root of `/usr/lpp/zWPS/V6R0/usr/lpp/zWPS/V6R0/zos.config/`**`XYZ.rsp`**, where **XYZ.rsp** is the name of the response file.

- /usr/lpp/zWPS/V6R0/usr/lpp/zWPS/V6R0/zos.config/**standAloneProfile.rsp**

Use this response file for a stand-alone configuration in a development/test environment only. It includes configuration parameters for the Cloudscape database.

- /usr/lpp/zWPS/V6R0/usr/lpp/zWPS/V6R0/zos.config/**standAloneProfileDB2.rsp**

  Use this response file for a stand-alone configuration with a DB2 for z/OS database.

  This response file supports DB2 for z/OS version 7 and DB2 for z/OS version 8. The default properties assume a DB2 for z/OS version 8. If you have DB2 for z/OS version 7 installed, you will need to modify the default values in this response file to support DB2 for z/OS version 7.

  The response file contains comments on those properties that apply to DB2 for z/OS version 7 specifically.

- usr/lpp/zWPS/V6R0/usr/lpp/zWPS/V6R0/zos.config/**dmgrProfile.rsp**

  Use this response file for a network deployment configuration with a DB2 for z/OS version 8 database.

  This response file supports DB2 for z/OS version 7 and DB2 for z/OS version 8. The default properties assume a DB2 for z/OS version 8. If you have DB2 for z/OS version 7 installed, you will need to modify the default values in this response file to support DB2 for z/OS version 7.

  The response file contains comments on those properties that apply to DB2 for z/OS version 7 specifically.

- usr/lpp/zWPS/V6R0/usr/lpp/zWPS/V6R0/zos.config/**managedProfile.rsp**

  Use this response file for a network deployment configuration with a DB2 for z/OS.

  This response file supports DB2 for z/OS version 7 and DB2 for z/OS version 8. The default properties assume a DB2 for z/OS version 8. If you have DB2 for z/OS version 7 installed, you will need to modify the default values in this response file to support DB2 for z/OS version 7.

  The response file contains comments on those properties that apply to DB2 for z/OS version 7 specifically.

When you run **zWPSConfig.sh** or **zWESBConfig.sh**, the keyword -response followed by the user defined absolute path of the response file.

As part of the configuration actions, the script passes the absolute path name of the response file into the configuration manager, which in turn runs a sequence of Ant scripts to update the administrative console with WebSphere Process Server for z/OS definitions and augment the default profile.

Sample response files

WebSphere Process Server for z/OS comes with sample response files that are installed into the read-only HFS.

Purpose

As part of the configuration command, you pass in a response file associated with one of the supported configurations.

The response file contains the property values for the WebSphere Process Server for z/OS product configuration.

The following sections contain notated samples of the default response files.

Response file for standalone configuration using Cloudscape - standAloneProfile.rsp

The following is a sample of the default response file used for configuring WebSphere Process Server for z/OS in a stand-alone configuration.

Note: The sample response file below contains properties for Business Process Choreographer and WebSphere Business Integration Server. Business Process Choreographer and WebSphere Business Integration Server are not configured for an ESB-only server configuration of WebSphere Process Server for z/OS. These properties are ignored by the configuration processing.

```
################################################################################
#
# Sample Response file for WebSphere Process Server for z/OS
# Stand-alone configuration using Cloudscape database
#
################################################################################
################################################################################
# GLOBAL Properties
################################################################################
 JMSUSER=ibmuser
 JMSPASS=ibmuser
 DBUSER=wsadmin
 DBPASS=wsadmin
 CONFIGSERVER=server1
 DBLOCATION=LOC1
#
# For a description of how to use Global properties, see the section
# on "Reading the response file" in # Working with response files.
#

################################################################################
#
# Common Properties
#
################################################################################
#
# Augment
#
# The "augment" keyword is required to augment the default profile.
# There is no value associated with this keyword. The "augment" keyword
# may be replaced with "unaugment" in order to undo the changes from a
# previous augment. However, "augment" must be specified in order to
# configure the product successfully.
#
# The "augment" keyword is also used by the uninstall operation.
# See About the uninstall command for information
# on how to uninstall the product.
#
augment

################################################################################
#
# Profile name
#
# On z/OS, there is always one and only one profile and that profile is named
# default in each of the configurations.
#
# The profile referred to here is the default profile installed and
# and configured during the WebSphere Application Server for z/OS install.
#
# Never alter this value.
#
profileName=default

################################################################################
#
# Profile path
#
# This value represents the default profile root directory for
# WebSphere Application Server for z/OS.
#
```

```
# The profile path contains the files that define the run-time environment,
# such as commands, configuration files, and log files.
#
# In most cases the WebSphere Application Server install will use the default
# value shown below.

# If the default value was not used, make sure the value below matches the
# profile path specified for the WebSphere Application Server for z/OS
# installation.
#
profilePath=/WebSphere/V6R0M0/AppServer/profiles/default

##############################################################################
#
# Template path
#
# This value determines which actions are performed as part of the profile
# augmentation. Valid values are as follows:
#
# /WebSphere/V6R0M0/AppServer/profileTemplates/default.wbicore
# /WebSphere/V6R0M0/AppServer/profileTemplates/default.bfm
# /WebSphere/V6R0M0/AppServer/profileTemplates/default.wbiserver
#
# /WebSphere/V6R0M0/AppServer/profileTemplates/default.*
#
# If one of the first three values is specified, only
# the actions within that directory are performed. The fourth
# value is the default (.../default.*), which causes all of the actions in the first
# three template paths to be performed.
#
# Caution: Changing the default value causes an incomplete configuration of
# WebSphere Process Server for z/OS and should only be done if there is a valid reason
# to do so.
#
templatePath=/WebSphere/V6R0M0/AppServer/profileTemplates/default.*

##############################################################################
#
# Cell name
#
# This value determines the cell name within the WebSphere Application Server for
# which this Process Server configuration
# applies. The default value is SY1.
#
cellName=SY1

##############################################################################
#
# Node name
#
# This value determines the node name within the WebSphere Application Server for
# which this Process Server configuration
# applies. Node name under one cell has to be unique. The default value is SY1.
#
nodeName=SY1

##############################################################################
#
# Server name
#
# This value determines the server name within the WebSphere Application Server
# for which this Process Server configuration
# applies. Server name under one node has to be unique. The default value is server1.
#
serverName=server1

##############################################################################
#
# WBI Core Properties
#
##############################################################################
#
# Common Event Infrastructure Configuration
#
# The following entries are used to create a sample Common Event Infrastructure
# (CEI) configuration. If ceiSampleJmsUser, ceiSampleJmsPwd, and ceiSampleServer
# values are not provided, then the sample CEI configuration will not be created.
#
##############################################################################
```

```
#
# Java messaging service (JMS) userid for the CEI sample configuration
#
# Specifies the value of the Java Message Service (JMS) userid for a Common
# Event Infrastructure (CEI) configuration. Default value is ibmuser.
#
ceiSampleJmsUser=ibmuser

################################################################################
#
# Java messaging service (JMS) password for the CEI sample configuration
#
# Specifies the value of the Java Message Service (JMS) password for a Common
# Event Infrastructure (CEI) configuration. Default value is ibmuser.
#
ceiSampleJmsPwd=ibmuser

################################################################################
#
# Server name for the CEI sample configuration
#
# Specifies the value of the server name for the Common Event Infrastructure (CEI)
# sample configuration. The default value is server1.
#
ceiSampleServerName=server1

################################################################################
#
# Database product name for Common Event Infrastructure
#
# Specifies the default value of the database product name for Common Event
# Infrastructure (CEI) processing. The default is CLOUDSCAPE_V51_1.
#
ceiDbProduct=CLOUDSCAPE_V51_1

################################################################################
#
# Configure SCA
#
################################################################################
#
# The following entries are used to create a Service Component Architecture
# sample configuration.
#
################################################################################
#
# Configure SCA security
#
# If you do not want to create a Service Component Architecture
# sample configuration, change the default value from true to false and
# leave the values for the properties specified below blank
#
configureScaSecurity=true

################################################################################
#
# SCA security user id
#
# If you chose to create a Service Component Architecture configuration above
# then specify the userid for that configuration. Default value is ibmuser.
#
scaSecurityUserId=ibmuser

################################################################################
#
# SCA security password
#
# If you chose to create a Service Component Architecture configuration above
# then specify the password for SCA security user id for that configuration.
#
scaSecurityPassword=ibmuser

################################################################################
#
# Business Process Choreographer Properties
#
################################################################################
#
# The following entries are used to create a sample Business Process
```

```
# Choreographer (BPC) configuration. If mqUser, mqPwd, and adminBFMGroups values
# are not provided, then the sample BPC configuration is not created.
#
.
# Note: Business Process Choreographer configuration properties are ignored
# when configuring WebSphere Process Server for z/OS as an ESB-only server
#
################################################################################
#
# MQ User Name
#
# Specify the user name for the configuration of resources for the Business
# Process Container using WebSphere MQ. The default is ibmuser.
#
mqUser=ibmuser

################################################################################
#
# MQ Password
#
# Specify the password for the MQ User Name for the configuration of resources
# for the Business Process Container using WebSphere MQ. The default is ibmuser.
#
mqPwd=ibmuser

################################################################################
#
# Admin BFM Group
#
# Specify the security role for the business process system administrator. This
# is the security role used by the Business Process Choreographer to run a
# business process. The default is ibmuser.
#
adminBFMGroups=ibmuser


################################################################################
#
# WBI Server Properties
#
################################################################################
#
# Application Scheduler configuration
# configureAppScheduler
#
# Application Scheduler is used to schedule migration application group events.
# If you choose to configure Application Scheduler, set the following property
# to "true". Otherwise, specify false.
#
configureAppScheduler=true

#########################################################################
#
# Indicate the application server name on this node if you chose to configure
# it for Application Scheduler use. Usually, for a stand-alone configuration the
# server name is "server1". The default is server1.
#
appSchedulerServer=server1

################################################################################
# Configure Database Authentication for process server
# Various WebSphere Process Server components use a database connection.
# Choose a database type and enter the database configuration information
# based on that type.
.
# Note: WBI Server configuration properties are ignored
# when configuring WebSphere Process Server for z/OS as an ESB-only server
#
################################################################################
#
# Database name
#
# Specify the name of the WebSphere Process Server database. The default is
# WBIDB.
#
dbName=WBIDB

################################################################################
#
```

```
# Database type
#
# Specify the type of the WebSphere Process Server database. The default is
# Cloudscape.
#
dbType=Cloudscape

################################################################################
#
# Create new or use existing database
#
# You can choose to use an existing database for WebSphere Process Server or
# you can create a new database. To use an existing database, set the following
# property to "false". To create a new database set the following property
# to "true". The default is "true".
#
dbCreateNew=true

################################################################################
#
# Location of database server (database product installation root)
#
# Specify the database location. The database server location is the
# installation root for the database product. The default location for db type
# of cloudscape is /WebSphere/V6R0M0/AppServer/cloudscape/databases/WBIDB
#
dbLocation=/WebSphere/V6R0M0/AppServer/cloudscape/databases/WBIDB
```

Response file for standalone configuration using DB2 for z/OS - standAloneProfileDB2.rsp

The following is a sample of the default response file used for configuring WebSphere Process Server for z/OS into a stand-alone configuration with a DB2 for z/OS database.

This response file supports DB2 for z/OS version 7 and DB2 for z/OS version 8. The default properties assume a DB2 for z/OS version 8. If you have DB2 for z/OS version 7 installed, you will need to modify the default values in this response file to support DB2 for z/OS version 7.

The response file contains comments on those properties that apply to DB2 for z/OS version 7 specifically.

Note: The sample response file below contains properties for Business Process Choreographer and WebSphere Business Integration Server. Business Process Choreographer and WebSphere Business Integration Server are not configured for and ESB-only server configuration of WebSphere Process Server for z/OS. These properties are ignored by the configuration processing.

```
################################################################################
# Licensed Materials - Property of IBM
# 5655-N53, 5655-R15
# (C) Copyright IBM Corporation 2006. All Rights Reserved.
# US Government Users Restricted Rights - Use, duplication or disclosure
# restricted by GSA ADP Schedule Contract with IBM Corp.
################################################################################
################################################################################
#
# Sample Response file for WebSphere Process Server for z/OS - Stand-alone
#
# The default values for the database properties in this response file assume
# a DB2 version 8 database.
#
# The values for the following DB2 properties may differ for DB2 version 7 or
# any other version beside version 8:
#
#     bpcdbVersion    bpcdbHome     esbDbProduct     ceiDbProduct
#     ceidbHome       dbType        dbJDBCClasspath  dbJDBCProperties
################################################################################
```

```
################################################################################
#
# Sample Response file for WebSphere Process Server for z/OS - Stand-alone
# configuration using DB2 for z/OS version 8 database
#
# Sample Response file for WebSphere Process Server for z/OS
#
################################################################################
################################################################################
# GLOBAL Properties
# Global properties prepended by the symbol $ may used as substitution for
# other values in this file. The global property value must be the complete
# value to be substituted. See examples of $JMSUSER used in this file.

################################################################################
JMSUSER=ibmuser
JMSPASS=ibmuser
DBUSER=wsadmin
DBPASS=wsadmin
CONFIGSERVER=server1
DBLOCATION=LOC1
# For a description of how to use Global properties, see the section
# on "Reading the response file" in # Working with response files
################################################################################
#
# Common Properties
#
################################################################################
#
# Augment
#
# The "augment" keyword is required to augment the default profile.
# There is no value associated with this keyword. The "augment" keyword
# may be replaced with "unaugment" in order to undo the changes from a
# previous augment. However, "augment" must be specified in order to
# configure the product successfully.
#
# The field must be left as "augment" even if the shell command line
# specifies "uninstall".
#
# The "augment" keyword is also used by the uninstall operation.
# See About the uninstall command for information
# on how to uninstall the product.
#
augment

################################################################################
#
# Profile name
#
# On z/OS, there is always one and only one profile and that profile is named
# default in each of the configurations.
#
# The profile referred to here is the default profile installed and
# and configured during the WebSphere Application Server for z/OS install.
#
# Never alter this value.
#
profileName=default

################################################################################
#
# Profile path
#
# This value represents the default profile root directory for
# WebSphere Application Server for z/OS.
#
# The profile path contains the files that define the run-time environment,
# such as commands, configuration files, and log files.
#
# In most cases the WebSphere Application Server install will use the default
# value shown below.
#
# This path must match the profile path specified on the ISPF customization
# dialog when you created your application server cell.

# If the default value was not used, make sure the value below matches the
# profile path specified for the WebSphere Application Server for z/OS
# installation.
```

```
profilePath=/WebSphere/V6R0M0/AppServer/profiles/default
################################################################################
#
# Template path
#
# This value determines which actions are performed as part of the profile
# augmentation. Valid values are as follows:
#
# /WebSphere/V6R0M0/AppServer/profileTemplates/default.wbicore
# /WebSphere/V6R0M0/AppServer/profileTemplates/default.bfm
# /WebSphere/V6R0M0/AppServer/profileTemplates/default.wbiserver
#
# /WebSphere/V6R0M0/AppServer/profileTemplates/default.*
#
# If one of the first three values is specified, only
# the actions within that directory are performed. The fourth
# value is the default (.../default.*), which causes all of the actions in the first
# three template paths to be performed.
#
# Caution: Changing the default value causes an incomplete configuration of
# WebSphere Process Server for z/OS and should only be
# done if there is a valid reason to do so.
#
#
templatePath=/WebSphere/V6R0M0/AppServer/profileTemplates/default.*
################################################################################
#
# Cell name
#
# This value determines the cell name within the WebSphere Application Server for
# which this Process Server configuration
# applies. The default value is SY1.
#
# This value must match the cell name specified on the ISPF customization
# dialog when you created your stand-alone application server cell.
#
cellName=SY1

################################################################################
#
# Node name
#
# This value determines the node name within the WebSphere Application Server for
# which this Process Server applies.
# Node name under one cell has to be unique.
#
# This value must match the node name specified on the ISPF customization
# dialog when you created your stand-alone application server cell.
#
# The default value is SY1.
#
nodeName=SY1
################################################################################
#
# Server name
#
# This value determines the server name within the WebSphere Application Server
# for which this Process Server applies.
# Server name under one node has to be unique.
#
# This value must match the server name specified on the ISPF customization
# dialog when you created your stand-alone application server cell.
#
# The default value is server1.
#
serverName=$CONFIGSERVER

################################################################################
#
# WBI Core Properties
#
################################################################################
#
# Common Event Infrastruction Configuration
#
# The following entries are used to create a sample Common Event Infrastructure
# (CEI) configuration. If ceiSampleJmsUser, ceiSampleJmsPwd, and ceiSampleServer
# values are not provided, then the sample CEI configuration will not be created.
```

```
#
################################################################################
#
# Java messaging service (JMS) userid for the CEI sample configuration
#
# Specifies the value of the Java Messaging Service (JMS) userid for a Common
# Event Infrastructure (CEI) configuration. Default value is ibmuser.
#
ceiSampleJmsUser=$JMSUSER
################################################################################
#
# Java messaging service (JMS) password for the CEI sample configuration
#
# Specifies the value of the Java Messaging Service (JMS) password for a Common
# Event Infrastructure (CEI) configuration. Default value is ibmuser.
#
ceiSampleJmsPwd=$JMSPASS
################################################################################
#
# Server name for the CEI sample configuration
#
# Specifies the value of the server name for the Common Event Infrastructure (CEI)
# sample configuration. The default value is server1.
#
ceiSampleServerName=$CONFIGSERVER

################################################################################
#
# Database product name for Common Event Infrastructure
#
# Specifies the default value of the database product name for Common Event
# Infrastructure (CEI) processing. The default is DB2UDBOS390_V8_1.
#
# Valid Values are:
#            DB2UDBOS390_V8_1  for DB2 Version 8
#            DB2UDBOS390_V7_1  for DB2 Version 7
#
ceiDbProduct=DB2UDBOS390_V8_1
################################################################################
#
# Database Home Directory for Common Event Infrastructure
#
# Specify the Database Home JCC Directory. This value will vary depending on the
# DB2 Version used and where it is installed. The default value is /db2810/jcc.
#
ceidbHome=/db2810/jcc

################################################################################
#
# Database User ID for Common Event Infrastructure
#
# Specify the User ID for the configuration of the Common Event Infrastructure
# database. The default value is wsadmin.
#
ceiDbUser=$DBUSER
################################################################################
#
# Database Password for Common Event Infrastructure
#
# Specify the Password for the configuration of the Common Event Infrastructure
# database. The default value is wsadmin.
#
ceiDbPwd=$DBPASS

################################################################################
#
# Database Storage Group for Common Event Infrastructure
#
# Specify the Database Storage Group name for the configuration of the Common
# Event Infrastructure database. This value will be substituted in all CEI
# generated DDL/SQL definitions. This value must match Storage Group name used
# in the pre-Configuration Database and Storage Group creation task.
# The default value is EVTSTO.
#
#
# See Creating databases and storage groups for more information.
#
#
ceiDbStorageGroup=EVTSTO
```

```
################################################################################
#
# Configure SCA
#
################################################################################
#
# The following entries are used to create a Service Component Architecture
# sample configuration.
#
################################################################################
#
# Configure SCA security
#
# If you do not want to create a Service Component Architecture
# sample configuration, change the default value from true to false and
# leave the values for the properties specified below blank
#
configureScaSecurity=true

################################################################################
#
# SCA security user id
#
# If you chose to create a Service Component Architecture configuration above
# then specify the userid for that configuration. Default value is ibmuser.
#
scaSecurityUserId=$JMSUSER
################################################################################
#
# SCA security password
#
# If you chose to create a Service Component Architecture configuration above
# then specify the password for SCA security user id for that configuration.
#
scaSecurityPassword=$JMSPASS

################################################################################
#
# ESB Properties
#
# The following entries are used to create a Enterprise Service Bus Repository
# sample configuration.
################################################################################
#
# Database product name for Enterprise Service Bus
#
# Specify the value of the database product name for Enterprise Service
# Bus (ESB) processing. The default value is DB2UDBOS390_V8_1.
# Valid Values are:
#            DB2UDBOS390_V8_1  for DB2 Version 8
#            DB2UDBOS390_V7_1  for DB2 Version 7
#
#
esbDbProduct=DB2UDBOS390_V8_1

################################################################################
# Database name for Enterprise Server Bus
#
# Specify the name of the Enterprise Service Bus database. This value will be
# substituted in the generated DDL/SQL produced during install of ESB. This value
# must match the Database name used in the pre-Configuration Database and Storage
# Group creation task. The default value is ESBDB.

# The value that you specify below MUST MATCH the value that was assigned to the
# CREATE DATABASE property in the sample DDL used to create the database and storage
# group for Enterprise Service Bus.
#
#
# See Creating databases and storage groups for more information.
#
#
esbDbName=ESBDB

################################################################################
#
# Database Storage Group for Enterprise Service Bus
#
# Specify the Database Storage Group name for the configuration of the Enterprise
```

```
# Service Bus database. This value will be substituted in the generated DDL/SQL
# produced during the install for ESB. This value must match Storage Group name
# used in the pre-Configuration Database and Storage Group creation task.
#
# The default value is ESBDBSTO.
#
#
#
#
# See Creating databases and storage groups for more information.
#
#
esbDbStorageGroup=ESBDBSTO
################################################################################
#
# Database Schema Name for Enterprise Service Bus
#
# Specify the Database schema name for the configuration of the Enterprise
# Service Bus database. This value will be substituted in the generated DDL/SQL
# produced during the install for ESB. The default value is ESBLOG.
#
esbDbSqlId=ESBLOG

################################################################################
#
# Business Process Choreographer Properties
#
################################################################################
#
# The following entries are used to create a sample Business Process
# Choreographer (BPC) configuration. If mqUser, mqPwd, and adminBFMGroups values
# are not provided, then the sample BPC configuration  will not be created.
#
.
# Note: Business Process Choreographer configuration properties are ignored
# when configuring WebSphere Process Server for z/OS as an ESB-only server
#
################################################################################
#
# Business Process Choreographer MQ User Name
#
# Specify the user name for the configuration of resources for the Business
# Process Container using WebSphere MQ. The default value is ibmuser.
#
bpcmqUser=$JMSUSER

################################################################################
#
# Business Process Choreographer MQ Password
#
# Specify the password for the MQ User Name for the configuration of resources
# for the Business Process Container using WebSphere MQ. The default value is
# ibmuser.
#
bpcmqPwd=$JMSPASS

################################################################################
#
# Business Process Choreographer Admin Group
#
# Specify the security role for the business process system administrator. This
# is the security role used by the Business Process Choreographer to run a
# business process. The default value is ibmuser.
#
bpcadminGroups=IBMGROUP

################################################################################
#
# BPC Database User ID
#
# Specify the User name for authenicating the JDBC resources for the Business
# Process Container. The default value is wsadmin.
#
bpcdbUser=$DBUSER

################################################################################
#
# JDBC Password
#
```

```
# Specify the password for authenticating the JDBC resources
# for the Business Process Container. The default is value is wsadmin.
#
#
bpcdbPwd=$DBPASS

################################################################################
# BPC Database Type
#
# Specifies the default value of the database product name for Business Process Container.
# The default is zOS-DB2.
#
bpcdbType=zOS-DB2
################################################################################
#
# Business Process Choreographer Database Version
#
# Specify the Database Version for the configuration of the Business Process
# Container. The default value is 8. Valid values are:
#           8 = DB2 Version 8
#           7 = DB2 Version 7
#
#
bpcdbVersion=8

################################################################################
# Business Process Choreographer Database Subsytem Location
#
# Specifies the default database location for Business Process Container. This value
# will be used in place of the database name on the Websphere datasource definition.
# The default value is LOC1.
#
bpcdbSubSystem=$DBLOCATION

################################################################################
#
# Business Process Choreographer Database Home Directory
#
# Specify the Database JCC Home Directory that contains the following directories
# on z/OS:  classes  db2jcc.jar,  db2jcc_javax.jar, db2jcc_license_cisuz.jar.
# This value will vary depending on the DB2 Version used and where it is installed.
# The default value is /db2810/jcc.
#
#
bpcdbHome=/db2810/jcc
################################################################################
# Business Process Choreographer database name
#
#  Specify the DB2 Database name used by the Business Process Container. This
#  value will be substituted in the DDL/SQL produced by the BPC install. This
#  value must match Database name used in the pre-Configuration Database and
#  Storage Group creation task.
#  The default is BPEDB.
#
#
# See Creating databases and storage groups for more information.
#
bpcdbName=BPEDB
################################################################################
# Business Process Choreographer Database storage group
#
#
# Specify the Database storage group name for the configuration of the Business
# Process Container database. This value will be substituted in the DDL/SQL
# produced by the BPC install. This value must match Storage Group name used in the
# pre-Configuration Database and Storage Group creation task.
# The default value is BPEDBSTO.
#
#
# See Creating databases and storage groups for more information.
#
#
bpcdbStorageGroup=BPEDBSTO

################################################################################
# Business Process Choreographer Database Schema Name
#
#  Specify the DB2 Schema qualifier used by the Business Process Container. This value will
#  be substituted in the install produced ddl/sql.
```

```
#  The default is the global property $DBUSER.
#
bpcdbSQLID=$DBUSER

################################################################################
#
# WBI Server Properties
#
.
# Note: WBI Server configuration properties are ignored
# when configuring WebSphere Process Server for z/OS as an ESB-only server
#
#
################################################################################
################################################################################
#
# Application Scheduler configuration
# configureAppScheduler
#
# Application Scheduler is used to schedule migration application group events.
# If you choose to configure Application Scheduler, set the following property
# to "true". Otherwise, specify false.
#
configureAppScheduler=true

################################################################################
#
# Indicate the application server name on this node if you chose to configure
# it for Application Scheduler use. Usually, for a stand-alone profile the
# server name is "server1". The default is the global property $CONFIGSERVER.
#
appSchedulerServer=$CONFIGSERVER


################################################################################
# Configure Database Authentication for the product
# Various product components use a database connection.
# Choose a database type and enter the database configuration information
# based on that type.
#
################################################################################
#
# Database name
#
# Specify the name of the WebSphere Process Server for z/OS database. This value will be
# substituted in the generated DDL/SQL produced during the product install. This
# value must match database name used in the pre-Configuration Database and
# Storage Group creation task. The default value is WPSDB.
#
#
# See Creating databases and storage groups for more information.
#
dbName=WPSDB

################################################################################
#
# Database type
#
# Specify the type of the WebSphere Process Server database. The default is
# DB2UDBOS390_V8_1.
#
# Valid values are:
#            DB2UDBOS390_V8_1  for DB2 Version 8
#            DB2UDBOS390_V7_1  for DB2 Version 7

#
dbType=DB2UDBOS390_V8_1

################################################################################
# Database User ID (to authenticate with the database)
#
#  Specify the User ID for the WebSphere Process Server database
#  The default value is wsadmin
#
dbUserId=$DBUSER

################################################################################
# Database password (to authenticate with the database)
#
```

```
# Specify the Password for the WebSphere Process Server database.
# The default value is wsadmin
dbPassword=$DBPASS

################################################################################
# Database Storage Group
#
# Specify the Storage Group name for the WebSphere Process Server database.
# This value will be substituted in the generated DDL/SQL produced during the
# install for WPS. This value must match Storage Group name used in the
# pre-Configuration Database and Storage Group creation task.
# The default value is WPSDBSTO
dbStorageGroup=WPSDBSTO

################################################################################
# Database host name
#
# This property is used for defining a type 4 DB2 DataSource and will only be used
# for annotating the WebSphere Datasource. Install will only create a type2 provider.
# The default value is localhost.
#
dbHostName=localhost

################################################################################
#
# Database server port
#
# This property is used for defining a type 4 DB2 DataSource and will only be used
# for annotating the WebSphere Datasource. This value is not required.
#
# Install will only create a type2 provider. The default value is 446.
dbServerPort=446
################################################################################
#
# Database Connection Location
#
# Specify the Database Connection Location WebSphere Process Server.
# Specifies the default database location for WBI Server. This value
# will be used in place of the database name on the Websphere datasource definition.
# The default value is LOC1.
#
dbConnectionLocation=$DBLOCATION
################################################################################
#
# Database Home Classes Directory
#
# Specify the Database Home Classes Directory. This value will vary depending on
# the DB2 Version used and where it is installed. The default value is /db2810/jcc.
#
#
dbJDBCClasspath=/db2810/jcc/classes
################################################################################
#
# Database DB2 JCC Properties Directory
#
#  Specify the directory that contains the DB2JccConfiguration.properties file.
#  This file will be required by systems containing multiple DB2 instances or
#  using DB2 Data Sharing
#
dbJDBCProperties=/db2810/jcc/properties
################################################################################
#
# Create new or use existing database
#
# For DB2, the database for WebSphere Process Server must be created in the
# pre-Configuration Database and Storage Group creation task. Therefore, this
# defaults to "false" and should not be changed.
#
#
# For information, see Creating the databases and storage groups.
#
# The default is false.
#
dbCreateNew=false
################################################################################
#
# Execute the database table and datasource definitions
#
# Specify whether the definitions for the database table and datasource created
```

```
# during configuration task should run to create the resources.
# If the required databases and Storage groups have been defined to DB2, the installer
# may connect to the database and define the product definitions as part of the augment.
# Specify "true" to cause the definitions to be executed automatically during
# augmentation.
#
# Specify "false" to not execute the definitions. If "false" is
# specified, the definitions must be executed manually as a post-configuration
# task. The definitions are generated to the following directory:
#
#         $WAS_Home/profiles/default/databases.
#
# The default value is true
#
dbDefineSQL=true
```

Response file for network deployment configuration - dmgrProfile.rsp

The following is a sample of the default response file used for configuring WebSphere Process Server for z/OS in a network deployment configuration:

```
################################################################################
# Licensed Materials - Property of IBM
# 5655-N53, 5655-R15
# (C) Copyright IBM Corporation 2006. All Rights Reserved.
# US Government Users Restricted Rights - Use, duplication or disclosure
# restricted by GSA ADP Schedule Contract with IBM Corp.
################################################################################
################################################################################
#
# Sample Response file for WebSphere Process Server v6.0 Installer for z/OS
#
# This response defaults to DB2 version 8. The following properties need
# consideration if this profile is to be used for DB2 version 7 or other version:
#     dbType        dbJDBCClasspath  dbJDBCProperties
################################################################################
################################################################################
# GLOBAL Properties
# Global properties prepended by the symbol $ may used as substitution for
# other values in this file. The global property value must be the complete
# value to be substituted. See examples of $JMSUSER used in this file.
################################################################################
JMSUSER=ibmuser
JMSPASS=ibmuser
DBUSER=wsadmin
DBPASS=wsadmin
CONFIGSERVER=dmgr
DBLOCATION=LOC1

################################################################################
#
# Common Properties
#
################################################################################
#
# Augment
#
# The "augment" keyword is required in order for the install to perform the
# the profile augmentation process. There is no value associated with this
# keyword. The "augment" keyword may be replaced with "unaugment" in order to
# undo the changes from a previous augment. However, "augment" must be specified
# for an install to complete successfully. The field must be left as "augment"
# even if the install shell command line specifies "uninstall".
#
augment

################################################################################
#
# Profile name
#
# On z/OS, there is always one and only one profile and that profile is named
# default in each of the configurations.
#
# The profile referred to here is the default profile installed and
# and configured during the WebSphere Application Server for z/OS install.
#
profileName=default
```

```
################################################################################
#
# Profile path
#
# This value represents the default profile root directory for
# WebSphere Application Server for z/OS.
#
# The profile path contains the files that define the run-time environment,
# such as commands, configuration files, and log files.
#
# This path must match the profile path specified on the ISPF customization
# dialog when you created your deployment manager cell.
#
# If the default value was not used, make sure the value below matches the
# profile path specified for the WebSphere Application Server for z/OS
# installation.
#
profilePath=/WebSphere/V6R0M0/DeploymentManager/profiles/default

################################################################################
#
# Template path
#
# This value determines which actions are performed during the profile augment
# portion of the install. For WebSphere Process Server, the valid values are:
#
# /WebSphere/V6R0M0/DeploymentManager/profileTemplates/dmgr.wbicore
# /WebSphere/V6R0M0/DeploymentManager/profileTemplates/dmgr.bfm
# /WebSphere/V6R0M0/DeploymentManager/profileTemplates/dmgr.wbiserver
# /WebSphere/V6R0M0/DeploymentManager/profileTemplates/dmgr.*
#
# If one of the first three values is specified,
# only the actions within that directory are performed. The fourth
#
# value is the default (.../dmgr.*), which causes all of the actions in the first
# three template paths to be performed.
#
# Changing the default value will cause an incomplete install of the product
# and should only be done if there is a valid reason to do so.
#
# This path must match the template path specified on the ISPF customization
# dialog when you created your deployment manager cell.
#
templatePath=/WebSphere/V6R0M0/DeploymentManager/profileTemplates/dmgr.*


################################################################################
#
# Cell name
#
# This value determines the cell name within the WebSphere Application Server for
# which this Process Server install is being performed.
# The default value is v1dmnode.
#
# This must match the Cell name specified on the ISPF customization
# dialog when you created your deployment manager cell.
#
cellName=v1dmnode

################################################################################
#
# Node name
#
# This value determines the node name within the WebSphere Application Server for
# which this Process Server install is being performed. Node name under one cell
# has to be unique. The default value is v1cell.
#
# This must match the Node name specified on the ISPF customization
# dialog when you created your deployment manager cell.
#
nodeName=v1cell

################################################################################
#
# Server name
#
# This value determines the server name within the WebSphere Application Server
# for which this Process Server install is being performed. Server name under
```

```
# one node has to be unique. The default value is dmgr.
#
# This must match the Server name specified on the ISPF customization
# dialog when you created your deployment manager cell.
#
serverName=$CONFIGSERVER

################################################################################
#
# WBI Core Properties
#
################################################################################
#
# Configure SCA
#
################################################################################
#
# The following entries are used to create a Service Component Architecture
# sample configuration.
#
################################################################################
#
# Configure SCA security
#
# If you do not want to create a Service Component Architecture
# sample configuration, change the default value from true to false and
# leave the values for the properties specified below blank.
#
configureScaSecurity=true

################################################################################
#
# SCA security user id
#
# If you chose to create a Service Component Architecture configuration above
# then specify the userid for that configuration. Default value is ibmuser.
#
scaSecurityUserId=$JMSUSER

################################################################################
#
# SCA security password
#
# If you chose to create a Service Component Architecture configuration above
# then specify the password for SCA security user id for that configuration.
# Default value is ibmuser.
#
scaSecurityPassword=$JMSPASS

################################################################################
#
# WBI Server Properties
#
################################################################################
#
# Application Scheduler configuration
# configureAppScheduler
#
# Application Scheduler is used to schedule migration application group events.
# If you choose to configure Application Scheduler, set the following property
# to "true". Otherwise, specify false.
#
configureAppScheduler=false

################################################################################
#
# Indicate the application server name on this node if you chose to configure
# it for Application Scheduler use. Usually, for a stand-alone profile the
# server name is "server1". The default is value dmgr.
#
appSchedulerServer=$CONFIGSERVER

################################################################################
#
# Configure Database Authentication for process server.
# Various WebSphere Process Server components use a database connection.
# Choose a database type and enter the database configuration information
# based on that type.
#
```

```
################################################################################
#
# Database name
#
# Specify the name of the WebSphere Process Server database. This value will be
# substituted in the generated DDL/SQL produced during the install for WPS. This
# value must match database name used in the pre-Configuration Database and
# Storage Group creation task. The default value is WPSDB.
#
dbName=WPSDB

################################################################################
#
# Database type
#
# Specify the type of the WebSphere Process Server database. The default value
# is DB2UDBOS390_V8_1. Valid values are:
#            DB2UDBOS390_V8_1  for DB2 Version 8
#            DB2UDBOS390_V7_1  for DB2 Version 7
#
dbType=DB2UDBOS390_V8_1

################################################################################
#
# Database User ID
#
# Specify the User ID for the WebSphere Process Server database.
# The default value is wsadmin.
#
dbUserId=$DBUSER

################################################################################
#
# Database Password
#
# Specify the Password for the WebSphere Process Server database.
# The default value is wsadmin.
#
dbPassword=$DBPASS

################################################################################
#
# Database Storage Group
#
# Specify the Storage Group name for the WebSphere Process Server database.
# This value will be substituted in the generated DDL/SQL produced during the
# install for WPS. This value must match Storage Group name used in the
# pre-Configuration Database and Storage Group creation task.
# The default value is WPSDBSTO.
#
dbStorageGroup=WPSDBSTO

################################################################################
#
# Database Host Name
#
# This property is used for defining a type 4 DB2 DataSource and will only be used
# for annotating the WebSphere Datasource. Install will only create a type2 provider.
# The default value is localhost.
#
dbHostName=localhost

################################################################################
#
# Database Port Number
#
# This property is used for defining a type 4 DB2 DataSource and will only be used
# for annotating the WebSphere Datasource. This value is not required. # Install will
# only create a type2 provider. The default value is 446.
#
dbServerPort=446

################################################################################
#
# Database Connection Location
#
# Specify the Database Connection Location WebSphere Process Server.
# Specifies the default database location for WBI Server. This value
# will be used in place of the database name on the Websphere datasource definition.
```

```
# The default value is LOC1.
#
dbConnectionLocation=$DBLOCATION

################################################################################
#
# Database Home Classes Directory
#
# Specify the Database Home Classes Directory. This value will vary depending on
# the DB2 Version used and where it is installed. The default value is /db2810/jcc.
#
dbJDBCClasspath=/db2810/jcc/classes

################################################################################
#
# Database DB2 JCC Properties Directory
#
# Specify the directory that contains the DB2JccConfiguration.properties file.
# This file will be required by systems containing multiple DB2 instances or
# using DB2 DataSharing.
#
dbJDBCProperties=/db2810/jcc/properties

################################################################################
#
# Create new or use existing database
#
# For DB2, the database for WebSphere Process Server must be created in the
# pre-Configuration Database and Storage Group creation task. Therefore, this
# defaults to "false" and should not be changed.
#
dbCreateNew=false

################################################################################
#
# Execute the database table and datasource definitions
#
# Specify whether the definitions for the database table and datasource created
# during configuration task should actually be executed to create the resources.
# If the required databases and Storagegroups have been defined to DB2, the installer
# may connect to the database and define the WPS definitions as part of the augment.
# Specify "true" to cause the definitions to be executed automatically during
# augmentation. Specify "false" to not execute the definitions. If "false" is
# specified, the definitions must be executed manually as a post-configuration
# task. The default value is "false".
#
dbDefineSQL=false
```

Response file for managed node in a network deployment configuration - managedProfile.rsp

The following is a sample of the default response file used for configuring WebSphere Process Server for z/OS in a managed node in a network deployment configuration:

```
################################################################################
# Licensed Materials - Property of IBM
# 5655-N53, 5655-R15
# (C) Copyright IBM Corporation 2006. All Rights Reserved.
# US Government Users Restricted Rights - Use, duplication or disclosure
# restricted by GSA ADP Schedule Contract with IBM Corp.
################################################################################
################################################################################
#
# Sample Response file for WebSphere Process Server v6.0 Installer for z/OS
#
# This response defaults to DB2 version 8. The following properties need
# consideration if this profile is to be used for DB2 version 7 or other version:
#     dbType        dbJDBCClasspath    dbJDBCProperties
################################################################################
################################################################################
# GLOBAL Properties
# Global properties prepended by the symbol $ may used as substitution for
# other values in this file. The global property value must be the complete
# value to be substituted. See examples of $JMSUSER used in this file.
################################################################################
```

```
JMSUSER=ibmuser
JMSPASS=ibmuser
DBUSER=wsadmin
DBPASS=wsadmin
DBLOCATION=LOC1


###############################################################################
#
# Common Properties
#
###############################################################################
#
# Augment
#
# The "augment" keyword is required in order for the install to perform the
# the profile augmentation process. There is no value associated with this
# keyword. The "augment" keyword may be replaced with "unaugment" in order to
# undo the changes from a previous augment. However, "augment" must be specified
# for an install to complete successfully. The field must be left as "augment"
# even if the install shell command line specifies "uninstall".
#
augment

###############################################################################
#
# Profile name
#
# On z/OS, there is always one and only one profile and that profile is named
# default in each of the configurations.
#
# The profile referred to here is the default profile installed and
# and configured during the WebSphere Application Server for z/OS install.
#
profileName=default

###############################################################################
#
# Profile path
#
# This value represents the default profile root directory for
# WebSphere Application Server for z/OS.
#
# The profile path contains the files that define the run-time environment,
# such as commands, configuration files, and log files.
#
# This must match the profile path specified on the ISPF customization
# dialog when you created your empty managed node.
#
profilePath=/WebSphere/V6R0M0/AppServer/profiles/default

###############################################################################
#
# Template path
#
# This value determines which actions are performed during the profile augment
# portion of the install. The valid values are:
#
# /WebSphere/V6R0M0/DeploymentManager/profileTemplates/dmgr.wbicore
# /WebSphere/V6R0M0/DeploymentManager/profileTemplates/dmgr.bfm
# /WebSphere/V6R0M0/DeploymentManager/profileTemplates/dmgr.wbiserver
# /WebSphere/V6R0M0/DeploymentManager/profileTemplates/managed.*
#
# If one of the first three values is specified,
# only the actions within that directory are performed. The fourth
#
# value is the default (.../managed.*), which causes all of the actions in the first
# three template paths to be performed.
#
# Changing the default value will cause an incomplete install of the product
# and should only be done if there is a valid reason to do so.
#
# This must match the template path specified on the ISPF customization
# dialog when you created your empty managed node.
#
templatePath=/WebSphere/V6R0M0/AppServer/profileTemplates/managed.*

###############################################################################
#
```

```
# Cell name
#
# This value determines the cell name within the WebSphere Application Server for
# which this Process Server install is being performed.
# The default value is v1nodea.
#
# This must match the cell name specified on the ISPF customization
# dialog when you created your empty managed node.
#
cellName=v1nodea

################################################################################
#
# Node name
#
# This value determines the node name within the WebSphere Application Server for
# which this Process Server install is being performed. Node name under one cell
# has to be unique. The default value is v1cellx.
#
# This must match the Node name specified on the ISPF customization
# dialog when you created your empty managed node.
#
nodeName=v1cellx

################################################################################
#
# WBI Core Properties
#
################################################################################
#
# Configure SCA
#
################################################################################
#
# The following entries are used to create a Service Component Architecture
# sample configuration.
#
################################################################################
#
# Configure SCA security
#
# If you do not want to create a Service Component Architecture
# sample configuration, change the default value from true to false and
# leave the values for the properties specified below blank.
#
configureScaSecurity=true

################################################################################
#
# SCA security user id
#
# If you chose to create a Service Component Architecture configuration above
# then specify the userid for that configuration. Default value is ibmuser.
#
scaSecurityUserId=$JMSUSER

################################################################################
#
# SCA security password
#
# If you chose to create a Service Component Architecture configuration above
# then specify the password for SCA security user id for that configuration.
# Default value is ibmuser.
#
scaSecurityPassword=$JMSPASS

################################################################################
#
# WBI Server Properties
#
################################################################################
#
# Application Scheduler configuration
# configureAppScheduler
#
# Application Scheduler is used to schedule migration application group events.
# This value defaults to false for a managed node and should not be changed.
#
configureAppScheduler=false
```

```
################################################################################
#
# Configure Database Authentication for process server.
# Various WebSphere Process Server components use a database connection.
# Choose a database type and enter the database configuration information
# based on that type.
#
################################################################################
#
# Database name
#
# Specify the name of the WebSphere Process Server database. This value will be
# substituted in the generated DDL/SQL produced during the install for WPS. This
# value must match database name used in the pre-Configuration Database and
# Storage Group creation task. The default value is WPSDB.
#
dbName=WPSDB

################################################################################
#
# Database type
#
# Specify the type of the WebSphere Process Server database. The default value
# is DB2UDBOS390_V8_1. Valid values are:
#            DB2UDBOS390_V8_1  for DB2 Version 8
#            DB2UDBOS390_V7_1  for DB2 Version 7
#
dbType=DB2UDBOS390_V8_1

################################################################################
#
# Database User ID
#
# Specify the User ID for the WebSphere Process Server database.
# The default value is wsadmin.
#
dbUserId=$DBUSER

################################################################################
#
# Database Password
#
# Specify the Password for the WebSphere Process Server database.
# The default value is wsadmin.
#
dbPassword=$DBPASS

################################################################################
#
# Database Storage Group
#
# Specify the Storage Group name for the WebSphere Process Server database.
# This value will be substituted in the generated DDL/SQL produced during the
# install for WPS. This value must match Storage Group name used in the
# pre-Configuration Database and Storage Group creation task.
# The default value is WPSDBSTO.
#
dbStorageGroup=WPSDBSTO

################################################################################
#
# Database Host Name
#
# This property is used for defining a type 4 DB2 DataSource and will only be used
# for annotating the WebSphere Datasource. Install will only create a type2 provider.
# The default value is localhost.
#
dbHostName=localhost

################################################################################
#
# Database Port Number
#
# This property is used for defining a type 4 DB2 DataSource and will only be used
# for annotating the WebSphere Datasource. This value is not required. # Install will
# only create a type2 provider. The default value is 446.
#
dbServerPort=446
```

```
################################################################################
#
# Database Connection Location
#
# Specify the Database Connection Location WebSphere Process Server.
# Specifies the default database location for WBI Server. This value
# will be used in place of the database name on the Websphere datasource definition.
# The default value is LOC1.
#
dbConnectionLocation=$DBLOCATION

################################################################################
#
# Database Home Classes Directory
#
# Specify the Database Home Classes Directory. This value will vary depending on
# the DB2 Version used and where it is installed. The default value is /db2810/jcc.
#
dbJDBCClasspath=/db2810/jcc/classes

################################################################################
#
# Database DB2 JCC Properties Directory
#
# Specify the directory that contains the DB2JccConfiguration.properties file.
# This file will be required by systems containing multiple DB2 instances or
# using DB2 DataSharing.
#
dbJDBCProperties=/db2810/jcc/properties

################################################################################
#
# Create new or use existing database
#
# For DB2, the database for WebSphere Process Server must be created in the
# pre-Configuration Database and Storage Group creation task. Therefore, this
# defaults to "false" and should not be changed.
#
dbCreateNew=false

################################################################################
#
# Execute the database table and datasource definitions
#
# Specify whether the definitions for the database table and datasource created
# during configuration task should actually be executed to create the resources.
# If the required databases and Storagegroups have been defined to DB2, the installer
# may connect to the database and define the WPS definitions as part of the augment.
# Specify "true" to cause the definitions to be executed automatically during
# augmentation. Specify "false" to not execute the definitions. If "false" is
# specified, the definitions must be executed manually as a post-configuration
# task. The default value is "false".
#
dbDefineSQL=false
```

Modifying a response file

WebSphere Process Server for z/OS is packaged with response files that comprise profile information for each of the supported configurations. Before running the configuration script you must modify the content of the response file in order to customize configuration to your environment.

Before you modify a response file, make sure of the following:

- Create the databases and storage groups for the WebSphere Process Server for z/OS components that utilize databases.

  See Creating the databases and storage groups information.
- Understand the various configuration types supported by WebSphere Process Server for z/OS.

  See Environment configuration types.
- Understand how the response file is organized and how to read and edit the response files.

See Working with response files.

- This task assumes that you have access to the installed files and that you have knowledge of MVS system commands and tools for editing flat files.

You modify the property values in a default response file before running the WebSphere Process Server for z/OS configuration script.

1. Make a copy the desired response file from the HFS into which it was installed. Copy the file to a directory in which you have edit authority and save it with a new name such as **my_options_file.rsp**.
2. Edit the values in the response file to customize the profile property values for your installation. The response file contains detailed comments that explain the properties. The commented response files are also documented in Sample response files.
3. Save the modified file to the directory in which it was copied. When you run the configuration script the command will include the path to the modified response file.

After modifying the response file you can run the product configuration script.

Configuring the product

In general, the phrase *configuring the product* refers to those activities that a product administrator performs after the SMP/E install is complete in order to prepare the product for use.

Configuring the product results in augmenting the default profile with WebSphere Process Server configuration data.

In WebSphere Process Server for z/OS, you configure the product for use by running the WebSphere Process Server for z/OS configuration shell script (**zWPSConfig.sh** or **zWESBConfig.sh**).

About running the configuration script

Run the configuration script to augment the default profile with WebSphere Process Server for z/OS configuration data.

The configuration script relies on response files for specific configuration information. So, the response files provided represent the configurations supported by the product.

When you run the configuration script, the command includes a path to the response file for a particular configuration. The property values that you set in the response file are used to create your product configuration.

The following is a list of configurations supported by the product configuration script.

- Stand-alone server configuration supporting the following databases:
  - Cloudscape
  - DB2 for z/OS version 7
  - DB2 for z/OS version 8

  Default response files for these configurations reside on the install HFS, as follows:

- /usr/lpp/zWPS/V6R0/usr/lpp/zWPS/V6R0/zos.config/**standAloneProfile.rsp**
- /usr/lpp/zWPS/V6R0/usr/lpp/zWPS/V6R0/zos.config/**standAloneProfileDB2.rsp**

Note: The information required to create the DB2 for z/OS database is in the response file named standAloneProfileDB2.rsp. This response file contains properties that can be set to support DB2 for z/OS version 7 or DB2 for z/OS version 8. The default values are set for DB2 for z/OS version 8.

- Network Deployment configuration supporting the following databases:
  - DB2 for z/OS version 7
  - DB2 for z/OS version 8

  The default response file for these configurations resides on the install HFS, as follows:
  - /usr/lpp/zWPS/V6R0/usr/lpp/zWPS/V6R0/zos.config/**dmgrProfile.rsp**

  Note: The information required to create the DB2 for z/OS database is in the response file named dmgrProfile.rsp. This response file contains properties that can be set to support DB2 for z/OS version 7 or DB2 for z/OS version 8. The default values are set for DB2 for z/OS version 8.
- Managed node configuration supporting the following databases:
  - DB2 for z/OS version 7
  - DB2 for z/OS version 8

  he default response file for these configurations resides on the install HFS, as follows:
  - /usr/lpp/zWPS/V6R0/usr/lpp/zWPS/V6R0/zos.config/**managedProfile.rsp**

  Note: The information required to create the DB2 for z/OS database is in the response file named managedProfile.rsp. This response file contains properties that can be set to support DB2 for z/OS version 7 or DB2 for z/OS version 8. The default values are set for DB2 for z/OS version 8.

Although the process of running the command is almost identical for each of the WebSphere Process Server for z/OS configurations (the difference being the response file that is referenced in the command), there are nuances with regard to each of the WebSphere Process Server for z/OS configurations. Before you run the configuration script from the command line, read about each of the WebSphere Process Server for z/OS types.

About running the configuration script for a standalone application server

Running the configuration script for a standalone application server cell augments the default profile with WebSphere Process Server for z/OS configuration data.

To augment the default profile with WebSphere Process Server for z/OS configuration data you must run the configuration script with the `-augment` keyword and the -response keyword and the keyword parameter representing the path to the response file associated with the stand-alone configuration.

Things to know about profile augmentation in a standalone configuration

1. The standalone configuration supports a Cloudscape database and a DB2 for z/OS database.

The information required to create the Cloudscape database is in the response file named `standAloneProfile.rsp`.

The information required to create the DB2 for z/OS database is in the response file named `standAloneProfileDB2.rsp`. This response file contains properties that can be set to support DB2 for z/OS version 7 or DB2 for z/OS version 8. The default values are set for DB2 for z/OS version 8.

The absolute path of this response file is included as an argument in the configuration command.

2. You will need to perform DB2 for z/OS setup procedures before you run the configuration script.
3. You will need to edit default values in the response file before you run the configuration script.
4. There are properties in the stand-alone response file that must match values that you specified when you Created the stand-alone application server cell. These values include the following:
   - Cell name
   - Node name
   - Server name
   - Template path
   - Profile path

If you specified values for the Business Process Choreographer component in the response file, a sample configuration that includes the business process container, the human task container and the Business Process Choreographer Explorer are now part of your Process Server configuration. You can check to see if these components are configured by looking in the administrative console for enterprise applications with names that start with BPEContainer, BPCExplorer, and TaskContainer.

Note: The sample configuration uses a Cloudscape database and the WebSphere default messaging provider. This sample configuration is not suitable for a production system. Because you can only have one Business Process Choreographer configuration, you must remove the sample configuration, as described in the topic *Removing Business Process Choreographer* in the Uninstalling section, before you can continue configuring Business Process Choreographer to use WebSphere MQ or a different database. For more information, see *Configuring Business Process Choreographer*.

About running the configuration script for a network deployment cell

Running the configuration script for a network deployment cell augments the default profile with WebSphere Process Server for z/OS configuration data.

To augment the default profile with WebSphere Process Server for z/OS configuration data you must run the configuration script with the `-augment` keyword and the -response keyword and the keyword parameter representing the path to the response file associated with the network deployment configuration.

Things to know about profile augmentation in a network deployment configuration

1. The network deployment configuration supports a DB2 for z/OS database only.
2. You will need to perform DB2 for z/OS setup procedures before you run the configuration script.

3. You will need to edit default values in the response file before you run the configuration script.
4. Do not specify values for the Business Process Choreographer sample configuration in the response file.

   The sample Business Process Choreographer configuration enabled through response file settings is valid for a stand-alone configuration only.

   To create a Business Process Choreographer configuration for the network deployment configuration, run bpeconfig.jacl after you run the configuration script, as described in the section Configuring Business Process Choreographer.
5. There are properties in the deployment manager response file that must match values that you specified when you Created the Network Deployment cell. These values include the following:
   - Cell name
   - Node name
   - Server name
   - Template path
   - Profile path

About running the configuration script for a managed node in a network deployment cell

Running the configuration script for a managed node in a network deployment cell augments the default profile with WebSphere Process Server for z/OS configuration data.

To augment the default profile with WebSphere Process Server for z/OS configuration data you must run the configuration script with the `-augment` keyword and the -response keyword and the keyword parameter representing the path to the response file associated with the managed node configuration.

Things to know about profile augmentation in a managed node configuration

1. You must configure WebSphere Application Server for z/OS to support an empty cluster prior to creating the WebSphere Process Server for z/OS managed node configuration. WebSphere Application Server for z/OS configuration tasks.
2. The managed node configuration supports a DB2 for z/OS database only.
3. You will need to perform DB2 for z/OS setup procedures before you run the configuration script.
4. You will need to edit default values in the response file before you run the configuration script.
5. Do not specify values for the Business Process Choreographer sample configuration in the response file.

   The sample Business Process Choreographer configuration enabled through response file settings is valid for a stand-alone configuration only.

   To create a Business Process Choreographer configuration for the managed node configuration, run bpeconfig.jacl after you run the configuration script as described in the section Configuring Business Process Choreographer.
6. There are properties in the managed node response file that must match values that you specified when you Created the Empty Managed Node. These values include the following:
   - Cell name
   - Node name

- Server name
- Template path
- Profile path

7. When creating the empty managed node, do not run job BBOWMNAN to federate the node into the Deployment Manager cell.
8. When you run the WebSphere Process Server for z/OS managed node configuration, it results in a serverless node.

   You must clone a stand-alone server into the serverless node in order for the WebSphere Process Server for z/OS to work.

Run the configuration script

Running **zWPSConfig.sh** or **zWESBConfig.sh** augments the WebSphere Application Server for z/OS default profile server with WebSphere Process Server for z/OS configuration data.

The WebSphere Process Server for z/OS configuration process assumes a working knowledge of z/OS UNIX System Services. You may want to have access to the following documentation:

- z/OS V1R7.0 UNIX System Services User's Guide.
- z/OS V1R7.0 UNIX System Services Command Reference.

Before you run the configuration script make sure of the following:

1. Review Task overview: installing for an understanding of the task flow of the installation and configuration process.
2. That the systems programmer who is in charge of security and system access at your installation has set you up so that you can properly access the shell.

   This may involve making modifications to your RACF (security) profile and creating a home directory within the UNIX shell. The home directory is where you always start off when you begin a UNIX session. Within this directory, you keep any environmental variable files you may need to execute programs. These files contain information that's required in certain languages, like the location of the Java classes that are used in Java programs. You can also use the home directory as the root directory for keeping your work data. See Resource Access Control Facility Tools in the WebSphere Application Server for z/OS for information on tools and techniques to help you manage the security definitions used for WebSphere Application Server for z/OS.
3. Read about running the configuration command for the particular configuration you are about to create.

   For a stand-alone configuration, read About running the configuration script for a standalone application server.

   For a network deployment configuration, read About running the configuration script for a network deployment cell.

   For a managed node configuration, read About running the configuration script for a managed node in a network deployment cell.
4. The WebSphere Application Server for z/OS is installed and customized.

   The system programmer has completed the SMP/E installation for WebSphere Application Server for z/OS and has performed the required customization tasks and has run the resulting configuration jobs to configure WebSphere Application Server for z/OS.

   If you are creating a managed node configuration for WebSphere Process Server for z/OS, you must follow the steps for creating an empty cluster as described in the WebSphere Application Server for z/OS information center.

As part of the post configuration process for creating a WebSphere Process Server for z/OS managed node configuration, you clone a stand-alone application server and you populate the serverless node that is created when you run the configuration script using the managed node response file.

5. The WebSphere Process Server for z/OS product code has been loaded onto the system.

   The system programmer has completed the SMP/E installation by successfully loading the contents of the WebSphere Process Server for z/OS tape onto the system.

6. The WebSphere Process Server for z/OS definitions have been created.

   The install script **zSMPInstall.sh** with the `-install` argument was run to update the administrative console with the WebSphere Process Server for z/OS definitions.

7. Create the databases and storage groups required for your product configuration.

   See Creating the databases and storage groups for information.

8. Choose the response file that is associated with the product configuration you want.

   See Choosing a response file for information.

9. Edit the property values in the response file that will be used as the keyword parameter of the **-response** argument.

   See Working with response files for information on editing property values in the response file and for links to response file samples.

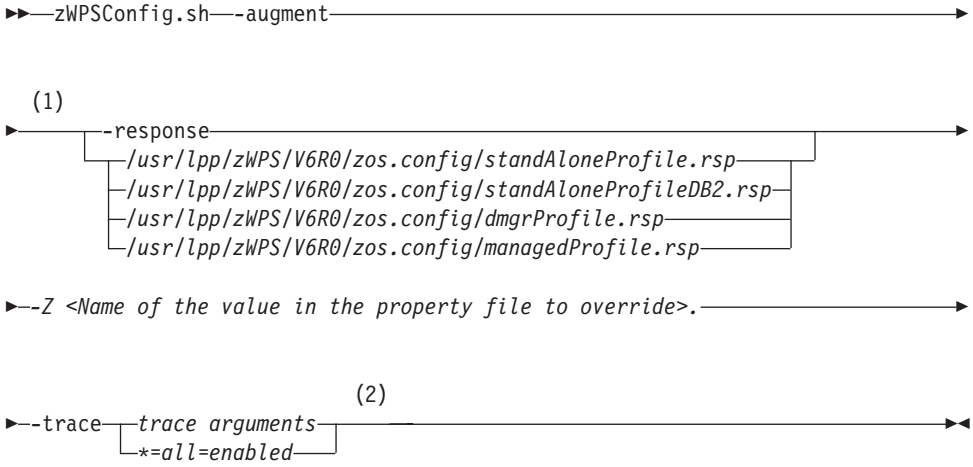
Perform this task when your goal is to augment the default profile with WebSphere Process Server for z/OS configuration data.

1. Access the OS/390 UNIX command shell. Enter the TSO command OMVS at the ISPF Command Shell or TSO OMVS from any other ISPF panel.

   Once you're in the UNIX shell, a command prompt (usually a dollar ($) or pound (#) sign) indicates that the system is ready to accept input.

2. From the command prompt, run the WebSphere Process Server for z/OS configuration command. Use the following syntax diagram as guide on how run the command:

Syntax diagram for configuring WebSphere Process Server for z/OS

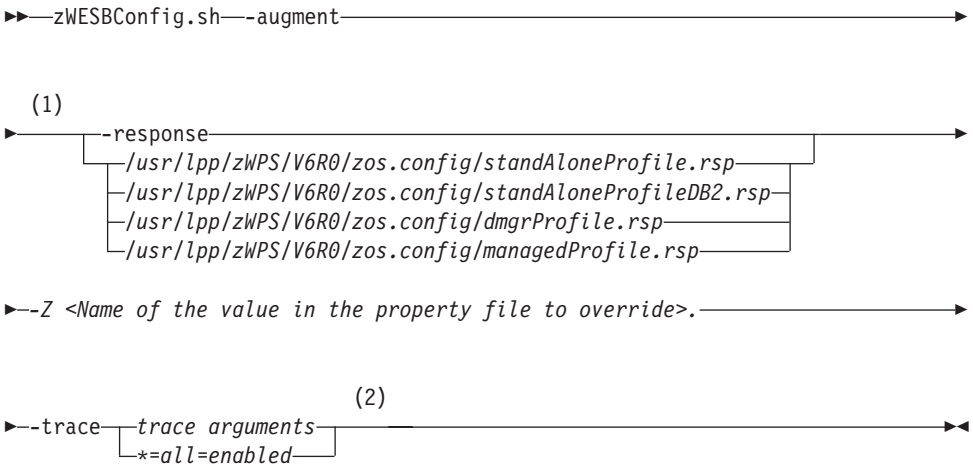


Notes:

1 Type the absolute path of the response file that is associated with the product configuration you are creating. This example shows the default path to the response files packaged with the product. When you run the command, the **-response** parameter will represent the absolute path to the response file that you have modified with the values specific to your environment.

2 By including **-trace** the command writes to the appropriate trace file.

Syntax diagram for configuring WebSphere Process Server for z/OS as an ESB-only server



Notes:

1 Type the absolute path of the response file that is associated with the configuration onto which you are installing WebSphere Process Server for z/OS. This absolute pathname in this example assumes you are using the default response files. If you wanted to customize the response file, this pathname would need to represent the absolute path of the customized file.

2 By including **-trace** the command writes to the appropriate log file.

Note: The instructions described above indicate how to run the install script from a OS/390 UNIX command shell. Alternatively you can run the install script from a prompt using Telnet. To run the install script using Telnet, type the following from the system prompt: telnet **TCPIP-Adress 2023** and run the command as indicated in the instructions above.

Review the messages from Standard Out. These are either displayed on the screen from which the **zWPSConfig.sh** or **zWESBConfig.sh** was run or in the file specified if the redirect **(">")** symbol was used on the command line.

There should be no error messages displayed.

An example of a successful execution of **zWPSConfig.sh** or **zWESBConfig.sh** with the -augment option is as follows:

```
parsing command arguments...
parsing arguments complete
setting up configuration...
```

```
runtimeRootDirName is: /WebSphere/V6R0M0/AppServer
WAS_HOME is: /WebSphere/V6R0M0/AppServer
WBI_HOME is: /WebSphere/V6R0M0/AppServer
set up configuration complete
augmenting profile(s)...
augmenting profile(s) complete
```

If you specified values for the Business Process Choreographer component in the response file, a sample configuration that includes the business process container, the human task container and the Business Process Choreographer Explorer are now part of your Process Server configuration. You can check to see if these components are configured by looking in the administrative console for enterprise applications with names that start with BPEContainer, BPCExplorer, and TaskContainer.

Note: The sample configuration uses a Cloudscape database and the WebSphere default messaging provider. This sample configuration is not suitable for a production system. Because you can only have one Business Process Choreographer configuration, you must remove the sample configuration, as described in as described in the topic *Removing Business Process Choreographer* in the Uninstalling section, before you can continue configuring Business Process Choreographer to use WebSphere MQ or a different database.

Post configuration

Post configuration tasks are those tasks that you perform after you have run the configuration script successfully.

Certain settings in the response file and certain environment conditions may necessitate post configuration tasks. These tasks are as follows:

- Configuring the database manually
- Running additional Jacl scripts

Configuring the database manually

Certain property value settings in the response file necessitate manual configuration of the database.

You must configure the database manually, when the following conditions exist:

- You have created the databases and storage groups.
- You have set the **Define SQL tables to DB2** property in the response file to false (**dbDefineSQL=false**).

  If the databases and storage groups have been created, the configuration script can connect to the database and create the database definitions automatically. However, if the database administrator at your site does not allow scripts to automatically update the database, you must set the **Define SQL tables to DB2** property to false.

  When you set **dbDefineSQL=false** in the response file, the configuration script generates the SQL, but does not execute the code to automatically create the database definitions. Instead, the SQL is written out to a directory.

The database administrator, or a person with authority to create database definitions, performs the following steps after to configure the database manually:

1. Access the generated SQL required to create the database definitions. The SQL for Business Process Choreographer, Common Event Infrastructure and enterprise service bus, exist in the following directory $WAS_Home/profiles/default/databases.

The SQL for components using the common database exist in the following directory$WAS_Home/profiles/default/databases/WPSDB.

2. Using the database tool of your choice, run the SQL to create the database definitions.

You have configured the database manually.

Configuring Business Process Choreographer

This describes how to configure the Business Process Choreographer containers for business processes and for human tasks, and the Business Process Choreographer Explorer.

Business Process Choreographer supports enterprise applications that include business processes and human tasks. It provides a container for business processes and a container for human tasks. These containers must be installed and configured before being used. The human task container requires the business process container and the staff service. The Business Process Choreographer Explorer provides a Web client interface for human interaction and administrating business processes and human tasks.

1. If you ran the WebSphere Process Server for z/OS install script and specified values for the Business Process Choreographer component in the response file, a sample configuration that includes the business process container, the human task container and the Business Process Choreographer Explorer already exists.

   You can check to see if these components are configured by looking in the administrative console for enterprise applications with names that start with `BPEContainer`, `BPCExplorer`, and `TaskContainer`.

   The sample configuration uses a Cloudscape database and the WebSphere default messaging provider. This sample configuration is not suitable for a production system. Because you can only have one Business Process Choreographer configuration, you must remove the sample configuration, as described in "Removing the Business Process Choreographer configuration" on page 237 before you can continue configuring Business Process Choreographer to use WebSphere MQ or a different database.

2. Configure the business process container.

   You can configure the business process container in one of the following ways:

   - Manually (**recommended**), by setting properties on the administrative console or through an administrative scripting process.

     If you want to use an administrative script to configure Business Process Choreographer, see "Using the bpeconfig.jacl script file to configure Business Process Choreographer" on page 138.
   - Automatically, by using the installation wizard available on the administrative console.

     If you want to use the installation wizard available on the administrative console, perform both of the following:

     – "Configuring the business process container using the installation wizard" on page 147
     – "Configuring the human task container, using the installation wizard" on page 169

     Note: The installation wizard configures WebSphere resources only. If you choose to configure the business process container by using the installation wizard, you will still need to run the corresponding manual step(s) to create a database (Cloudscape or DB2) and to create

the WebSphere MQ queues (if you are using WebSphere MQ as your Java Message Service (JMS) provider).

3. If you are using an LDAP staff plug-in, perform: "Configuring the LDAP staff plug-in provider" on page 176. The system and user registry staff plug-in providers can be used without configuring them.
4. If you have a Network Deployment (ND) environment, and you either used the human task container install wizard, or an error occurred when running the bpeconfig.jacl script, then you must setup the scheduler calendars application. Perform one of the following:
   - If you have already installed the scheduler calendars application on a server, install it on additional servers by performing the following steps:
     a. Select **Applications → Enterprise Applications**.
     b. Select **SchedulerCalendars**.
     c. Under the Additional Properties section, select **Map modules to servers**.
     d. Select the check box for the **Module Calendars**.
     e. Select all the servers and the clusters on which you have configured a business process container, be sure to also select any servers or clusters where you want the SchedulerCalendars application to remain.
     f. Select **Apply**.
     g. Select **OK**.
     h. Save and Synchronize changes with Nodes.
   - If this is the first time that you are installing the scheduler calendars application on a server, perform the following steps:
     a. Select **Applications → Install New Application**.
     b. In the file selector window, browse to the `installableApps` subdirectory of the *`install_root`* directory.
     c. Select **ScheduleCalendars.ear**.
     d. Select **Next**.
     e. Accept the default values and select **Next** again.
     f. Continue to accept the default values until you get to the 'Map modules to servers' step, then select any servers and clusters on which you want to load the ScheduleCalendars application, then select **Next**.
     g. On the summary step, select **Finish**.
     h. After the application has finished installing, select **Save to Master Configuration**.
     i. Save and synchronize changes.
5. Activate Business Process Choreographer: Perform "Activating Business Process Choreographer" on page 211.
6. **Optional:** If you have not yet installed and configured the Business Process Choreographer Explorer, you can configure it now. Perform "Configuring Business Process Choreographer Explorer" on page 201.
7. **Optional:** Verify that Business Process Choreographer works: Perform "Verifying that Business Process Choreographer works" on page 211.

Business Process Choreographer is configured and working.

Now you can run enterprise applications that contain business processes or human tasks, or both.

Using the bpeconfig.jacl script file to configure Business Process Choreographer:

This sample script file configures all the resources that are required by Business Process Choreographer.

Purpose

This script can either be run interactively, or in batch mode. It configures a working business process container, and human task container without using the installation wizard or administrative console. It can create a local database, and the necessary messaging resources, and also configure the Business Process Choreographer Explorer. If the script is applied to an application server that is in a cluster, all servers in the cluster will be configured for using Business Process Choreographer.

Location

The bpeconfig.jacl script file is located in the Business Process Choreographer samples directory: *install_root*/ProcessChoreographer/sample

Restrictions

This sample script has the following restrictions:

In an ND environment or a cluster
To configure several application servers in an ND environment or a cluster, the bpeconfig.jacl script must run interactively. The script cannot be used to perform this type of configuration when run non-interactively.

Runing the script in a stand-alone server environment

In a stand-alone server environment:

- Include the `-conntype NONE` option only if the application server is not running.
- If the server is running and global security is enabled, include the `-username` and `-password` options.

Runing the script in an ND environment

In a Network Deployment environment:

- Run the bpeconfig.jacl script on the deployment manager node.
- Include the `-conntype NONE` option only if the deployment manager is not running.
- If global security is enabled, include the `-username` and `-password` options.

Configuring the business process container non-interactively

If you provide the necessary parameters on the command line, you will not be prompted for them. To configure Business Process Choreographer, the following command: if your current directory is *install_root*/ProcessChoreographer, enter

```
../bin/wsadmin.sh -f bpeconfig.jacl parameters
```

where *parameters* are as follows:

```
  -conntype NONE
  -user  userName
  -password  userPassword
  -profileName profileName
 {(-node nodeName -server serverName) | -cluster clusterName}
  {-adminBFMUsers userList    | -adminBFMGroups groupList}
  {-monitorBFMUsers userList | -monitorBFMGroups groupList}
  -jmsBFMRunAsUser userID
  -jmsBFMRunAsPwd password
  {-adminHTMUsers userList    | -adminHTMGroups groupList}
  {-monitorHTMUsers userList | -monitorHTMGroups groupList}
  -jmsHTMRunAsUser userID
  -jmsHTMRunAsPwd password
  -mailServerName mailServerName
  -mailUser mailUserID
  -mailPwd mailPassword
  -hostName explorerVirtualHostname
  -explorerHost explorerURL
  -createDB { yes | no }
  -dbType databaseType
  -dbVersion DB2zOSversion
  -dbHome databaseInstallPath
  -dbJava JDBCDriverPath
  -dbName databaseName
  -dbUser databaseUser
  -dbPwd databasePassword
  -dbAdmin databaseAdministratorUserID
  -driverType JDBCDriverType
  -dbTablespaceDir databaseTablespacePath
  -dbServerName databaseServerName
  -dbServerPort databaseServerPort
  -dbStorageGroup DB2zOSStorageGroup
  -dbSubSystem DB2zOSSubSystem
  -dbSQLID DB2zOSSchemaQualifier
  -dbInstance InformixInstance
  -mqType JMSProviderType
  -createQM { yes | no }
  -qmNameGet getQueueManagerName
  -mqClusterName appServerClusterName
  -qmNamePut putQueueManagerName
  -mqHome MQInstallationDirectory
  -mqUser JMSProviderUserID
  -mqPwd JMSProviderPassword
  -mqSchemaName mqSchemaName
  -mqCreateTables  { true | false }
  -mqDataSource datasourceName
  -shell shell
```

Note: Some of the above parameters are optional, depending on the values provided for other parameters. The dependencies between parameters and the conditions that determine whether a parameter is optional or required is described for each parameter in the descriptions below. Any required parameters that are not specified on the command line are prompted for interactively.

Parameters

You can use the following parameters when invoking the script using wsadmin:

conntype NONE
This specifies that no administration connection is available. Only include this option if the application server (for stand-alone) or deployment manager (for ND) is not running.

user *userName*
If global security is enabled, you must provide a user ID for authentication.

password *userPassword*
If global security is enabled, you must provide the password for the user ID *userName*.

profileName *profileName*
Where *profileName* is the name of a user-defined profile. On z/OS the profile name is default.

node *nodeName*
Where *nodeName* is the name of the node where Business Process Choreographer is to be configured. If you have only one node and exactly one server, this parameter is optional. Do not use this option if you use the cluster option to specify a cluster.

server *serverName*
Where *serverName* is the name of the server where Business Process Choreographer is to be configured. If you have only one node and exactly one server, this parameter is optional. Do not use this option if you use the cluster option to specify a cluster.

cluster *clusterName*
Where *clusterName* is the name of the cluster where Business Process Choreographer is to be configured. Do not specify this option in a stand-alone server environment, nor if you specify the node and server. This option cannot be used non-interactively.

adminBFMUsers *userList*
Where *userList* is the list of names of users, from the user registry, to which to map the BPESystemAdministrator Java 2 Enterprise Edition (J2EE) role. The separator character is the vertical line (|). This property is needed to install the business process container. This parameter has no default value. Either one or both of the adminBFMUsers or adminBFMGroups options must be set.

adminBFMGroups *groupList*
Where *groupList* is the list of names of groups, from the user registry, to which to map the BPESystemAdministrator J2EE role. The separator character is the vertical line (|). This property is needed to install the business process container. This parameter has no default value. Either one or both of the adminBFMUsers or adminBFMGroups options must be set.

monitorBFMUsers *userList*
Where *userList* is the list of names of users, from the user registry, to which to map the BPESystemMonitor J2EE role. The separator character is the vertical line (|). This property is needed to install the business process container. This parameter has no default value. Either or both monitorBFMUsers or monitorBFMGroups must be set.

monitorBFMGroups *groupList*
Where *groupList* is the list of names of groups, from the user registry, to which to map the BPESystemMonitor J2EE role. The separator character isthe vertical line (|). This property is needed to install the business process container. This parameter has no default value. Either or both monitorBFMUsers or monitorBFMGroups must be set.

jmsBFMRunAsUser *userID*
Where *userID* is the run-as user ID from the user registry for the business process container JMS API. This property is needed to install the business process container. This parameter has no default value. It must be set.

jmsBFMRunAsPwd *password*
Where *password* is the password for the business process container JMS API. This property is needed to install the business process container. This parameter has no default value. It must be set.

adminHTMUsers *userList*
Where *userList* is the list of names of users, from the user registry, to which to map the TaskSystemAdministrator Java 2 Enterprise Edition (J2EE) role. The separator character is the vertical line (|). This property is needed to install the task container. This parameter has no default value. Either one or both of the adminHTMUsers or adminHTMGroups options must be set.

adminHTMGroups *groupList*
Where *groupList* is the list of names of groups, from the user registry, to which to map the TaskSystemAdministrator J2EE role. The separator character is the vertical line (|). This property is needed to install the task container. This parameter has no default value. Either one or both of the adminHTMUsers or adminHTMGroups options must be set.

monitorHTMUsers *userList*
Where *userList* is the list of names of users, from the user registry, to which to map the TaskSystemMonitor J2EE role. The separator character is the vertical line (|). This property is needed to install the task container. This parameter has no default value. Either or both monitorHTMUsers or monitorHTMGroups must be set.

monitorHTMGroups *groupList*
Where *groupList* is the list of names of groups, from the user registry, to which to map the TaskSystemMonitor J2EE role. The separator character is the vertical line (|). This property is needed to install the task container. This parameter has no default value. Either or both monitorHTMUsers or monitorHTMGroups must be set.

jmsHTMRunAsUser *userID*
Where *userID* is the run-as user ID from the user registry for the human task container JMS API. This property is needed to install the human task container. This parameter has no default value. It must be set.

jmsHTMRunAsPwd *password*
Where *password* is the password for the human task container JMS API. This property is needed to install the human task container. This parameter has no default value. It must be set.

mailServerName *mailServerName*
Where *mailServerName* is the host name of the mail server to be used by the Human Task Manager to send notification mails. It is needed when configuring the mail session. If this parameter is not set, the mail session configuration will be skipped. The default value is the fully qualified host name of the local host.

mailUser *mailUserID*
Where *mailUserID* is the user ID to access the mail server. It is needed to create the mail session for the Human Task Manager to send notification mails. The default value is empty, which is only appropriate if no authentication is required.

mailPwd *mailPassword*
Where *mailPassword* is the password to access the mail server. It is needed to create the mail session for the Human Task Manager to send notification mails.

hostName *explorerVirtualHostname*
Where *explorerVirtualHostname* is the virtual host where the Business Process Choreographer Explorer will run. The default value is `default_host`.

explorerHost *explorerURL*
Where *explorerURL* is the URL of the Business Process Choreographer Explorer. If this parameter is not specified for a non-cluster environments, a default value is computed, for example, http://localhost:9080. The value of this parameter is used for the EscalationMail.ClientDetailURL custom property of the Human Task Manager.

createDB { yes | no }
Possible values are `yes` or `no`. If set to `yes`, the script will create the database. For z/OS databases, this script cannot create the database, it can only create the table spaces and tables. For other database types, the default value is `yes`.

For information on how to create the database and storage groups for DB2 for z/OS, see *Creating databases and storage groups* in the section titled Planning your configuration.

dbType *databaseType*
Where *databaseType* is the database type. This is needed for installing the business process container, for creating the database or database tables, and for creating the data source. There is no default value. Possible values are:

- Cloudscape
- zOS-DB2

dbVersion *DB2zOSversion*
Where *DB2zOSversion* is either the value 7 or 8. This parameter is only required when the database type is DB2 for z/OS. It has no default value.

dbHome *databaseInstallPath*
Where *databaseInstallPath* is the installation directory of the database system. It is used to create the database or the database tables, and for creating the data source. The default value, and requirements depends on the database and on the platform:

For DB2:

- The default is `/home/${dbUser}/sqllib`.

The directories `${dbHome}/bnd` and `${dbHome}/bin` must exist.

dbJava *JDBCDriverPath*
Where *JDBCDriverPath* is the directory where the JDBC driver is located. This parameter is only required for the following combinations of database and driver types:

- DB2 or DB2 for z/OS, with a type 4 driver. The default value is `${dbHome}/java`.

dbName *databaseName*
Where *databaseName* is the name of the Business Process Choreographer database. It is used to create the database or the database tables, and for creating the data source. The default value is `BPEDB`..

dbUser *databaseUser*
Where *databaseUser* is the user ID to access the database. It is used to create the database tables and the data source. The default value depends on the database and platform. For DB2 for z/OS the default value is db2inst1.

dbPwd *databasePassword*
Where *databasePassword* is the password for the user ID *databaseUser*.

dbAdmin *databaseAdministratorUserID*
Where *databaseAdministratorUserID* is the user ID of the database administrator. It is only required when creating the database and database tables for the following database types:

- For DB2 for z/OS, the default is `db2inst1`.

driverType *JDBCDriverType*
Where *JDBCDriverType* is the type of JDBC driver. It is used to create the data source.

- For DB2, possible values are `Universal` or `CLI`. It is also used for creating the database tables.

dbTablespaceDir *databaseTablespacePath*
Where *databaseTablespacePath* is the directory where the database table spaces are created. It is used to create the database and database tables. This parameter is only required for the following database types:

- For DB2, the default value is empty, which means that no table spaces are created.

dbServerName *databaseServerName*
Where *databaseServerName* is the host name server that hosts the database for Business Process Choreographer. database. It is used to create the data source. For Sybase it is also used to create the database.

- For DB2, the default value is empty.
- For all other database types, the default value is the fully qualified host name of the local host.

dbServerPort *databaseServerPort*
Where *databaseServerPort* is the TCP/IP port for the database server for Business Process Choreographer. This parameter is required if dbServerName is specified. For DB2, the default value is `50000`.

dbStorageGroup *DB2zOSStorageGroup*
Where *DB2zOSStorageGroup* is the storage group used to create the Business Process Choreographer database table . This parameter is only required for DB2 on z/OS. There is no default value, and must not be empty.

dbSubSystem *DB2zOSSubSystem*
Where *DB2zOSSubSystem* is the DB2 sub system used to create the Business Process Choreographer database table and the data source. This parameter is only required for DB2 on z/OS. The default value is `BPEDB`.

dbSQLID *DB2zOSSchemaQualifier*
Where *DB2zOSSchemaQualifier* is the schema qualifier used to create the database tables. This parameter is only required for DB2 on z/OS. There is no default value. The value can be empty. Only specify a value when using the `Universal` JDBC driver type.

mqType *JMSProviderType*
Where *JMSProviderType* is the type of Java Message Service (JMS) provider to use for Business Process Choreographer. It is used to create the queue manager and the queues, the listener ports or ActivationSpecs, and the queue connection factories.

Where *JMSProviderType* is one of the following values:

WPM For default messaging (WebSphere Platform Messaging). This option is always available.

MQSeries
For WebSphere MQ. This option requires that the product WebSphere MQ is installed.

createQM *{ yes | no})*
Controls whether the script creates a local WebSphere MQ queue manager. This option only has an effect if the parameter mqType has the value `MQSeries`. The default value for this parameter is `yes`. Use the value `no` if you do not want the script to create the WebSphere MQ queue manager, for example, if you want to create the queue manager on a different server to the one where you are running the script.

qmNameGet *getQueueManagerName*
Where *getQueueManagerName* is the name of the queue manager for GET requests. It is used to create the queue manager and the queues, and to create the listener ports and the queue connection factories. It must not contain the `-` character. The default value for *getQueueManagerName* is `BPC_nodeName_serverName`. This option only has an effect if the parameter mqType has the value `MQSeries`.

mqClusterName *appServerClusterName*
Where *appServerClusterName* is the name of the WebSphere Application Server cluster where the default JMS provider's message engines are to be created. This has nothing to do with a WebSphere MQ cluster. This option is only used when configuring Business Process Choreographer in a cluster and the mqType option is set to WPM.

qmNamePut *putQueueManagerName*
Where *putQueueManagerName* is the queue manager name for PUT requests. It is used only when the mqClusterName parameter has been set. It is used to create the queue manager and the queues, and to create the listener ports and the queue connection factories. It must not contain the `-` character, and it must not be the same as the queue manager name specified for the qmNameGet parameter. The default value for *putQueueManagerName* is `BPCC_nodeName_serverName`.

mqHome *MQInstallationDirectory*
Where *MQInstallationDirectory* is the installation directory of WebSphere MQ. This is used to create the queue manager and the queues (Windows systems only) and for creating the listener ports and the queue connection factories. If the WebSphere variable MQ_INSTALL_ROOT is set, its value is used, and is not modified. This option only has an effect if the parameter mqType has the value `MQSeries`.

If MQ_INSTALL_ROOT is not set, the default value used for *MQInstallationDirectory* depends on the platform:

AIX: /usr/mqm

Solaris and HP-UX:
/opt/mqm

mqUser *JMSProviderUserID*
Where *JMSProviderUserID* is the user ID to access the JMS provider.

- If mqType has the value `WPM`, this parameter is used to create the ActivationSpecs and the connection factories; the default value is the currently logged on user.
- If mqType has the value `MQSeries`, this parameter is used on non-Windows platforms to create the queue manager and the queues. The default value for *JMSProviderUserID* is :

mqm

mqPwd *JMSProviderPassword*
Where *JMSProviderPassword* is the password for the user ID provided for mqUser. This parameter has no default value.

mqSchemaName *mqSchemaName*
Where *mqSchemaName* is the name of the database schema for the default JMS provider's messaging engine. The default value is BPEME. This option is only used when configuring Business Process Choreographer in a cluster and the mqType option is set to WPM.

mqCreateTables *{ true | false})*
This Boolean parameter controls whether the default JMS provider automatically creates its tables in the message engine database upon the first connection. The default value is true. This option is only used when configuring Business Process Choreographer in a cluster and the mqType option is set to WPM.

mqDataSource *datasourceName*
Where *datasourceName* is the JNDI name of the data source to be used by the default JMS provider's message engine. This must be a cluster-level data source in the WebSphere cluster identified by *mqClusterName*. The underlying database for the default JMS provider must be created manually. This option is only used when configuring Business Process Choreographer in a cluster and the mqType option is set to WPM.

shell *shell*
This parameter determines the shell that is used to run external commands. The default value is /bin/sh.

Running the configuration script interactively

This example, illustrates running the bpeconfig.jacl script to install and configure a business process container that uses an existing DB2 database, a human task container, and a Business Process Choreographer Explorer.

1. On the server, or for ND, on the deployment manager, start the script by entering the command:

```
install_root/bin/wsadmin.sh
      -f install_root/ProcessChoreographer/sample/bpeconfig.jacl
```

2. Interactively enter responses to the questions that are displayed:
   a. In an ND environment, you will be offered a cluster to configure in. If it is not the correct cluster, enter **No** to be offered the next cluster. If it is the correct cluster, enter **Yes**.
   b. For the question Install the business process container?, enter **Yes**.
   c. For the question User(s) to add to role BPESystemAdministrator, enter the user IDs for the users who will perform the role of business process administrator.
   d. For the question Group(s) to add to role BPESystemAdministrator, enter the groups from the domain user registry that are mapped onto the role of business process administrator.
   e. For the question User(s) to add to role BPESystemMonitor, enter the user IDs for the users who will perform the role of business process monitor.
   f. For the question Group(s) to add to role BPESystemMonitor, enter the groups from the domain user registry that are mapped onto the role of business process monitor.

g. If you get the question `Use WebSphere default messaging or WebSphere MQ`, enter one of the two displayed options.

h. For the question `Run-as UserId for role JMSAPIUser`, enter the run-as user ID that will be used for the JMSAPIUser role.

i. Enter the password for the run-as user ID.

j. For the question `Use WebSphere default messaging or WebSphere MQ [WPM/MQSeries]?`, select the JMS provider that you want to use.

k. For the question `Create the DataSource for the Process Choreographer database?`, enter **Yes**.

l. Enter the database name.

m. For the question `Universal or CLI?`, enter the type of the JDBC driver.

n. For the question `DB2 User ID`, enter the user ID used to create the database tables and schema.

o. For the question `Database server name (may be empty, set to use the type 4 driver)`, enter the name of the server that hosts the database.

p. `For the question Database server port`, enter the database server port, for example, 50000.

q. For the question `Create the Process Choreographer database?`, if your user ID has sufficient authority to create the database you can enter **Yes**, otherwise, if the database already exists, or if your user ID does not have sufficient authority to create the database, enter **No**.

r. For the question `DB2 tablespace directory (may be empty)` enter the directory for the table space, or leave it empty.

s. For the question `Create the ActivationSpecs for the business flow manager?`, enter **Yes** or **No**.

t. If you get the question `User ID for access to default messaging`, enter the user ID to use to access the default JMS provider.

u. If you get the question `Name of the message engine cluster`, enter the name of the message engine cluster.

v. If you get the question `Name of the message engine database schema`, enter the name of the message engine database schema.

w. If you get the question `Automatically create the message engine database tables [true/false]?`, enter `true` to automatically create the message engine database tables, otherwise enter `false`.

x. If you get the question `Message engine datasource JNDI name`, enter the JNDI name of the message engine data source.

y. For the question `Install the task container?`, enter **Yes**.

z. For the question `User(s) to add to role TaskSystemAdministrator`, enter the user IDs for the users who will perform the role of task administrator.

aa. For the question `Group(s) to add to role TaskSystemAdministrator`, enter the groups from the domain user registry that are mapped onto the role of task administrator.

ab. For the question `User(s) to add to role TaskSystemMonitor`, enter the user IDs for the users who will perform the role of task monitor.

ac. For the question `Run-as UserID for role EscalationUser`, enter the run-as user ID for the role of escalation user.

ad. For the question `Create the mail notification session for the human task manager?`, enter **No** if you do not want to create the mail notification session for the Human Task Manager. Otherwise, enter **Yes**, and specify the mail transport host and user ID.

ae. For the question `Create the ActivationSpecs for the human task manager?`, enter **Yes** to create J2EE ActivationSpecs for the Human Task Manager Message Driven Bean (MDB), otherwise enter **No**.

af. If you get the question `Configure in cluster 'MECluster' [Yes/no]?`, enter **Yes** to configure in the specified cluster, otherwise, enter **No**.

ag. If you get the question `Add JDBC provider permissions to server.policy [Yes/no]?`, enter **Yes** to automatically add the permissions for the JDBC provider to the server.policy file, otherwise, enter **No**.

ah. For the question `Install the Business Process Choreographer Explorer?`, enter **Yes** to install the Business Process Choreographer Explorer, then for the `Virtual host for the Business Process Choreographer Explorer`, enter the name of the virtual host for the Business Process Choreographer Explorer, for example, **default_host**, then for the question `Precompile JSPs?`, enter **Yes** if you want Java Server Pages (JSPs) to be precompiled, otherwise enter **No** .

ai. For the question `Create aliases for` *`your_server`* `in host` *`your_host`*`?`, enter **Yes** to create aliases for your server in the your virtual host, otherwise enter **No**.

aj. For the question `Enable global security using the Local OS user registry?`, enter **Yes** to enable global security using the local operating system user registry, otherwise, enter **No**.

ak. For the question `Server user ID`, enter the server user ID.

al. For the question `Enforce Java 2 security?`, enter **Yes** to enforce Java 2 security, otherwise, enter **No**.

am. For the question `Set 'com.ibm.SOAP.loginUserid' in soap.client.props?`, enter **Yes** to set the login user ID in the SOAP client properties, otherwise, enter **No**.

an. For the question `Delete the temporary directory?`, enter **Yes** to delete the temporary directory specified, otherwise, enter **No**.

3. In case of problems, check the log files.

Log files

If you have problems creating the configuration using the bpeconfig.jacl script file, check the following log files:

- bpeconfig.log
- wsadmin.traceout

Both files can be found in the logs directory for your profile:

- In the directory *`install_root`*`/profiles/`*`profileName`*`/logs/`.

Configuring the business process container using the installation wizard:

This describes how to create the necessary resources and then run the business process container installation wizard.

You must configure the necessary resources and install the business process container application before you can run applications that contain business processes or human tasks.

1. Make sure that the server is started and that you are logged on to the administrative console with a user ID with sufficient administration rights.

2. In the administrative console, select the server where you want to install the business process container. Click **Servers** → **Application Servers** → ***serverName***.

   Where *serverName* is the name of the application server where you want to install the business process container. In a cluster, you can select any application server, and the business process container is installed simultaneously on all application servers in the cluster.
3. Go to the Business Process Container settings. On the **Configuration** tab, under **Container Settings**, expand **Business process container settings**, and click **Business process container**.
4. Verify that the business process container is not installed. There should be a message indicating that the business process container is not currently installed. If the business process container is already installed, perform "Removing the Business Process Choreographer configuration" on page 237 before starting the installation wizard.
5. Start the installation wizard. In the **Additional Properties** section, click the link **Business process container installation wizard**.
6. Select the database configuration (wizard step 1):
   a. In the **JDBC Providers** drop-down list, select the entry with the database system, system version and Java Database Connectivity (JDBC) driver that you are using. Where possible, the installation wizard offers appropriate default values in the parameter fields. However, with some combinations of browser and platform, no defaults are provided. In this case, you can view the recommended values in "Business process container installation wizard settings" on page 156.
   b. For the **Implementation class name** use the default class name that is provided for the JDBC driver implementation.
   c. For **Classpath** enter the location of the Java archive (JAR) or the compressed file that contains the JDBC driver. To use the path variable that is displayed in the text field, it must be defined in **Environment** → **Manage WebSphere Variables**.
   d. The **Data source user name** must be a user ID that has the authority to connect to the database and to modify the data. If the user ID has the authority to create tables and indexes in the database, then the database schema will be updated automatically, when necessary, after applying a service or fix pack. This is not required for a Cloudscape database.
   e. Enter the **Data source password** for the data source user name. This is not required for a Cloudscape database.
   f. The **Custom properties** field contains default values for the database that you selected.
      - If you are using a Cloudscape database that is not in the default directory, change the value for the custom property **databaseName** to specify the fully qualified location of the database.
      - You might need to change or add some other properties. For more information, see the Installation wizard settings page and the product documentation for your database system.
   g. Click **Next** to go to the next step in the installation wizard.
7. Select the JMS provider and security (wizard step 2):
   a. In the drop-down list for **JMS provider**, select the messaging service for the business process container to use.
      - For default messaging, select `Default messaging provider`.
      - For WebSphere MQ, select `WebSphere MQ`.

b. Use the default value for **Queue Manager** (`BPC_nodeName_serverName`). If you are using the default messaging provider, this field is ignored.
c. If you are using the WebSphere MQ JMS provider and the WebSphere environment variable `${MQ_INSTALL_ROOT}` is not defined, make sure that the **Classpath** points to the WebSphere MQ Java `lib` directory. By default, `MQ_INSTALL_ROOT` is predefined with the value `${WAS_INSTALL_ROOT}/lib/WMQ`.
d. For the **JMS user ID**, enter a user ID that has administration rights for the messaging service. Use `root`
e. For the **JMS password**, enter the password for the JMS user ID.
f. For the **JMS API User ID**, enter a user ID from the user registry. This user ID will be used to process asynchronous API calls.
g. For the **JMS API Password**, enter the password for the JMS API User ID.
h. For the **Administrator security role mapping**, enter the name of the group, defined in the user registry, that will map onto the role of Business Process Administrator. On Windows systems, for example, you can specify the group Administrators.
i. For the **System monitor security role mapping**, enter the name of the group in the user registry to map onto the role of Business Process System Monitor. On Windows systems, for example, you can specify the group Administrators.
j. Click **Next** to go to the next step in the installation wizard.

8. Select the JMS Resources and Business Process Choreographer Explorer (wizard step 3): Either select **Create new JMS resources using default values**, or perform the following:
   a. Select **Select existing JMS resources**.
   b. Use the **Connection Factory** drop-down list to select **BPECF**.
   c. Use the **Internal Queue** drop-down list to select **BPEIntQueue**.
   d. Use the **External Request Processing Queue** drop-down list to select **BPEApiQueue**.
   e. Use the **Hold Queue** drop-down list to select **BPEHldQueue**.
   f. Use the **Retention Queue** drop-down list to select **BPERetQueue**.
9. **Optional:** To install **Business Process Choreographer Explorer**, select the check box; otherwise, clear the check box.
10. **Optional:** To use the Common Event Infrastructure, select **Enable Common Event Infrastructure logging for all processes running in this container**.
11. **Optional:** To enable the audit log, select **Enable audit logging for all processes running in this container**.
12. Click **Next** to view the summary (wizard step 4).
13. Check that the information on the summary page is correct. The summary includes reminders of which external resources are necessary. If you have not already created them, you can continue configuring the business process container, but you must create the resources before you activate the business process container. Printing the summary page helps you to create the correct resources.
    a. To make corrections, click **Previous**.
    b. To install the business process container and define its resources, click **Finish**. The progress is shown on the **Installing** page.
14. If the installation did not succeed, check for any error messages that can help you correct the problem, then try again.

15. If the installation succeeded, click **Save Master Configuration**, then click **Save**.
16. If you configured Business Process Choreographer in a cluster and you are using the WebSphere MQ JMS provider: Perform "Customizing the WebSphere MQ JMS resources in a cluster" on page 168.
17. Restart the application server.

The business process container is configured.

Continue configuring at step 136.

Creating the queue manager and queues for the business process container:

This describes how to create the WebSphere MQ queue manager and queues.

If you are using WebSphere

MQ as an external Java

Message Service (JMS) provider, you must create the queue manager and queues.

1. If you are not creating a WebSphere cluster setup, perform the following actions:
   a. Make sure that your user ID has the authority to create WebSphere MQ queues.
   b. Create the queue manager and queues: Type the following:

      ```
      cd install_root/ProcessChoreographer
      createQueues.sh queueManager
      ```

      where:

      queueManager
      is the name of an existing queue manager, or the name to give to a new queue manager. If the named queue manager already exists, it is used to create the queues. If the queue manager does not exist, it is created and started before the default queues are created.
2. If you are creating a WebSphere cluster setup that uses a WebSphere MQ cluster, only perform Creating clustered queue managers and queues.
3. If you are creating a WebSphere cluster setup that uses a central queue manager, perform the following actions:
   a. Copy the create queues script file from the `ProcessChoreographer` subdirectory of the *install_root* directory on the server that hosts WebSphere Process Server to the server that hosts the central queue manager: Copy the file:

      install_root`/ProcessChoreographer/createQueues.sh`
   b. On the server that hosts the queue manager, make sure that WebSphere MQ is installed, and that your user ID has the authority to create WebSphere MQ queues.
   c. Create the queue manager and queues: Type the following:

      ```
      cd install_root/ProcessChoreographer
      createQueues.sh queueManager
      ```

where *queueManager* is the name to give to the new queue manager.

d. Add a listener for the new queue manager by entering the command:

```
runmqlsr -t tcp -p port -m queueManager
```

Where *port* is the port on which the listener listens.

e. Add definitions for the port and queue manager service:

1) Add the port for the queue manager to the `/etc/services` file:

```
<Service:Name> <port>/tcp
  <Service:Name>    name of the queue manager service
  <port>            port for the queue manager
```

2) Add the service specified in the `/etc/services` file to the `/etc/inetd.conf` file:

```
<Service:Name> stream tcp nowait mqm /usr/mqm/bin/amqcrsta amqrsta
               -m QueueManager
  <Service:Name>    name of the queue manager service
  <Service:Name>    name of the queue manager
```

The queue manager and queues exist.

Creating clustered queue managers and queues for the business process container:

If you are creating a WebSphere

cluster setup of Business Process Choreographer using a WebSphere

MQ cluster, you must create the queue managers, queues, cluster, repositories, channels, and listeners.

1. Perform the following actions on each node:

a. Make sure that your user ID has the authority to create WebSphere MQ queues.

b. Create the get and put queue managers, make them members of the WebSphere MQ cluster, and create the queues by entering the commands:

```
cd install_root/ProcessChoreographer
createQueues.sh getQueueManager clusterName putQueueManagerName
```

where:

getQueueManager
The unique name to give to the get queue manager. This queue manager hosts all of the local queues.

clusterName
The name of the WebSphere MQ cluster.

putQueueManager
The unique name to give to the put queue manager.

If the queue managers already exist, they are used. If the queue managers do not exist, they are created and used.

c. Start the WebSphere MQ command processor by entering the command:

```
runmqsc getQueueManager
```

d. For complex setups, it is recommended to enable remote administration of the queue manager by entering the following MQ command:

```
DEFINE CHANNEL('SYSTEM.ADMIN.SVRCONN') TYPE(CHLTYPE)
```

e. If this queue manager is to be a repository for the WebSphere MQ cluster enter the MQ command:

```
ALTER QMGR REPOS('clusterName') REPOSNL(' ')
```

f. Define a sender and a receiver channel for the queue manager to each repository that is not hosted on this server, by entering the following MQ commands. For each cluster receiver channel:

```
DEFINE CHANNEL('TO.repositoryQueueManager.TCP') +
     CHLTYPE(CLUSRCVR) +
     CLUSTER('clusterName') +
     CLUSNL(' ') +
     CONNAME('repositoryIP-Address(port)') +
     DESCR('Cluster receiver channel at repositoryQueueManager TCPIP') +
     MAXMSGL(4194304) +
     TRPTYPE(TCP) +
     MCAUSER('principal') +
     REPLACE
```

For each cluster sender channel:

```
DEFINE CHANNEL('TO.repositoryQueueManager.TCP') +
     CHLTYPE(CLUSSDR) +
     CONNAME('repositoryIP-Address(port)') +
     CLUSTER('clusterName') +
     CLUSNL(' ') +
     DESCR('Cluster sender channel to repositoryQueueManager TCPIP') +
     MAXMSGL(4194304) +
     TRPTYPE(TCP) +
     MCAUSER('targetPrincipal') +
     REPLACE +
     NPMSPEED (NORMAL)
```

where:

repositoryQueueManager
: The name of the queue manager hosting a repository.

clusterName
: The name of the WebSphere MQ cluster of which all the queue managers are a member.

repositoryIP-Address
: The IP address of the node where the repository queue manager resides.

port
: The IP port that the repository queue manager is using.

principal, *targetPrincipal*
: The MCAUSER to use for the receive and send channels. For more information about this value, refer to the WebSphere MQ documentation.

g. For each queue manager, start a listener by entering the MQ command:

```
runmqlsr -t tcp -p port -m QueueManager
```

2. To verify the status of the channels on a server, enter the MQ command:

```
display chstatus(*)
```

The queue managers, queues, cluster, repositories, channels, and listeners exist.

Creating the database for the business process container:

The business process container requires a database. This topic describes how to create the database for Business Process Choreographer.

In a clustered Business Process Choreographer setup, one database serves all the business process containers in the WebSphere

cluster. In a non-clustered setup, the database is dedicated to the business process container on one application server.

1. On the server that hosts the database, create the database according to the description for your database system.
   - "Creating a Cloudscape database for Business Process Choreographer."
   - "Creating a DB2 UDB for z/OS database for Business Process Choreographer" on page 154.
2. On each server that runs Business Process Choreographer without a local database, you must make the remote database accessible:
   a. Install a suitable database client or Java Database Connectivity (JDBC) driver on the server that hosts the application server.
   b. If you are not using a type-4 JDBC driver, make the new database known to the database client as follows:

      For Cloudscape
      No action is required, because Business Process Choreographer supports only the embedded version of Cloudscape, which does not support remote access. The Cloudscape Network Server is not supported, because it has no XA support.

      For DB2 Universal Database™
      The database must be cataloged and accessible through an alias name.
   c. Test the connection to the database. Click **Resources → JDBC Providers → *provider_name* → Data sources**. Select the check box for the data source, and click **Test connection**.

The Business Process Choreographer database exists and is accessible from the servers that host the application server and the deployment manager.

Creating a Cloudscape database for Business Process Choreographer:

Use this task to create a Cloudscape database for Business Process Choreographer.

The Cloudscape

database system is implemented in the Java

language. It comes with the WebSphere

Process Server as several Java

Archive (JAR) files.

The Cloudscape license that comes with WebSphere Process Server is only for development and test, not for production purposes. Cloudscape cannot be used as database system for Business Process Choreographer in a Network Deployment environment. The Cloudscape version that comes with this product includes the Cloudscape Network Server that supports client/server JDBC access over the

Distributed Relational Database Architecture™ (DRDA®) protocol. Because the version of Cloudscape Network Server that is provided with this version of WebSphere Process Server has no XA support, Business Process Choreographer can only use the Embedded Cloudscape version that cannot be accessed remotely.

To create a Cloudscape database named `BPEDB`, perform the following actions:

1. Prepare to run the database creation script file by performing one of the following:
   - To prepare to create the database in the default location, manually create a `databases` subdirectory in the appropriate profile directory. Create *`install_root`*`/profiles/`*`Profile_name`*`/databases`. Change to the new directory.
   - To prepare to create a database location other than the default location, change to the directory where you want the new database created. If you run the business process container installation wizard, you must remember to specify the fully qualified database location as the value of the custom property `databaseName`.
2. Check whether you have Java configured on your server. Enter the command:

   ```
   java -version
   ```

   If you get an error message, then in step 5, when you run the database creation script, you must prefix the Java command with the full path to the Java executable, add the path *`install_root`*`/java/bin/`
3. Copy the database creation script to the current directory. Copy the file *`install_root`*`/ProcessChoreographer/createDatabaseCloudscape.ddl`
4. Read the instructions in the header of the database creation script, `createDatabaseCloudscape.ddl`, in an editor. On Windows systems, avoid using the Notepad editor, as it does not display the file in a readable format.
5. Run the database creation script file. Type the following:

   ```
   java -Djava.ext.dirs=install_root/cloudscape/lib
        -Dij.protocol=jdbc:db2j: com.ibm.db2j.tools.ij
        install_root/ProcessChoreographer/createDatabaseCloudscape.ddl
   ```

The Cloudscape

database for Business Process Choreographer exists.

Continue configuring at step 2.

Creating a DB2 UDB for z/OS database for Business Process Choreographer:

Use this task to create a DB2 UDB for z/OS database for Business Process Choreographer.

See DB2 Universal JDBC Driver Support for information on support for DB2 Universal JDBC Driver in WebSphere Process Server for z/OS.

When using DB2 UDB for z/OS, the following updates may be required:

- DB2 configuration parameters (zParms) need to be increased to support Business Process Choreographer LOBs.
  - _LOBVALA
  - _LOBVALS
- Required DB2 Conversion services:
  - CONVERSION 367,1208,ER;

– CONVERSION 1208,367,ER;

This topic describes how to create a DB2

UDB for z/OS

database and how to verify that it is reachable from the server that hosts the application server.

1. You must have already installed WebSphere Process Server on a z/OS server.
2. On the z/OS server that hosts the database:
   a. Log on the native z/OS environment.
   b. If multiple DB2 systems are installed, decide which subsystem you want to use.
   c. Make a note of the IP port to which the DB2 subsystem is listening.
   d. Using the DB2 administration menu, create a new database, for example, named `BPEDB`. Note the name of the database.
   e. Create a storage group and note the name.
   f. Decide which user ID is used to connect to the database from the remote server running WebSphere Process Server. Normally, for security reasons, this user ID is not the one that you used to create the database.
   g. Grant the user ID the rights to access the database and storage group. The user ID must also have permission to create new tables for the database.
   h. Decide if you want to create the tables and views in the schema of the connected user ID or if you want to customize the schema qualifier (`_SQLID`). If a single user ID accesses multiple databases with tables of the same name, you must use different schema qualifiers to avoid name collisions.
3. On the server that hosts the Process Server:
   a. Take note of the following information:

      An important difference exists between DB2 UDB and DB2 UDB for z/OS. DB2 UDB does not have the concept of a subsystem, but DB2 UDB for z/OS does. To avoid confusion between database name and subsystem name, it is important to understand that because DB2 UDB for z/OS runs in a subsystem, the catalog node and catalog database commands must identify the appropriate subsystem. On DB2 UDB, the subsystem name is not a known concept, so the database name that the catalog command makes a link to is really the name of the DB2 UDB for z/OS subsystem.
   b. Change to the *ProcessChoreographer* subdirectory in the installation root directory of the application server.
   c. Edit the `createTablespaceDb2zOSv8.ddl` script. Replace `@STOGRP@` with the storage group name and replace `@DBNAME@` with the database name (not the subsystem name).
   d. Run your customized version of the `createTablespaceDb2zOSv8.ddl` script, as described in the header of the script. If you want to drop the table space, use the `dropTablespaceDb2zOSv8.ddl` script.
   e. Edit the `createSchemaDb2zOSv8.ddl` script.
      1) Replace `@STOGRP@` with the storage group name.
      2) Replace `@DBNAME@` with the database name (not the subsystem name).
      3) Replace `@_SQLID@` with the schema qualifier or remove @_SQLID@ (including the following dot) from the script. A custom schema qualifier

can only be used with the DB2 Universal JDBC driver and requires that the configuration customSQLID property is set to the appropriate value.

4)

f. Run your customized version of the createSchemaDb2zOSv8.ddl script, as described in the header of the script. If this script does not work, or if you want to remove the tables and views, use the dropSchemaDb2zOSv8.ddl script to drop the schema, but replace @_SQLID@ before running the script.

The DB2 UDB for z/OS database for Business Process Choreographer exists.

Note: The SQL/DDL definitions are provided, they will need to be added to your DB2 environment manually.

Granting permission to the JDBC driver on the deployment manager:

Use this task when using Java 2 security in a Network Deployment (ND) environment to grant the required permissions to the Java Database Connectivity (JDBC) driver.

1. If you are using Java 2 security in an ND environment you might have to grant the required permissions to the Java Database Connectivity (JDBC) driver. Depending on the JDBC provider, you might need to manually update the server.policy file on the deployment manager. If you are using one of the database drivers listed in the following table, make sure that the server.policy file on the deployment manager contains the template text for your JDBC driver.
   - The server.policy file is located in *install_root*/profiles/*deployment_manager_profile*/properties/.

JDBC driver	Template

DB2 Universal JDBC Driver on z/OS	`// DB2 Universal JDBC Driver Provider`
	`grant codeBase "file:/${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar" {`
	`  permission java.security.AllPermission;`
	`};`
	`grant codeBase "file:/${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar" {`
	`  permission java.security.AllPermission;`
	`};`
	`grant codeBase "file:/${DB2UNIVERSAL_JDBC_DRIVER_PATH}/`
	`  db2jcc_license_cisuz.jar" {`
	`  permission java.security.AllPermission;`
	`};`

Important: Make sure that you always use forward slashes (/), and resolve any WebSphere variables in the template. Any syntax errors in the server.policy file can cause the deployment manager to fail to start.

2. Restart the deployment manager.

The JDBC driver works in an ND environment that has Java 2 security enabled.

Continue configuring at step 1 on page 147.

Business process container installation wizard settings:

Use the installation wizard to install and configure the business process container.

Access the business process container installation wizard by clicking **Servers** → **Application servers** → *server_name* → **Business Process Container Settings** → **Business process container** → **Business process container installation wizard**. This page describes the installation wizard fields, in the order that they display in the wizard.

Step 1 database configuration:

- JDBC provider
- Implementation class name
- Class path (for data source)
- Data source user name
- Data source password
- Custom properties

Step 2 JMS provider and security:

- JMS provider
- Queue manager
- Class path (for JMS provider)
- JMS user ID
- JMS password
- JMS API user ID
- JMS API password
- Administrator security role mapping
- System monitor security role mapping

Step 3 JMS resources and Business Process Choreographer Explorer:

- JMS resources (new or existing)
- Connection factory
- Internal queue
- External request processing queue
- Hold queue
- Retention queue
- Install Business Process Choreographer Explorer
- Enable CEI logging
- Enable audit logging

Attention: After the container is configured, you can only change the logging options. If you want to change any of the other values, you must remove the existing Business Process Choreographer configuration and then create a new one.

JDBC provider:

You must create a new data source that is only used by Business Process Choreographer.

Type	**Value**
Mandatory	Yes
Data type	Drop-down list

Type	Value
Choices for z/OS	Create a new XA data source for z/OS: • Cloudscape 5.1 (Cloudscape JDBC Provider (XA)) • DB2 Universal JDBC Driver on z/OS (type 2) **Note:** If this driver is not selectable from the drop down list, configure this data source as described in "Creating a DB2 UDB for z/OS database for Business Process Choreographer" on page 154.

Implementation class name:

The Java class name of the Java Database Connectivity (JDBC) driver implementation.

Type	Value
Mandatory	Yes
Data type	String
Default for Cloudscape 5.1 (Cloudscape JDBC Provider (XA))	`com.ibm.db2j.jdbc.DB2jXADataSource`
Default for DB2 Universal JDBC Driver on z/OS (type 2)	`com.ibm.db2.jcc.DB2ConnectionPoolDataSource`

For more information about properties and settings for the database, refer to Vendor-specific data sources minimum required settings.

class path (data source):

The path to the Java archive (JAR) file or zip file that contains the Java Database Connectivity (JDBC) driver. The JDBC driver provides the data source implementation class. If the database is remote, this path indicates where the JDBC driver is installed on the client computer.

Type	Value
Mandatory	**For Cloudscape** No, the JDBC driver is already on the WebSphere classpath. **For DB2 Universal JDBC Driver on z/OS** Yes
Data type	String
Default for Cloudscape 5.1	`${CLOUDSCAPE_JDBC_DRIVER_PATH}/db2j.jar` The value for `${CLOUDSCAPE_JDBC_DRIVER_PATH}` is predefined and does not need to be set.

Type	**Value**

Default for DB2 Universal JDBC Driver on z/OS 7 (type 2)	`${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar`
	`${UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar`
	`${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cisuz.j`
	
	The value for `${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar`depends on the installation root directory of the corresponding DB2 Client or DB2 Connect, and must be set in **Environment> Manage WebSphere Variables**. Typical values are:
	
	On z/OS:
	`/home/db2inst1/sqllib/java`

Data source user name:

A user ID that has the authority to connect to the database and to modify the data. If the user ID has the authority to create tables and indexes in the database, then the database schema will be updated automatically, when necessary, after applying a service or fix pack.

Type	**Value**
Mandatory	**For Cloudscape** No **For DB2 Universal JDBC Driver on z/OS** Yes
Data type	String
Default	The user ID that is currently logged on to the administrative console.

Data source password:

The password for the data source user ID.

Type	**Value**
Mandatory	**For Cloudscape** No **For DB2 Universal JDBC Driver on z/OS** Yes
Data type	String
Default	None

Custom properties:

Extra parameters that are required by the database system.

CAUTION:
It is not recommended that you change any of the optional properties before you have configured and verified that your business process container is working. Making such changes belongs to advanced tuning and troubleshooting, and can cause your system to stop working.

Type	**Value**
Mandatory	Yes
Data type	String
Data format	Multiple lines of *Property=Value*
Minimum required properties	Refer to Vendor-specific data sources minimum required settings.
Properties that are not listed in this table	Properties that are optional or that are ignored are not listed in this table. For information about such properties, refer to the documentation for your JDBC provider.
Required properties	All of the required properties for each JDBC provider are described below.
Required properties for Cloudscape	**databaseName = ${USER_INSTALL_ROOT}/databases/BPEDB** Required string. Defines which database to access. The value must be a fully qualified path.

Type	Value
Properties for DB2 z/OS 7 (DB2 Universal JDBC Driver Provider)	**databaseName=BPEDB** Required string. For DB2 UDB it defines which database to access. For DB2 z/OS it defines which subsystem contains the DB2 z/OS database. **driverType=2** Required integer. The JDBC connectivity-type of a data source. The only permitted value is 2. **serverName=""** Optional string. The TCP/IP address or host name for the DRDA server. **portNumber=50000** Optional integer. The TCP/IP port number where the DRDA server resides. **enableSQLJ=false** Optional boolean. This value is used to indicate whether SQLJ operations may be performed with this data source. If enabled, this data source can be used for both JDBC and SQLJ calls. Otherwise, only JDBC usage is permitted. **description=DataSource for Business Process Choreographer** Optional string. Description of the data source. Not used by the data source object. Used for information purposes only. **fullyMaterializeLobData=true** Optional boolean. This setting controls whether or not LOB locators are used to fetch LOB data. If enabled, LOB data is not streamed, but is fully materialized with locators when the user requests a stream on the LOB column. The default value is true. **resultSetHoldability=2** Optional integer. Determine whether ResultSets are closed or kept open when committing a transaction. The possible values are: 1 (HOLD_CURSORS_OVER_COMMIT), 2 (CLOSE_CURSORS_AT_COMMIT). **currentPackageSet=""** Optional string. This property is used in conjunction with the DB2Binder - collection option which is given when the JDBC/CLI packageset is bound during installation by the DBA. **readOnly=false** Optional boolean. This property creates a read only connection. **deferPrepares=false**

JMS provider:

Specifies which messaging service the business process container uses.

Type	**Value**
Mandatory	Yes
Data type	Drop-down list
Choices	`Default messaging provider` `WebSphere MQ`

Queue manager:

The name of the queue manager that is used by the business process container.

Type	**Value**
Mandatory	If you selected **`WebSphere MQ JMS Provider`**; otherwise, this field is disabled.
Data type	String
Value	Your queue manager name, for example, `BPC_`*`nodeName_serverName`*.

class path (JMS provider):

The path to the MQ Java `lib` directory.

Type	**Value**
Mandatory	If the WebSphere environment variable `${MQ_INSTALL_ROOT}` is not defined to point to the WebSphere MQ installation root directory.
Enabled	If you selected `WebSphere MQ JMS Provider`; otherwise, this field is disabled.
Data type	String
Default	The default value for the class path depends on the local MQ installation: **For z/OS** `/opt/mqm/java/lib`

JMS user ID:

Used to authenticate the connection to the Java Message Service (JMS) provider. This user ID must have administration rights for the messaging service.

Type	**Value**
Mandatory	Yes
Data type	String
Restrictions	If you are using WebSphere default messaging, the JMS user ID must be less than or equal to 12 characters.
Default	The user ID that you used to log into the administrative console.
For z/OS	Use `root`. The user ID must be a member of the group `mqm`.

JMS password:

The password for the Java Message Service (JMS) user ID.

Type	**Value**
Mandatory	If you selected **WebSphere JMS Provider**; otherwise, this field is disabled.
Data type	String
Default	None

JMS API user ID:

The user ID that the business process container message-driven bean (MDB) uses when processing asynchronous API calls.

Type	**Value**
Mandatory	If WebSphere security is enabled, even if you do not use the Java Message Service API.
Data type	String
Description	If WebSphere security is enabled and you do not use the JMS API, you must specify a valid user ID. This ID does not need any special authorizations. If WebSphere security is enabled and you plan to use the JMS API, this user ID must either be one that is given the appropriate authorities when the process is modeled, or more commonly, it must be a member of a group that was granted the necessary authorities during modeling. The possible staff authorities associated with processes are: Administrator, Reader, and Starter. For activities, a user ID can only perform the sendEvent action if it is a potential owner of the associated receiveEvent. If you want to support all the actions on processes through the JMS API, you can specify a user ID that is a member of the J2EE BPESystemAdministrator role. However, in a production system, the more fine-grained security approach is recommended. **Setting up Roles using RACF security:** These RACF permissions apply when the following security field is specified: • `com.ibm.security.SAF.delegation= true` `RDEFINE EJBROLE JMSAPIUser UACC(NONE) APPLDATA(' userid')`

JMS API password:

The password for the JMS API User ID.

Type	Value
Mandatory	If WebSphere security is enabled (even if you do not use the JMS API)
Data type	String

Administrator security role mapping:

The group from the domain user registry that is mapped onto the role of business process administrator.

Type	Value
Mandatory	Yes
Data type	String
Default	None

Restrictions	The group specified must already exist in the domain user registry. The user registry can be the local operating system, Lightweight Directory Access Protocol (LDAP), or custom registry.
	
	Setting up Roles using RACF security: These RACF permissions apply when the following security field is specified:
	• `com.ibm.security.SAF.authorization=true`
	`RDEFINE EJBROLE BPESystemAdministrator UACC(NONE)`
	`PERMIT BPESystemAdministrator CLASS(EJBROLE) ID(userid) ACCESS(READ)`

System monitor security role mapping:

The group from the domain user registry that is mapped onto the role of business process monitor.

Type	Value
Mandatory	Yes
Data type	String
Default	None

Restrictions	The group specified must already exist in the domain user registry. The user registry can be the local operating system, Lightweight Directory Access Protocol (LDAP), or custom registry.
	
	Setting up Roles using RACF security: These RACF permissions apply when the following security fields are specified:
	• `com.ibm.security.SAF.authorization=true`
	`RDEFINE EJBROLE BPESystemMonitor UACC(NONE)`
	`PERMIT BPESystemMonitor CLASS(EJBROLE) ID(userid) ACCESS(READ)`

JMS resources (new or existing):

You must either create new Java Message Service (JMS) resources or select existing JMS resources.

Type	**Value**
Mandatory	Yes
Data type	Radio buttons
Choices	• Create new JMS resources using default values. • Use existing JMS resources. Default values for JMS resources: **Connection factory:** BPECF **Internal queue** BPEIntQueue **External request processing queue** BPEApiQueue **Hold queue** BPEHldQueue **Retention queue** BPERetQueue

Connection factory:

The queue connection factory for the business process container to use.

Type	**Value**
Mandatory	Only if you chose **Select existing JMS resources**
Data type	Drop-down list
Default	`BPECF`

Internal queue:

The JNDI name of the queue for internal business process container messages.

Type	**Value**
Mandatory	Only if you chose **Select existing JMS resources**
Data type	Drop-down list
Default	`BPEIntQueue`

External request processing queue:

The JNDI name of the queue for external (JMS API) requests to the business process container.

Type	**Value**
Mandatory	Only if you chose **Select existing JMS resources**
Data type	Drop-down list

Type	Value
Default	`BPEApiQueue`

Hold queue:

The JNDI name of the queue that holds any messages that failed processing more times than the retry limit.

Type	Value
Mandatory	Only if you chose **Select existing JMS resources**
Data type	Drop-down list
Default	`BPEHldQueue`

Retention queue:

The JNDI name of the queue that contains messages that cannot be processed currently, and that require a retry later.

Type	Value
Mandatory	Only if you chose **Select existing JMS resources**
Data type	Drop-down list
Default	`BPERetQueue`

Business Process Choreographer Explorer:

If this check box is selected, the Business Process Choreographer Explorer is also installed.

Type	Value
Data type	Check box
Default	selected

Enable Common Event Infrastructure logging:

Common Event Infrastructure (CEI) logging can be enabled or disabled.

Type	Value
Data type	Check box
Default	Not selected

Enable Audit Logging:

Audit logging can be enabled or disabled.

Type	Value
Data type	Check box
Default	Not selected

Business process container settings:

Use this panel to manage business process containers.

A business process container provides services to run business processes within an application server. To view this administrative console page, click **Servers** → **Application Servers** → *server_name* → **Container Settings** → **Business Process Container**.

Enable Common Event Infrastructure logging:

Common Event Infrastructure (CEI) logging can be enabled or disabled.

Type	**Value**
Data type	Check box
Default	Not selected

Enable Audit Logging:

Audit logging can be enabled or disabled.

Type	**Value**
Data type	Check box
Default	Not selected

Retry Limit:

Specifies the maximum number of retries for processing a message. When the limit is reached, the message is sent to the Listener Port for Unprocessed Messages.

Type	**Value**
Data type	Integer
Default	5
Range	2 to 10 (recommended)

Retention Queue Message Limit:

The maximum number of messages that can be stored in the retention queue. When the limit is reached, the messages are sent to the queue for internal messages again and the process container switches into quiesce mode.

Type	**Value**
Data type	Integer
Default	20

Retention Queue:

The JNDI name of the queue that contains messages that cannot be processed currently, and that require a retry later.

Type	**Value**
Data type	Read-only string
Default	jms/BPERetQueue

Hold Queue:

The JNDI name of the queue that holds any messages that failed processing more times than the retry limit.

Type	Value
Data type	Read-only string
Default	jms/BPEHldQueue

Customizing the WebSphere MQ JMS resources in a cluster:

Use this task to customize the connection factory resources for business process containers that are in a cluster and use the WebSphere MQ JMS provider.

Do not perform this task if you are using default messaging. If you are using the WebSphere MQ JMS provider, perform the following steps for each application server in the cluster:

1. Open the connection factory page: Click **Resources → JMS Providers → WebSphere MQ → Scope: Server → Apply → WebSphere MQ connection factories**.
2. Select the business process container connection factory **BPECF** and set the property values for the type of queue manager configuration that you are using:
   - For a central queue manager:

Property	Description
Host	The host name of the server that is hosting the central queue manager.
Port	The port number that the central queue manager is using.
Transport Type	`Client`
Client ID	The message channel agent (MCA) user ID to use. This is normally the owner or creator of the queue manager, typically this is the root user..
CCSID	Use the value `819` .

   - For a cluster of queue managers:

Property	Description
Transport Type	`Bindings` or `Client`
Queue Manager	The name of the server get queue manager.

     When using WebSphere MQ, the local bindings transport type is slightly faster than using the client transport type, but has the effect that you must stop the entire application server to stop the local WebSphere MQ queue manager. If you specify `Client`, you must also provide the host name and port number for the get queue manager.
3. Select the business process container connection factory **BPECFC** and set the property values for the type of queue manager configuration you are using:
   - For a central queue manager:

Property	Description
Host	The host name of the server that is hosting the central queue manager.
Port	The port number that the central queue manager is using.
Transport Type	`Client`

Property	Description
Client ID	The message channel agent (MCA) user ID to use. This is normally the owner or creator of the queue manager, typically this is the root user.
CCSID	Use the value `819`

- For a cluster of queue managers:

Property	Description
Host	The host name of the application server node.
Port	The port number used by the put queue manager of this application server's .
Transport Type	`Client`
Client ID	The message channel agent (MCA) user ID to use. This is normally the owner or creator of the queue manager, typically this is the root user.
CCSID	`819`

4. Select the human task manager connection factory **HTMCF** and set the property values for the type of queue manager configuration that you are using:
   - For a central queue manager:

Property	Description
Host	The host name of the server that is hosting the central queue manager.
Port	The port number that the central queue manager is using.
Transport Type	`Client`
Client ID	The message channel agent (MCA) user ID to use. This is normally the owner or creator of the queue manager, typically this is the root user..
CCSID	Use the value `819` .

   - For a cluster of queue managers:

Property	Description
Transport Type	`Bindings` or `Client`
Queue Manager	The name of the server get queue manager.

When using WebSphere MQ, the local bindings transport type is slightly faster than using the client transport type, but has the effect that you must stop the entire application server to stop the local WebSphere MQ queue manager. If you specify `Client`, you must also provide the host name and port number for the get queue manager.

The connection factories for the business process containers have been installed in the cluster and are configured.

Continue configuring at step 17 on page 150.

Configuring the human task container, using the installation wizard:

Use this task to configure the human task container.

Before configuring the human task container, see information on configuring the business process container manually in "Configuring Business Process Choreographer" on page 136.

If you have run the bpeconfig.jacl script, the human task container is already configured. The following steps describe how to configure the human task container using the installation wizard.

1. In the administrative console, click **Servers** → **Application servers** → *Server_Name*. Then in the **Container Settings** section, click **Human task container settings** → **Human task container** → **Human task container installation wizard** (in the **Additional Properties** section). Where possible, the installation wizard offers appropriate default values in the parameter fields, you can view the recommended values on the "Human task container installation wizard settings" on page 171.
2. Verify that the human task container is not configured. There should be a message indicating that the Human Task Manager is not currently installed. If the human task container is already configured, remove the configuration before you start the installation wizard. For details about how to remove the configuration, see "Removing the Business Process Choreographer configuration" on page 237.
3. Select the JMS provider and security settings (step 1):
   a. In the drop-down list for **JMS provider**, select the messaging service that is used by the business process container.
      - For default messaging, select `Default messaging provider`.
      - For WebSphere MQ, select `WebSphere MQ`.
   b. Use the default value for **Queue Manager** (`BPC_`*`nodeName_serverName`*). If you are using the default messaging provider, this field is ignored.
   c. If you are using external messaging (WebSphere MQ JMS provider) and you have not defined the WebSphere environment variable `${MQ_INSTALL_ROOT}`, make sure that **Classpath** points to the WebSphere MQ Java `lib` directory.
   d. For the **JMS user ID**, enter a user ID that has administration rights for the messaging service. This user ID will be used to connect to the JMS queue manager. Use `root`
   e. For the **JMS password**, enter the password for the JMS user ID.
   f. For **Escalation user ID**, enter the user ID that will be used by the human task container to perform scheduled actions, for example triggering escalations to verify the expected task state, timed task deletion, and task expiration. Use `root`
   g. For **Escalation password**, enter the password for the escalation user ID.
   h. For the **Administrator security role mapping**, enter the name of the group, defined in the user registry, that will map onto the role of Business Process Administrator. On Windows systems, for example, you can specify the group Administrators.
   i. For the **System monitor security role mapping**, enter the name of the group in the user registry to map onto the role of Business Process System Monitor. On Windows systems, for example, you can specify the group Administrators.
   j. Click **Next** to go to the next step in the installation wizard.

4. Select the JMS Resources (step 2): If you are not reusing an existing JMS connection factory and queues, which is normally the case, select **Create new JMS resources using default values**. Otherwise, perform the following:
   a. Select **Select existing JMS resources**.
   b. Use the **Connection Factory** drop-down list to select the connection factory you want to use.
   c. Use the default **Hold Queue** value **HTMHldQueue**.
5. **Optional:** Select **Mail session** to create the default mail session resource with cell scope, named mail/HTMNotification_*nodeName_serverName*.

   Attention: If this is not set, no escalation mails are sent.
6. **Optional:** To use the Common Event Infrastructure, select **Enable Common Event Infrastructure logging** .
7. **Optional:** To enable the audit log, select **Enable audit logging for all human tasks** .
8. Click **Next** to view the Summary (step 3).
9. Check that the information on the summary page is correct. The summary includes reminders of which external resources are necessary. If you have not already created them, you can continue configuring the human task container, but you must create the resources before you activate the human task container. Printing the summary page helps you to create the correct resources.
   a. To make corrections, click **Previous**.
   b. To install the human task container and define its resources, click **Finish**. The progress is shown on the Installing page.
   c. Verify that no error messages are displayed.
10. If you selected the Mail session option in step 5, you must set the mail transport host:
    a. Click **Resources** → **Mail Providers**.
    b. Select the cell scope: **Built-in Mail Provider**.
    c. Under **Mail sessions**, click **HTMMailSession_*nodeName_serverName*** and set the **Mail transport host**.
    d. If the mail transport host is secured, also set **Mail transport user ID** and **Mail transport password**.
    e. Click **OK**.
11. Click **Save Master Configuration**, then click **Save**.
12. Restart the application server.
13. If the container did not install successfully, check for any error messages that can help you correct the problem, then repeat this task.

    Check the administrative console or the `SystemOut.log` file for the application server. On a cluster, check the log for all application servers in the cluster.

The human task container is configured.

Continue configuring at step 3 on page 137.

Human task container installation wizard settings:

Use the installation wizard to install and configure the human task container.

Access the human task container installation wizard by clicking **Servers** → **Application servers** → ***server_name***. Then in the **Container Settings** section, click

Human task container settings → Human task container → Human task container installation wizard. This page describes the installation wizard fields, in the order in which they are displayed in the wizard.

Step 1 JMS provider and security:

- JMS provider
- Queue manager
- Class path
- JMS user ID
- JMS password
- Escalation user ID
- Escalation password
- Administrator security role mapping
- System monitor security role mapping

Step 2 JMS resources:

- JMS resources (new or existing)
- Connection factory
- Hold queue
- Mail session
- Enable CEI logging
- Enable audit log

Attention: After applying these fields, you can only enable and disable the logging options.

JMS provider:

Specifies which messaging service the human task container uses.

Type	**Value**
Mandatory	Yes
Data type	Drop-down list
Choices	`WebSphere MQ` `Default messaging provider`

Queue manager:

The name of the queue manager that is used by the human task container.

Type	**Value**
Mandatory	If you selected **`WebSphere MQ JMS Provider`**; otherwise, this field is disabled.
Data type	String
Value	Your queue manager name, for example, `BPC_`*`nodeName_serverName`*.

Class path:

The path to the MQ Java `lib` directory.

Type	Value
Mandatory	If the WebSphere environment variable `${MQ_INSTALL_ROOT}` is not defined to point to the WebSphere MQ installation root directory.
Enabled	If you selected `WebSphere MQ JMS Provider`; otherwise, this field is disabled.
Data type	String
Default	The default value for the class path depends on the local MQ installation: **For z/OS** `/opt/mqm/java/lib`

JMS user ID:

Used to authenticate the connection to the Java Message Service (JMS) provider. This user ID must have administration rights for the messaging service. It will be used to connect to the JMS queue manager.

Type	Value
Mandatory	Yes
Data type	String
Restrictions	If you are using WebSphere default messaging, the JMS user ID must be less than or equal to 12 characters.
Default	The user ID that you used to log into the administrative console.
For z/OS	Use `root`. The user ID must be a member of the group `mqm`.

JMS password:

The password for the Java Message Service (JMS) user ID.

Type	Value
Mandatory	If you selected **`WebSphere JMS Provider`**; otherwise, this field is disabled.
Data type	String
Default	None

Escalation user ID:

A user ID used by the human task container to perform scheduled actions.

Type	Value
Mandatory	Yes, even if no escalation mails will be sent.
Data type	String
Description	This is the run-as user ID for the Human Task Manager MDB to perform scheduled escalation, deletion and expiration actions.

Escalation password:

The password for the escalation user ID.

Type	Value
Mandatory	Yes
Data type	String

Administrator security role mapping:

The group from the user registry that is mapped onto the role of task administrator.

Type	Value
Mandatory	Yes
Data type	String
Default	None
Restrictions	The user registry can be the local operating system, Lightweight Directory Access Protocol (LDAP), or custom registry. The group that is specified must already exist in the user registry being used.

System monitor security role mapping:

The group from the user registry that is mapped onto the role of task monitor.

Type	Value
Mandatory	Yes
Data type	String
Default	None
Restrictions	The user registry can be the local operating system, Lightweight Directory Access Protocol (LDAP), or custom registry. The group that is specified must already exist in the user registry being used.

JMS resources (new or existing):

You must either create new Java Message Service (JMS) resources or select existing JMS resources.

Type	Value
Mandatory	Yes
Data type	Radio buttons
Choices	• Create new JMS resources using default values. • Use existing JMS resources. Default values for JMS resources: **Connection factory:** HTMCF **Hold queue** HTMHldQueue

Connection factory:

The queue connection factory for the human task container to use.

Type	Value
Mandatory	Only if you chose **Select existing JMS resources**
Data type	Drop-down list
Default	`HTMCF`

Hold queue:

The JNDI name of the queue that holds any messages that could not be processed.

Type	Value
Mandatory	Only if you chose **Select existing JMS resources**
Data type	Drop-down list
Default	`HTMHldQueue`

Mail Session:

If you select the mail session check box, a mail session with the name mail/HTMNotification_*nodeName_serverName* will be created with cell scope. This is necessary for sending escalation mails.

Type	Value
Data type	Check box
Default	Not selected

Enable Common Event Infrastructure logging:

Common Event Infrastructure (CEI) logging can be enabled or disabled.

Type	Value
Data type	Check box
Default	Not selected

Enable Audit Logging:

Audit logging can be enabled or disabled.

Type	Value
Data type	Check box
Default	Not selected

Human task container settings:

Use this panel to manage human task containers.

A human task container provides services to run human task within an application server. To view this administrative console page, click **Servers** → **Application Servers** → ***server_name*** → **Human Task Container**.

Enable Common Event Infrastructure logging:

Common Event Infrastructure (CEI) logging can be enabled or disabled.

Type	Value
Data type	Check box
Default	Not selected

Enable Audit Logging:

Audit logging can be enabled or disabled.

Type	Value
Data type	Check box
Default	Not selected

Mail Session:

The Java Naming and Directory Interface (JNDI) name of the mail session resource that will be used by the human task container to send escalation mails.

Type	Value
Data type	Read-only string
Default	mail/HTMNotification_*nodeName_serverName*

Configuring the LDAP staff plug-in provider:

Use this task to configure the LDAP staff plug-in provider that Business Process Choreographer uses to determine who can start a process or claim an activity or a task.

Each type of supported user directory service requires a corresponding staff plug-in. The following staff plug-ins are supported:

Table 15. Supported staff plug-in providers

User directory service	Plug-in provider
Lightweight Directory Access Protocol (LDAP)	`LDAP Staff Plug-in Provider`
Local operating system user registry	`System Staff Plug-in Provider`
WebSphere Application Server user registry	`User Registry Staff Plug-in Provider`

All of these plug-ins are already installed. You can use the user registry and system plug-ins without any configuration.

The LDAP staff plug-in is configured for an LDAP server with anonymous access; the LDAP server is local to the installed application server. You can change the configuration of the LDAP plug-in.

1. In the administrative console, click **Resources** → **Staff Plugin Provider**.
2. If the scope is not set to Node, select **Node** and click **Apply**.
3. To create a new LDAP configuration:
   a. Click the name of the LDAP staff plug-in provider.
   b. Select **Staff Plugin Configuration**.
   c. Click **New** → **Browse**, and select the sample Extensible Stylesheet Language (XSL) transformation file to use. The standard XSL transformation for LDAP is located in*install_root*`/ProcessChoreographer/Staff/`

`LDAPTransformation.xsl` Do not modify this transformation file. If you need to customize the transformations to match the LDAP schema of your organization, modify a copy that has a different file name.

d. Click **Next**.

e. Enter an administrative name for the staff plug-in provider.

f. Enter a description.

g. Enter the Java Naming and Directory Interface (JNDI) name for business processes to use in referencing this plug-in, for example, `bpe/staff/ldapserver1`

h. Click **Apply**.

i. Click **Custom Properties**.

j. For each of the required properties and for any optional properties that you want to set, click the name of the property, enter a value, and click **OK**.

k. To apply the changes, click **Save**. This table describes each property for the LDAP plug-in.

LDAP plug-in property	Required or optional	Comments
AuthenticationAlias	Optional	The authentication alias used to connect to LDAP, for example, `mycomputer/My LDAP Alias`. You must define this alias in the administrative console by clicking **Security** → **JAAS** → **Configuration JAAS Configuration** → **J2C Authentication Data**. If this alias is not set, anonymous logon to the LDAP server is used.
AuthenticationType	Optional	If the AuthenticationType property is not set, the default logon is anonymous authentication. In all other cases, the default is simple authentication.
BaseDN	Required	The base distinguished name (DN) for all LDAP search operations, for example, `"o=mycompany, c=us"`
CasesentivenessForObjectclasses	Optional	Determines whether the names of LDAP object classes are case-sensitive.
ContextFactory	Required	Sets the Java Naming and Directory Interface (JNDI) context factory, for example, `com.sun.jndi.ldap.LdapCtxFactory`
ProviderURL	Required	This Web address must point to the LDAP JNDI directory server and port. The format must be in normal JNDI syntax, for example, `ldap://localhost:389`
SearchScope	Required	The default search scope for all search operations. Determines how deep to search beneath the baseDN property. Specify one of the following values: `objectScope`, `onelevelScope`, or `subtreeScope`
additionalParameterName1-5 and additionalParameterValue1-5	Optional	Use these name-value pairs to set up to five arbitrary JNDI properties for the connection to the LDAP server.

4. To activate the plug-in, stop and start the server.
5. If you have problems with any of these steps, refer to troubleshooting the staff service and staff plug-ins.

Processes can now use the staff support services to resolve staff queries, and to determine which activities can be performed by certain people.

Continue configuring at step .

Staff service settings:

Use this page to enable or disable the staff service, which manages staff plug-in resources used by the server.

To view this administrative console page, click **Servers** → **Application Servers** → ***server_name***. Then click **Business Integration** → **Staff Service**.

Enable service at server startup:

Specifies whether the server attempts to start the staff service.

Type
Value

Default
Selected

Range

Selected
When the application server starts, it attempts to start the staff service automatically.

Cleared
The server does not try to start the staff service. If staff plug-in resources are used on this server, the system administrator must start the staff service manually or select this startup property and then start the server again.

Staff plugin provider collection:

A staff plug-in is responsible for retrieving user information. Use this panel to manage staff plug-in providers.

To view this administrative console panel, click **Resources** → **Staff Plugin Provider**. Existing plug-in providers are displayed. Click **New** to configure a new provider, or click on the name of an existing provider to view or change its properties.

Name:

The name by which the staff plug-in provider is known for administrative purposes.

Type
Value

Data type
String

Description:

A description of the staff plug-in provider.

Type
Value

Data type
String

Staff plug-in provider settings:

Use this panel to modify the settings for a staff plug-in provider.

Staff plug-ins are used to get information from a directory of users. Each staff plug-in provider is registered with the run-time environment by specifying a name and a Java archive (JAR) file containing the plug-in. A configuration file in the JAR file defines the class name, which represents the plug-in as well as the properties for the plug-in.

To view this administrative console page, click **Resources** → **Staff Plugin Provider** → ***staffpluginprovider_name***.

To inspect or change the staff plug-in configuration and any custom properties, click on the name of the plug-in.

Scope:

The scope for this staff plug-in provider.

Type	**Value**
Data type	String
Valid values	Cell, node, or server.
Description	The scope determines the level at which the resource definition is visible. Use Cell scope for staff plug-in configurations because there are settings that are specific to a cell.

Name:

The name by which the staff plug-in provider is known for administrative purposes.

Type	**Value**
Data type	String

Description:

A description of the staff plug-in provider.

Type	**Value**
Data type	String

JAR File:

The file name, including the absolute path, of the JAR file containing the plug-in.

Type	**Value**
Data type	String

Staff plug-in configuration collection:

Use this page to manage staff plug-in configurations.

A staff plug-in configuration is defined for a staff plug-in provider. The staff plug-in configuration can define any custom properties specified by the staff plug-in provider. Each staff plug-in provider can have multiple staff plug-in

configurations. Click **New** to create a new configuration, or click on the name of an existing configuration to view or change its properties.

To view this administrative console page, click **Resources** → **Staff Plug-in Provider** → ***staffpluginprovider_name*** → **Staff Plug-in Configuration**.

Name:

The name of the staff plug-in configuration used for administrative purposes. Click on the name to view or change its configuration settings.

Type	**Value**
Data type	String

Description:

A description of the staff plug-in configuration.

Type	**Value**
Data type	String

JNDI Name:

The Java Naming and Directory Interface (JNDI) name used to look up the staff plug-in configuration in the namespace.

Type	**Value**
Data type	String

XSL Transform File:

The file name, including the absolute path, of the Extensible Style Language (XSL) transformation file.

Type	**Value**
Data type	String

Staff plugin configuration settings:

Use this page to modify the settings for a staff plug-in configuration.

To view this administrative console page, click **Resources** → **Staff Plug-in Provider** → ***staffpluginprovider_name*** → **Staff Plug-in Configuration** → ***staffpluginconfiguration_name***.

Scope:

The scope for this staff plug-in provider. The scope determines the level at which the resource definition is visible.

Type	**Value**
Data type	String
Valid values	Cell, node, or server.

Name:

The name of the staff plug-in configuration used for administrative purposes.

Type	**Value**
Data type	String

Description:

A description of the staff plug-in configuration.

Type	**Value**
Data type	String

JNDI Name:

The Java Naming and Directory Interface (JNDI) name used to look up the staff plug-in configuration in the namespace.

Type	**Value**
Data type	String

XSL Transform File:

The file name, including the absolute path, of the Extensible Style Language (XSL) transformation file. Default XSL transform files are provided for the sample plug-ins. If you have customized the transform file, specify the path to your file. The path name can include WebSphere environment variables.

Type	**Value**
Data type	String

About the staff service:

With Business Process Choreographer you can separate the logic of your business processes and human tasks from the staff resolution. Staff queries are resolved using a plug-in that is specific to the directory service. The basic aspects of using the staff service are described below:

- "Staff query and staff service concept" on page 182
- "Implementing a staff query" on page 182
- "Staff query verb set" on page 183
- "Repository-specific staff queries" on page 185
- "Staff verb XSL transformation files" on page 186
- "Using task and process context variables in staff verbs" on page 187
- "E-mail verb set" on page 187

For detailed information on the staff resolution plug-ins, refer to the *Process Choreographer: Staff Resolution Architecture*, the *Process Choreographer: Programming Model for Staff Resolution*, and the *Process Choreographer: Staff Resolution Parameter Reference* White papers in WebSphere Business Process Choreographer

Staff query and staff service concept

Use WebSphere Integration Developer to define staff queries for the staff support service. Staff queries are based on staff query templates, staff verbs, and are associated with the roles foreseen for human tasks and business processes, such as ProcessStarter and PotentialOwners.

A staff verb is identified by a unique name and includes a set of query parameters. The parameterized staff verb is transformed at application deployment time to determine a repository-specific staff query. This is used during execution of a business process or human task to retrieve the identities of users from a user repository.

Every business process or human task is associated with a specific staff plug-in configuration by its JNDI name. The configuration is extracted at deployment time from the process or task definition, and is used to map every staff verb found to a repository specific staff query. The mapping is governed by an XSL transformation file which takes a staff verb as input and produces the corresponding repository-specific query as output.

By default, three staff plug-in providers are included, representing different user repository options:

- The LDAP staff plug-in provider is used to generate staff queries which can be executed against an LDAP server.
- The user registry staff plug-in provider is used to generate staff queries which can be executed against the WebSphere Application Server user registry.
- The system staff plug-in provider is not associated with a user repository. Instead, it returns user identities that are derived directly from the staff verb parameters. This plug-in provider is intended for testing and prototyping purposes.

Each of the above staff plug-in providers is associated with at least one configuration. In particular, a configuration specifies an XSL transformation file that performs the mapping between staff verbs and staff queries that are specific to the repository. The following transformation files are provided by default:

- The LDAPTransformation.xsl file maps of staff verbs to LDAP-specific staff queries which can be executed via an JNDI interface.
- The UserRegistryTransformation.xsl file maps staff verbs to staff queries that are specific to the WebSphere user registry.
- The SystemTransformation.xsl file maps staff verbs to the actual user IDs specified in the verbs. It does not require a real user repository.
- The EverybodyTransformation.xsl file maps all staff verbs to the default result "everybody". It does not require a real user repository.

Implementing a staff query

The following example summarizes the steps involved in implementing a staff query:

1. Using WebSphere Integration Developer, a modeler associates a newly created task with the staff plug-in configuration bpe/staff/sampleldapconfiguration.
2. Using WebSphere Integration Developer, the modeler associates the roles for the task with corresponding staff verbs, for example, PotentialOwners is associated with the staff verb "Group Members" including the parameters :

- ″GroupName″ set to the value ″cn=group1,dc=mycomp,dc=com″
- ″IncludeSubgroups″ set to the value ″true″

3. In the context of the task, WebSphere Integration Developer stores the verb definition as an XML snippet:

```
<verb>
      <name>Group Members</name>
      <parameter id="GroupName">cn=group1,dc=mycomp,dc=com</parameter>
      <parameter id="IncludeSubgroups">true</parameter>
</verb>
```

4. When the task is deployed in WebSphere Application Server, the staff support service establishes that the LDAP staff plug-in provider bpe/staff/sampleldapconfiguration is to be used. The associated LDAPTransformation.xsl file is used to transform the staff verb into an LDAP-specific query, which is stored internally.

Staff query verb set

The staff support service accepts queries in an abstract form that is independent of the user repository infrastructure. Both the process editor and the task editor have a set of predefined staff verbs that can be used when you model processes and tasks. These verbs are contained in the `VerbSet.xml` file. This file is installed with WebSphere Integration Developer.

The individual staff resolution plug-ins and the XSLT mapping files do not support all of the verbs. The *Manager of Employee* verb, for example, is not available if you use the user registry or the system plug-in. You can modify the set of staff query verbs. Make your changes to a copy of the file. Ensure that the copied file has a different file name.

The following predefined set of verbs is available. For information on the parameters that can be used with each of the verbs, see Predefined staff verbs and their parameters.

Department Members
: Use this verb to define a query to retrieve the members of a department. The retrieved users belong to any of the specified departments (DepartmentName, AlternativeDepartmentName1, or AlternativeDepartmentName2). This verb is supported by the LDAP plug-in. You might need to customize the default mapping XSLT file to match the LDAP schema of your organization.

Everybody
: Use this verb to assign a work item to every user authenticated by the WebSphere Process Server. This verb is supported by the system, user registry, and LDAP plug-ins.

Group Members
: Use this verb to define a query to retrieve the members of up to three groups. The retrieved users belong to any of the specified groups (GroupName, AlternativeGroupName1, or AlternativeGroupName2). This verb is supported by the user registry and LDAP plug-ins. You might need to customize the default mapping XSLT file to match the LDAP schema of your organization.

Group Members without Named Users
: Use this verb to define a query to retrieve the members of a group except for explicitly named users of that group. One or more members can be specified for exclusion as a comma separated list. This verb is supported

by the user registry and LDAP plug-ins. You might need to customize the default mapping XSLT file to match the LDAP schema of your organization.

Group Members without Filtered Users
Use this verb to define a query to retrieve the members of a group except for a set of users defined by an LDAP search filter. This verb is supported by the LDAP plug-in. You might need to customize the default mapping XSLT file to match the LDAP schema of your organization.

Group Search
Use this verb to search for a group based on an attribute match and to retrieve the members of the group. This verb is supported by the user registry and LDAP plug-ins. You might need to customize the default mapping XSLT file to match the LDAP schema of your organization.

Manager of Employee
Use this verb to retrieve the manager of a person using the person's name. This verb is supported by the LDAP plug-in. You might need to customize the default mapping XSLT file to match the LDAP schema of your organization.

Manager of Employee by user ID
Use this verb to retrieve the manager of a person using the person's user ID. This verb is useful in combination with context queries. This verb is supported by the LDAP plug-in. You might need to customize the default mapping XSLT file to match the LDAP schema of your organization.

Native Query
Use this verb to define a native query based on directory-specific parameters. This verb is supported by the user registry and LDAP plug-ins. You might need to customize the default mapping XSLT file to match the LDAP schema of your organization.

Nobody
Use this verb to deny normal users access to the work item; For inline tasks, only the business process administrator and business process system administrator have access. For standalone tasks, only the human task administrator and human task system administrator have access. Depending on the API used, the authorized J2EE administrator will be different. For the business process API this will be the BPESystemAdministrator user, for the human task API this will be the TaskSystemAdministrator user. This verb is supported by the system, user registry, and LDAP plug-ins.

Person Search
Use this verb to search for a person based on an attribute match. This verb is supported by the user registry and LDAP plug-ins. You might need to customize the default mapping XSLT file to match the LDAP schema of your organization.

Role Members
Use this verb to retrieve the users associated with a staff repository role. The retrieved users belong to any of the specified roles (RoleName, AlternativeRoleName1, or AlternativeRoleName2). This verb is supported by the LDAP plug-in. You might need to customize the default mapping XSLT file to match the LDAP schema of your organization.

Users Use this verb to define a staff query for a user who is known by name. It is not recommended that you hard code user names in process templates.

This verb is useful for testing purposes. This verb is supported by the system, user registry, and LDAP plug-ins. You might need to customize the default mapping XSLT file to match the LDAP schema of your organization.

Users by user ID

Use this verb to define a staff query for a user whose user ID is known. Even though it is not recommended that you hard code user IDs in process and task templates, this verb is useful in combination with context queries, for example:

```
User [username='%wf:process.starter%']
```

This verb is useful for testing purposes. This verb is supported by the system, user registry, and LDAP plug-ins.

Repository-specific staff queries

The XSL transformation file that is associated with a staff plug-in configuration is used to generate staff queries that are specific to a particular repository. Each query can be executed by the respective staff plug-in to obtain a list of user IDs. The predefined queries which are available to a staff plug-in corresponds to the calls which can be executed by the plug-in and are therefore fixed.

Based on predefined queries, more complex queries can be formed using the following mechanisms:

- A union of query results implies that the user IDs returned by the individual queries will be added to the current result list of user identities. For example: The LDAP staff plug-in allows, among others, for predefined queries of the following types:

  The list of user IDs for the group members of a specified group:

  ```
  <sldap:usersOfGroup groupDN="cn=group1,dc=mycomp" recursive="yes">
  ...
  </sldap:usersOfGroup>
  ```

  The user ID of a specified user:

  ```
  <sldap:user dn="uid=user1,dc=mycomp" .../>
  ```

  A complex query can be constructed for the list of user IDs for the members of the specified group, plus the identity of the specified user:

  ```
  <sldap:staffQueries>
       <sldap:usersOfGroup groupDN="cn=group1,dc=mycomp" recursive="yes">
      ...
      </sldap:usersOfGroup>
      <sldap:user dn="uid=user1,dc=mycomp" .../>
  </sldap:staffQueries>
  ```

- A difference of query results implies that user IDs returned by a <remove> query will be removed from the current result list. For example, removing ″user1″ from the list of IDs retrieved for the specified group members:

  ```
  <sldap:staffQueries>
       <sldap:usersOfGroup groupDN="cn=group1,dc=mycomp" recursive="yes">
      ...
      </sldap:usersOfGroup>
      <sldap:remove value="user1"/>
  </sldap:staffQueries>
  ```

- Referencing query results implies that the results obtained from one query are used to influence the behavior of in a subsequent query. For example, in the following snippet, two queries are performed. First, the value of the ″manager″ attribute in the LDAP entry for the user ″uid=user1,...″ is retrieved and saved in

an intermediate variable ″supervisor″, which is then used to look up the manager's LDAP entry and retrieve the associated user identity.

```
<sldap:staffQueries>
       <sldap:intermediateResult name="supervisor">
         <sldap:user dn="uid=user1,dc=mycomp" attribute="manager" ... />
       </sldap:intermediateResult>
      <sldap:user dn="%supervisor% .../>
    </sldap:staffQueries>
```

Staff queries constructed according to above three combination rules can be executed by the staff plug-ins. For a detailed description of all predefined staff queries for each of the supported staff plug-ins and more examples of combining them consult the *Process Choreographer: Staff Resolution Parameter Reference* White papers in WebSphere Business Process Choreographer.

Staff verb XSL transformation files

The XSL transformation file specified for a staff plug-in configuration defines the mapping between staff verbs and repository-specific staff queries. Every staff plug-in configuration is expected to have its own XSL transformation file.

The default transformation files are:

- LDAPTransformation.xsl for the LDAP staff provider plug-in
- UserRegistryTransformation.xsl for the user registry staff provider plug-in
- SystemTransformation.xsl and EverybodyTransformation.xsl for the system staff provider plug-in

These transformation files map the predefined set of staff verbs to corresponding simple and composite repository-specific queries. These files are located in the *install_root*/ProcessChoreographer/Staff directory.

The transformation files assume certain semantics for staff verbs and their execution using generated repository-specific staff queries. If other semantics are required, the mapping in the transformation file must be changed accordingly.

For example, the LDAP staff plug-in comes with a predefined staff verb:

```
<staff:verb>
  <staff:name>Manager of Employee</staff:name>
   <staff:parameter id="EmployeeName">
          uid=anEmployeeName,cn=users,dc=ibm,dc=com
   </staff:parameter>
</staff:verb>
```

This is mapped by the LDAPTransformation.xsl file to an LDAP query:

```
<sldap:staffQueries>
     <sldap:intermediateResult name="supervisor">
     <sldap:user dn="anEmployeeName" attribute="manager"
                              objectclass="inetOrgPerson"/>
     </sldap:intermediateResult>
     <sldap:user dn="%supervisor%" attribute="uid" objectclass="inetOrgPerson"/>
</sldap:staffQueries>
```

Which explicitly assumes that the LDAP DN of the supervisor is stored in the employee's attribute ″manager″. If that the verb is to have different semantics, for example, if the supervisor should come from the LDAP attribute ″teacher″. Then the LDAP specific query must be changed accordingly:

```
<sldap:staffQueries>
     <sldap:intermediateResult name="supervisor">
     <sldap:user dn="anEmployeeName" attribute="teacher"
```

```
                        objectclass="inetOrgPerson"/>
    </sldap:intermediateResult>
    <sldap:user dn="%supervisor%" attribute="uid" objectclass="inetOrgPerson"/>
</sldap:staffQueries>
```

The means to achieve this is to adapt the LDAPTransformation.xsl file accordingly:

```
  <xsl:template name="ManagerOfEmployee">
      <sldap:staffQueries>...
      <sldap:intermediateResult>
           <xsl:attribute name="name">supervisor</xsl:attribute>
           <sldap:user>
                 <xsl:attribute name="dn">
                     <xsl:value-of select="staff:parameter[@id='EmployeeName']"/>
                 </xsl:attribute>
                 <xsl:attribute name="attribute">teacher</xsl:attribute>
                 ...
           </sldap:user>
      </sldap:intermediateResult>
      <sldap:user>
            <xsl:attribute name="dn">%supervisor%</xsl:attribute>
            ...
      </sldap:user>
      </sldap:staffQueries>
</xsl:template>
```

You can get an deeper understanding of the mapping behavior by viewing the default transformation files. The semantics of the default transformations are described in “Staff query verb set” on page 183.

Using task and process context variables in staff verbs

In certain staff verbs, you can use business process and human task context variables as parameter values. This enables the staff support service to resolve staff verbs at run time, based on information supplied by the contexts. For example, the staff verb:

```
<verb>
<name>Users by user ID</staff:name>
 <parameter id="UserID">%htm:input.\name%</staff:parameter>
</verb>
```

specifies as a parameter, the task context variable htm:input.\name, which denotes the ″name″ part of the input message received by the task when it is initiated. The staff support service dynamically replaces the context variable with the actual task context value.

For a description of the verbs and the parameters in which you can use context variables, see “Predefined staff verbs and their parameters” on page 188.

E-mail verb set

The e-mail verb set in WebSphere Integration Developer is for e-mail notifications for task escalations. These e-mail verbs are transformed during modeling and deployment into a set of queries that can be run on a staff repository. E-mail verbs are defined for the most common staff verbs supported by the LDAP plug-in. The following e-mail verbs are available:

- Email Address for Department Members
- Email Address for Group Members
- Email Address for Group Members without Names Users
- Email Address for Group Members without Filtered Users

- Email Address for Group Search
- Email Address for Role Members
- Email Address for Users
- Email Address for Users by User ID

For the other LDAP staff verbs, the user identifiers retrieved by the staff verbs are used as input to the Email Address for Users by User ID verb.

Before the e-mail verbs can be run as queries on a specific staff repository, they must be translated into executable queries using the LDAP XSL transformation. The result of a transformation (mapping) can be run by the LDAP staff resolution plug-in. At run time, the query returns a set of e-mail addresses, for example, user1@mycomp.com, user2@mycomp.com, and so on.

Predefined staff verbs and their parameters: You can use staff verbs in the WebSphere Integration Developer to model staff assignments in a business process or human task. These staff verbs are transformed during modeling and deployment into a set of queries that can be run on a staff repository. The parameters for the following predefined staff verbs are listed here:

- Department Members
- Everybody
- Group Members
- Group Members without Named Users
- Group Members without Filtered Users
- Group Search
- Manager of Employee
- Manager of Employee by user ID
- Native Query
- Nobody
- Person Search
- Role Members
- Users
- Users by user ID

Department Members

Use this verb to define a query to retrieve the members of a department.

Parameter	Use	Type	Supported by	Description
DepartmentName	Mandatory	string	LDAP	Department name of the users to retrieve.
IncludeNestedDepartments	Mandatory	boolean	LDAP	Specifies whether nested departments are considered in the query.
Domain	Optional	string	None	The domain to which the department belongs. Use this parameter to limit the query to a subset of the directory.

Parameter	Use	Type	Supported by	Description
AlternativeDepartmentName1	Optional	string	LDAP	An additional department to which the users can belong.
AlternativeDepartmentName2	Optional	string	LDAP	An additional department to which the users can belong.

Everybody

Use this verb to assign a work item to every user authenticated by WebSphere Process Server. This verb has no parameters; it is supported by the system, user registry, and LDAP plug-ins.

Group Members

Use this verb to define a query to retrieve the members of a group.

Parameter	Use	Type	Supported by	Description
GroupName	Mandatory	string	User registry, LDAP	Group name of the users to retrieve.
IncludeSubgroups	Mandatory	boolean	LDAP	Specifies whether nested subgroups are considered in the query.
Domain	Optional	string	None	The domain to which the group belongs. Use this parameter to limit the query to a subset of the directory.
AlternativeGroupName1	Optional	string	User registry, LDAP	An additional group to which the users can belong.
AlternativeGroupName2	Optional	string	User registry, LDAP	An additional group to which the users can belong.

Group Members without Named Users

Use this verb to define a query to retrieve all of the members of a group except for the explicitly named users.

Parameter	Use	Type	Supported by	Description
GroupName	Mandatory	string	User registry, LDAP	Group name of the users to retrieve. Supports custom properties that are evaluated at run time.
IncludeSubgroups	Mandatory	boolean	LDAP	Specifies whether nested subgroups are considered in the query.

Parameter	Use	Type	Supported by	Description
NamedUsers	Mandatory	string	User registry, LDAP	The user IDs of the users to exclude from the retrieved group members list. Supports context variables and custom properties, such as `%htm:task.originator%`

Group Members without Filtered Users

Use this verb to define a query to retrieve the all of the members of a group except for a set of users that is defined by an LDAP search filter.

Parameter	Use	Type	Supported by	Description
GroupName	Mandatory	string	LDAP	Group name of the users to retrieve.
IncludeSubgroups	Mandatory	boolean	LDAP	Specifies whether nested subgroups are considered in the query.
FilterAttribute	Mandatory	string	LDAP	Name of the attribute to use in the LDAP filter.
FilterValue	Mandatory	string	LDAP	Filter value to use in the LDAP filter.

Group Search

Use this verb to search for a group based on an attribute match and to retrieve the members of the group. You must set one attribute. If you set more than one attribute, only the first attribute is evaluated.

Parameter	Use	Type	Supported by	Description
GroupID	Optional	string	User registry, LDAP	The group ID of the users to retrieve.
Type	Optional	string	LDAP	The group type of the users to retrieve.
IndustryType	Optional	string	LDAP	The industry type of the group to which the users belong.
BusinessType	Optional	string	LDAP	The business type of the group to which the users belong.
GeographicLocation	Optional	string	LDAP	An indication of where the users are located.
Affiliates	Optional	string	LDAP	The affiliates of the users.
DisplayName	Optional	string	LDAP	The display name of the group.
Secretary	Optional	string	LDAP	The secretary of the users.

Parameter	Use	Type	Supported by	Description
Assistant	Optional	string	LDAP	The assistant of the users.
Manager	Optional	string	LDAP	The manager of the users.
BusinessCategory	Optional	string	LDAP	The business category of the group to which the users belong.
ParentCompany	Optional	string	LDAP	The parent company of the users.

Manager of Employee

Use this verb to retrieve the manager of a person using the person's name.

Parameter	Use	Type	Supported by	Description
EmployeeName	Mandatory	string	LDAP	The name of the employee whose manager is retrieved.
Domain	Optional	string	None	The domain to which the employee belongs. Use this parameter to limit the query to a subset of the directory.

Manager of Employee by user ID

Use this verb to retrieve the manager of a person using the person's user ID.

Parameter	Use	Type	Supported by	Description
EmployeeUserID	Mandatory	string	LDAP	The user ID of the employee whose manager is retrieved. Supports context variables and custom properties, such as `%wf:process.starter%`
Domain	Optional	string	None	The domain to which the employee belongs. Use this parameter to limit the query to a subset of the directory.

Native Query

Use this verb to define a native query based on directory-specific parameters.

Parameter	Use	Type	Supported by	Description
QueryTemplate	Mandatory	string	User registry, LDAP	The query template to use for the query. The default mapping files for the user registry and LDAP plug-ins support the templates search, user, and usersOfGroup.
Query	Mandatory	string	User registry, LDAP	Specifies the query. You can use context variables and custom properties, such as `%wf:process.starter%`. The type of query depends on the plug-in and the query template. **User registry** • search template: search pattern • user template: user name • usersOfGroup: group name **LDAP** • search template: search filter • user template: user dn • usersOfGroup: group dn
AdditionalParameter1	Optional	string	User registry, LDAP	Specifies the query. You can use context variables, such as `%wf:process.starter%`. The type of parameter depends on the plug-in and the query template. **User registry** • search template. Used for the search type. Supported values: `group` and `user`. • user template. Not supported • usersOfGroup. Not supported **LDAP** • search template. Used to specify whether recursive search is done. Supported values: `yes` and `no` • user template. Not supported • usersOfGroup. Used to specify whether recursive search is done. Supported values: `yes` and `no`
AdditionalParameter2	Optional	string	User registry, LDAP	Use this verb to specify an additional parameter.

Parameter	Use	Type	Supported by	Description
AdditionalParameter3	Optional	string	User registry, LDAP	Use this verb to specify an additional parameter. If you use the default mapping XSLT files, this parameter is not supported.
AdditionalParameter4	Optional	string	User registry, LDAP	Use this verb to specify an additional parameter. If you use the default mapping XSLT files, this parameter is not supported.
AdditionalParameter5	Optional	string	User registry, LDAP	Use this verb to specify an additional parameter. If you use the default mapping XSLT files, this parameter is not supported.

Nobody

For inline tasks, only the business process administrators have access. For standalone tasks, only the human task administrators have access. In addition, when using the business flow manager API the BPESystemAdministrator role members have access, for the human task manager API the TaskSystemAdministrator role members have access. This verb has no parameters. It is supported by the system, user registry, and LDAP plug-ins.

Person Search

Use this verb to search for people based on an attribute match. You must set one attribute. If you set more than one attribute, only the first attribute is evaluated.

Parameter	Use	Type	Supported by	Description
UserID	Optional	string	User registry, LDAP	The user ID of the users to retrieve.
Profile	Optional	string	LDAP	The profile of the users to retrieve.
LastName	Optional	string	LDAP	The last name of the users to retrieve.
FirstName	Optional	string	LDAP	The first name of the users to retrieve.
MiddleName	Optional	string	LDAP	The middle name of the users to retrieve.
Email	Optional	string	LDAP	The e-mail address of the users.
Company	Optional	string	LDAP	The company to which the users belong.
DisplayName	Optional	string	LDAP	The display name of the users.
Secretary	Optional	string	LDAP	The secretary of the users.

Parameter	Use	Type	Supported by	Description
Assistant	Optional	string	LDAP	The assistant of the users.
Manager	Optional	string	LDAP	The manager of the users.
Department	Optional	string	LDAP	The department to which the users belong.
Phone	Optional	string	LDAP	The telephone number of the users.
Fax	Optional	string	LDAP	The fax number of the users.
Gender	Optional	string	LDAP	Whether the user is male or female.
Timezone	Optional	string	LDAP	The time zone in which the users are located.
PreferredLanguage	Optional	string	LDAP	The preferred language of the user.

Role Members

Use this verb to retrieve the users associated with a business process role.

Parameter	Use	Type	Supported by	Description
RoleName	Mandatory	string	LDAP	Role name of the users to retrieve.
IncludeNestedRoles	Mandatory	boolean	LDAP	Specifies whether nested roles are considered in the query.
Domain	Optional	string	None	The domain to which the role belongs. Use this parameter to limit the query to a subset of the directory.
AlternativeRoleName1	Optional	string	LDAP	An additional role name for the user.
AlternativeRoleName2	Optional	string	LDAP	An additional role name for the user.

Users

Use this verb to define a staff query for a user who is known by name.

Parameter	Use	Type	Supported by	Description
Name	Mandatory	string	System, user registry, LDAP	The name of the user to retrieve.
AlternativeName1	Optional	string	System, user registry, LDAP	An additional user name. Use this parameter to retrieve more than one user.
AlternativeName2	Optional	string	System, user registry, LDAP	An additional user name. Use this parameter to retrieve more than one user.

Users by user ID

Use this verb to define a staff query for a user whose user ID is known. Use short names to specify values, for example, wpsadmin. This verb does not imply access to a staff repository.

Parameter	Use	Type	Supported by	Description
UserID	Mandatory	string	System, user registry, LDAP	The user ID of the user to retrieve.
AlternativeID1	Optional	string	System, user registry, LDAP	An additional user ID. Use this parameter to retrieve more than one user.
AlternativeID2	Optional	string	System, user registry, LDAP	An additional user ID. Use this parameter to retrieve more than one user.

Implementing new custom verbs:

This describes how to add new staff verbs the staff support service infrastructure so that they can be used in WebSphere Integration Developer when modeling business processes and human tasks.

You must add the new staff verb specification to the VerbSet.xml file which is part of your WebSphere Integration Developer installation. For example, for a new verb Mentor of Employee:

```
  <vs:DefineVerb name='Mentor of Employee'>
    <vs:Description>Assigns the mentor of an employee.
Supported by sample XSLT files for:
 - LDAP
    </vs:Description>
    <vs:Mandatory>
      <vs:Parameter>
        <vs:Name>EmployeeName</vs:Name>
        <vs:Type>xsd:string</vs:Type>
      </vs:Parameter>
    </vs:Mandatory>
    <vs:Optional>
      <vs:Parameter>
        <vs:Name>Domain</vs:Name>
        <vs:Type>xsd:string</vs:Type>
      </vs:Parameter>
    </vs:Optional>
  </vs:DefineVerb>
```

You must add the new verb to the dispatcher section of the LDAP transformation file. For example:

```
    <xsl:choose>
      ...
      <xsl:when test="$verb='Mentor of Employee'">
        <xsl:call-template name="MentorOfEmployee"/>
      ...
   </xsl:choose>
```

You must also add to the LDAP transformation file a template that implements the mapping. For example::

```
<!-- Begin template MentorOfEmployee -->
  <xsl:template name="MentorOfEmployee">
    <sldap:staffQueries>
      <xsl:attribute name="threshold">
        <xsl:value-of select="$Threshold"/>
      </xsl:attribute>

      <sldap:intermediateResult>
        <xsl:attribute name="name">mentorvariable</xsl:attribute>
        <sldap:user>
          <xsl:attribute name="dn">
            <xsl:value-of select="staff:parameter[@id='EmployeeName']"/>
          </xsl:attribute>
          <xsl:attribute name="attribute">mentor</xsl:attribute>
          <xsl:attribute name="objectclass">inetOrgPerson</xsl:attribute>
        </sldap:user>
      </sldap:intermediateResult>

      <sldap:user>
        <xsl:attribute name="dn">%mentorvariable%</xsl:attribute>
        <xsl:attribute name="attribute">uid</xsl:attribute>
        <xsl:attribute name="objectclass">inetOrgPerson</xsl:attribute>
      </sldap:user>
    </sldap:staffQueries>
  </xsl:template>
<!-- End template MentorOfEmployee -->
```

Verify that the mapping generates a valid LDAP specific query.

Adapting the LDAP transformation file:

Describes how to adapt the LDAP transformation XSL file to suit your LDAP schema.

The default LDAPTransformation.xsl file maps predefined staff verbs to LDAP queries, which make use of elements of the default LDAP schema assumed by WebSphere. This schema assumes the following:

- The LDAP object class for group entries is groupOfName.
- The group entry attribute containing the member DNs for the group is member.
- The LDAP object class for person entries is inetOrgPerson.
- The attribute containing the login ID in a person entry is uid.
- The person entry attribute containing the e-mail address of a person is mail.
- The person entry attribute containing the distinguished name of the manager of a person is manager.

If your LDAP schema features different object class and attribute names, you must change these settings in the LDAP transformation files that you use. For the default LDAPTransformation.xsl file, changing the setting can be done in the variable declaration part of the file:

```
  <xsl:variable name="DefaultGroupClass">groupOfNames</xsl:variable>
  <xsl:variable name="DefaultGroupClassMemberAttribute">member</xsl:variable>

  <xsl:variable name="DefaultPersonClass">inetOrgPerson</xsl:variable>
  <xsl:variable name="DefaultUserIDAttribute">uid</xsl:variable>
  <xsl:variable name="DefaultMailAttribute">mail</xsl:variable>
  <xsl:variable name="DefaultManagerAttribute">manager</xsl:variable>
```

You can apply changes within the XSL templates that transform the individual staff verbs, as illustrated in the following examples.

Example: DepartmentMembers

Changing the object class for person entries to ePerson and the login ID attribute to cn:

```
<sldap:StaffQueries>
  <xsl:attribute name="threshold">
    <xsl:value-of select="$Threshold">
  </xsl:attribute>

  <sldap:search>
...
    <sldap:attribute>
      <xsl:attribute name="name">cn</xsl:attribute>
      <xsl:attribute name="objectclass">ePerson</xsl:attribute>
      <xsl:attribute name="usage">simple</xsl:attribute>
    </sldap:attribute>

  </sldap:search>
</sldap:StaffQueries>
```

Example: GroupMembers

Changing the object class for group entries to groupOfUniqueNames, the group entry attribute containing the member DN list to uniqueMember, and the person entry attribute containing the login in to cn:

```
<sldap:usersOfGroup>
...

  <sldap:attribute>
    <xsl:attribute name="name">uniqueMember</xsl:attribute>
    <xsl:attribute name="objectclass">groupOfUniqueNames</xsl:attribute>
    <xsl:attribute name="usage">recursive</xsl:attribute>
  </sldap:attribute>

  ...
  <sldap:attribute>
    <xsl:attribute name="name">cn</xsl:attribute>
    <xsl:attribute name="objectclass">inetOrgPerson</xsl:attribute>
    <xsl:attribute name="usage">simple</xsl:attribute>
  </sldap:attribute>

</sldap:usersOfGroup>
```

Example: GroupMembersWithoutFilteredUsers

Changing the LDAP filter operator to >=.

```
<sldap:StaffQueries>
  <sldap:usersOfGroup>
    ...
  </sldap:usersOfGroup>

  <sldap:intermediateResult>
    <xsl:attribute name="name">filteredusers</xsl:attribute>
    <sldap:search>
      <xsl:attribute name="filter">
        <xsl:value-of select="staff:parameter[@id='FilterAttribute']"/>
         >=
        <xsl:value-of select="staff:parameter[@id='FilterValue']"/>
      </xsl:attribute>
     ...
    <sldap:search>
     ...
```

```
  </sldap:intermediateResult>
   ...
</sldap:StaffQueries>
```

Example: GroupSearch

Changing the search attribute to MyType, the object class to mypersonclass, and the attribute containing the login ID to myuid.

```
<sldap:StaffQueries>
 ...
  <sldap:search>
    <xsl:attribute name="filter">
      (&amp;
       ...
      <xsl:if test="staff:parameter[@id='MyType']!="">
       (<xsl:value-of select="$GS_Type"/>=
        <xsl:value-of select=staff:parameter[@id='Type']"/>)
      </xsl:if>
      )
       ...
    </xsl:attribute>

    <sldap:attribute>
      <xsl:attribute name="name">myuid</xsl:attribute>
      <xsl:attribute name="objectclass">mypersonclass</xsl:attribute>
      <xsl:attribute name="usage">simple</xsl:attribute>
    </sldap:attribute>
     ...
  <sldap:search>
</sldap:StaffQueries>
```

Example: Manager of Employee

Changing the attribute containing the manager DN to managerentry and the source of the manager login ID attribute to name.

```
<sldap:StaffQueries>

  <sldap:intermediateResult>
   ...
    <sldap:user>
     ...
      <xsl:attribute name="name">managerentry</xsl:attribute>
       ...
    </sldap:user>
  </sldap:intermediateResult>

  <sldap:user>
   ...
    <xsl:attribute name="name">name</xsl:attribute>
     ...
  </sldap:user>
</sldap:StaffQueries>
```

Example: PersonSearch

Changing the search attribute to MyAttribute, the object class to mypersonclass, and the source of the return attribute to myuid.

```
<sldap:StaffQueries>
 ...
  <sldap:search>
    <xsl:attribute name="filter">
      (&amp;
```

```
      ...
      <xsl:if test="staff:parameter[@id='MyAttribute']!="">
       (<xsl:value-of select="$PS_UserID"/>=
        <xsl:value-of select=staff:parameter[@id='UserID']"/>)
      )
      </xsl:if>
       ...
    </xsl:attribute>

    <sldap:attribute>
      <xsl:attribute name="name">myuid</xsl:attribute>
      <xsl:attribute name="objectclass">mypersonclass</xsl:attribute>
      <xsl:attribute name="usage">simple</xsl:attribute>
    </sldap:attribute>
     ...
  </sldap:search>
</sldap:StaffQueries>
```

Example: Users

Changing the source of the return attribute to myuid and the object class to mypersonclass.

```
<sldap:user>
 ...
  <xsl:attribute name="attribute">myuid</xsl:attribute>
  <xsl:attribute name="objectclass">mypersonclass</xsl:attribute>
</sldap:user>
```

Troubleshooting the staff service and the staff plug-ins:

This describes how to troubleshoot problems with the staff service and the staff plug-ins.

One of the following situations might be caused by a problem with the staff service or a staff plug-in:

- Stopped staff activities
- Changes to the staff repository that are not immediately reflected in work-item assignments
- Error and warning messages when accessing an LDAP server

Use this overview task to help resolve the problem. You can also go to the Technical support search page, to look for additional information.

Stopped staff activities

You encountered one or more of the following problems:

- Work items resulting from staff activities cannot be claimed although the business process started navigating successfully.
- The SystemOut.log file contains the following message: `CWWB0057I: Activity 'MyStaffActivity' of processes 'MyProcess' has been stopped because of an unhandled failure...`

This message indicates that WebSphere Application Server security might not be enabled. Human tasks and processes that use people authorization require that security is enabled and the user registry is configured. Take the following steps:

1. Check that WebSphere security is enabled. In the administrative console, go to **Security** → **Global Security** and make sure the check box for **Enable global security** is selected.

2. Check that the user registry is configured. In the administrative console, go to **Security** → **User Registries** and check the **Active user registry** attribute.

Changes to the staff repository that are not immediately reflected in work-item assignments

To optimize the staff query resolution performance, the retrieved query results are cached. The cache content is checked for currency when a new process instance is created or the corresponding staff activity gets scheduled. By default, the time after which the shared staff query results expire is one hour.To change how often the cache is refreshed, see *Setting the interval for refreshing staff queries* in the PDF for *Administering WebSphere Process Server*.

Error and warning messages when accessing an LDAP server

When tracing is enabled for the staff support service, that is when the trace string includes: `com.ibm.bpe.*=all: com.ibm.ws.staffsupport.ws.*=all`, the following typical erroneous cases are indicated by warning or error messages. The following are examples:

- `Could not connect to LDAP server` in the trace.log file indicates that the LDAP server cannot be connected to.
- `javax.xml.transform.TransformerException: org.xml.sax.SAXParseException: Element type "xsl:template" must be followed by either attribute specifications, ">" or "/>"` in the System.out or System.err files indicates that the LDAPTransformation.xsl file cannot be read.
- `LDAP object not found. dn: uid=unknown,cn=users,dc=ibm,dc=com [LDAP: error code 32 - No Such Object]` in the trace.log file indicates that an LDAP entry cannot be found.
- `Requested attribute "quid" not found in: uid=test222,cn=users,dc=ibm,dc=com` in the trace.log file indicates that an attribute cannot be found in an LDAP entry. .

Installing and starting Business Process Choreographer Explorer:

Business Process Choreographer Explorer provides a user interface for administering processes and handling tasks. It is a Java 2 Enterprise Edition (J2EE) Web application, based on the JavaServer Faces (JSF) technology and the Business Process Choreographer Explorer components.

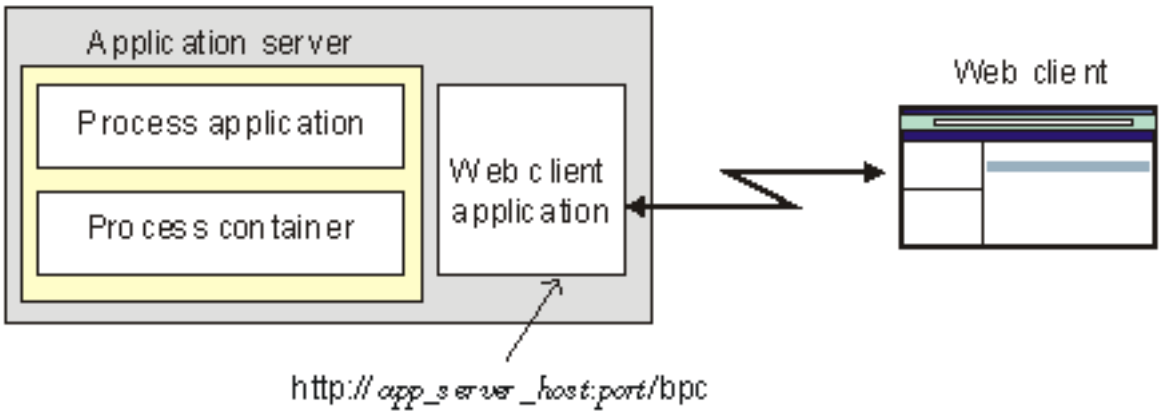
- "About Business Process Choreographer Explorer"
- "Configuring Business Process Choreographer Explorer" on page 201
- "Starting to use the Business Process Choreographer Explorer" on page 201

About Business Process Choreographer Explorer:

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

You can install Business Process Choreographer Explorer on the application server, or on the cluster, where you installed both a business process container and a human task container. If you want to work with business process applications or human task applications on several application servers or clusters, you must install

Business Process Choreographer Explorer on each of these application servers or clusters.



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

Configuring Business Process Choreographer Explorer:

Use this task to configure Business Process Choreographer Explorer using a script.

You have configured the business process container and human task container.

You have not yet installed Business Process Choreographer Explorer, or you want to add it to an existing Business Process Choreographer configuration.

1. Change to the Business Process Choreographer directory and invoke the `clientconfig.jacl` script.
2. The `clientconfig.jacl` script prompts you for any required information that was not provided as parameters.
3. **Optional:** If you have problems with the configuration, check the log file written by the `clientconfig.jacl` script. This log is located in the `profiles/`*`profileName`*`/logs/clientconfig.log` file. This directory also contains a `wsadmin.traceout` file that might contain more information about the problem.

Business Process Choreographer Explorer is configured and ready to use.

Start Business Process Choreographer Explorer.

Starting to use the Business Process Choreographer Explorer:

Before you can start using the Business Process Choreographer Explorer from a Web browser, you must have installed the business process container, human task container, and Business Process Choreographer Explorer applications, and they must be running.

To start using the Business Process Choreographer Explorer:

1. Open the following Web page in a Web browser:

   `http://app_server_host:port_no/bpc`

   Where:

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

   port_no
   The port number used by Business Process Choreographer Explorer. The port number depends on your system configuration.
2. If security is enabled, you must enter a user ID and password, then click **OK**.

The initial page of the Business Process Choreographer Explorer is displayed, which shows the work items in your To Do list. You can work with business processes on different application servers at the same time.

Changing the appearance of the default Web application:

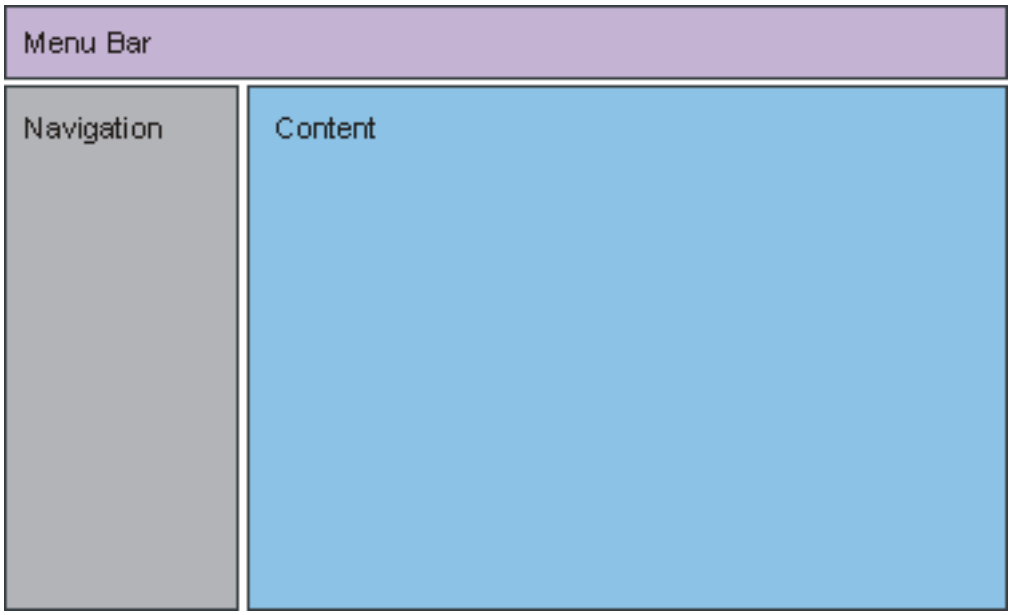
Business Process Choreographer Explorer provides a ready-to-use Web user interface based on JavaServer Pages (JSP) files and JavaServer Faces (JSF) files. You can adapt the user interface to fit a certain look and feel without writing any new code.

The user interface consists of a header, a navigation pane, and a content pane. A cascading style sheet (CSS) controls how the Web interface is rendered.

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

1. **Optional:** Modify the header. The `Menubar.jsp` file is always displayed in the user interface. The default `Menubar.jsp` file contains logos, images, and a link to the information center.
2. **Optional:** Modify the style sheet. The default style sheet, `style.css`, contains styles for the elements in the header, the navigation pane, and the content pane.

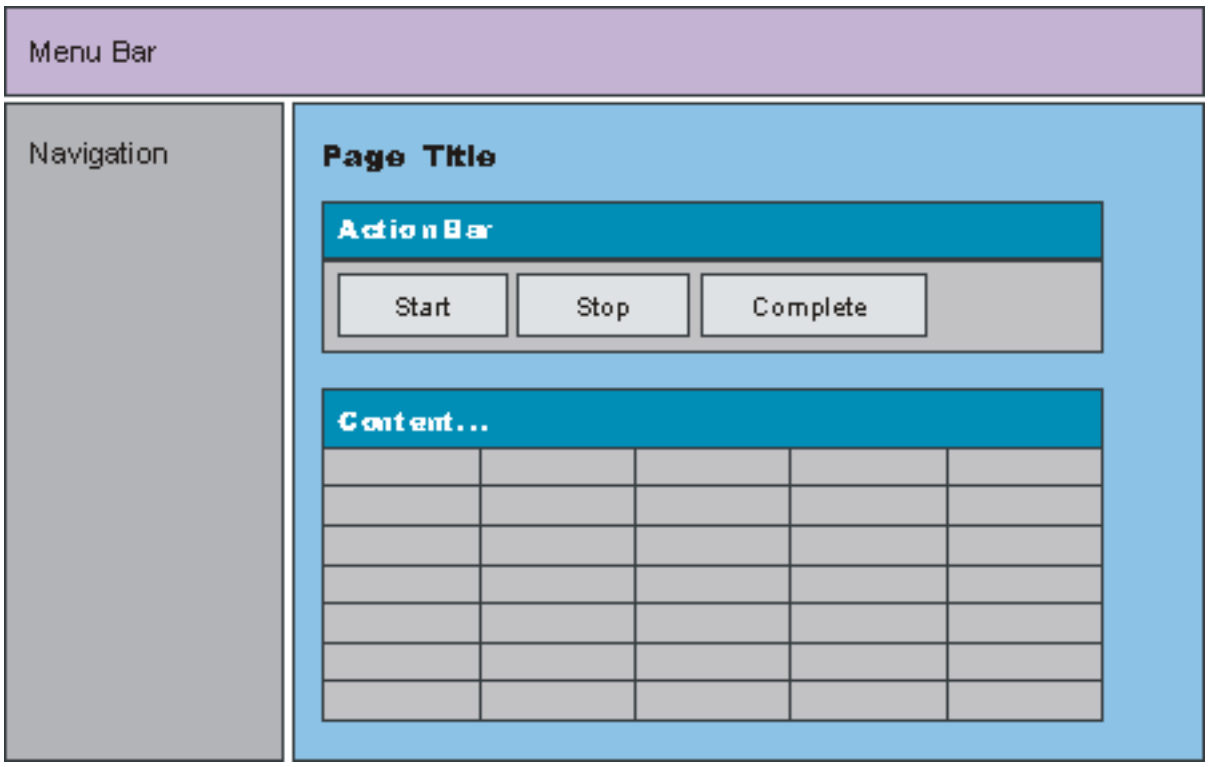
Layout of the default Web user interface: The default Web user interface for Business Process Choreographer Explorer consists of a menu bar, a navigation pane, and a content pane.



The menu bar and the navigation panes are always displayed. They are generated by the `Menubar.jsp` and `Navigation.jsp` files. Other JavaServer Pages (JSP) files use the `<jsp:include page=xxx>` tag to reference these files. The information shown in the content pane depends on the JSP file that is used to generate the page.

All of the JSP files used in the default user interface are in the *install_root*/pages/layouts/ directory.

The layout of the user interface is implemented in HTML forms; tables are used for tabular data only. The visual appearance of each of the sections is controlled by a style definition in the `style.css` file.



The main table has one row and one column and includes the `Navigation.jsp` pane. It also provides a table cell for the page content. Depending on the content, this cell can also contain tables, forms, and labels. The HTML template for the page layout looks similar to the following example:

```
<body>
 <jsp:include page="Header.jsp" flush="true"/>
 <table class="page">
  <tr>
   <jsp:include page="Navigation.jsp" flush="true"/>
   <td class="content">
     ...
    </td>
  </tr>
 </table>
</body>
```

Styles used in the Business Process Choreographer Explorer interface:

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

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

- "Body of a page" on page 205
- "Login page" on page 205
- "Menu bar" on page 205
- "Navigator" on page 205
- "Content panels" on page 206
- "Command bar" on page 206
- "Lists" on page 206

Body of a page

Style name	Description
.pageBody	The main content area in a table layout with two columns: navigator and content.
.pageBody td	The individual cell in the overall page body layout.
.pageBodyNavigator	The column that contains the navigator.
.pageBodyContent	The column that contains the content.

Login page

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

Menu bar

Style name	Description
.menubar	The JSF subview.
.menuContainer	The container panel including the menu items, for example, labels, and links.
.menuItem	An item on the menu bar.
.menuitem a	A menu item that is a link.
.menuitem a:visited	A menu item that is a link that the user has visited.
.menuitem a:hover	Hovering over a menu item that is a link.

Navigator

Style name	Description
.navigator	JSF subview for navigator which contains the links to the lists.
.navigatorTitle	The title for each navigator box.

Style name	Description
.navigatorFrame	The division for each navigator box, for example, to draw a border.
.taskNavigatorTitle	A class of titles for navigation boxes. They are used to distinguish between links to lists of business process objects and human task objects.
.navigatorItem	An item in the navigator box.
.navigatorItemList	An item that represents a list.
.expanded / .expanded div / .expanded a .expanded a:visited	Used when the navigator boxes are expanded.
.collapsed	Used when the navigator boxes are collapsed.

Content panels

Style name	Description
.panelContainer	The division panel that contains the list, details or messages. This element is embedded in the pageBodyContent column.
.panelTitle	The title for the displayed content, for example, My Tasks.
.panelHelp	The division container that contains the help text and the icon.
.panelGroup	The division container that contains the command bar and list, details or message.

Command bar

Style name	Description
.commandbarHeader	The title above the command bar.
.commandbar	The division container around the command-bar area.

Lists

Style name	Description
.listHeader	The style used in the header row of the list.
.list	The table that contains the rows.
.list tbody td, .list th	Styles for the header row.
.list thead input, .list tbody input	Check boxes for lists.
.list tfoot td	Last row entry is the footer information.
.list tfoot div	The division container around the footer elements.
.list tfoot input	The input controls in the footer.
.list a / .list a:visited	For links rendered in the list.

Details panel

Style name	Description
.details	The division container around a details panel.
div.details	Details styles that are embedded in the division container.
table.details	Details styles that are embedded in the table container.
td.detailsProperty	The label for a property name.
td.detailsValue	The text for a property value.

Message data

Style name	Description
.messageData	The division container around a message.
.messageData table	The table container in which the message is placed.
.messageDataButton	Button style for Add and Remove buttons in the message form.
.messageData td / .messageData th	Body and header cells.
.messageDataOutput	For rendering read-only text.
.messageDataValidInput	For message values that are valid.
.messageDataInvalidInput	For message values that are not valid.

Tabbed panes

Style name	Description
.tabbedPane	The division container around all of the tabbed panes.
.tabHeader	The tab header of a tabbed pane.
.tabHeader ul	Each header is organized in an unordered list.
.tabHeader li	Each header label is a list item.
.tabHeader a / .tabHeader a:hover / .tabHeader a.tab	The header label as a link.
.tabHeader a.selectedTab	The selected tab header.
.tabPane	The division container that encloses a tabbed pane.
.tabPane table	A pane is always embedded in a panel grid. This action results in a table container around the pane.
.tabPane .list th, .tabPane .list tfoot div	Settings for lists on a tabbed pane.

Search pages

Style name	Description
.searchPanel	The tabbed pane for a search panel. See also tabbed panes.
.searchPanelFilter	The table container for a search form.
.searchLabel	The label for a search form control.

Style name	Description
.searchListBox	The list box control for select options.

Error details

Style name	Description
.errorPage	The tabbed pane for an error page.
.errorLink / .errorLink a / .errorLink a:visited	Styles uses to render the button links on a page.
.errorDetails	Tabbed pane with error details.
.errorDetailsStack	Tabbed pane with an exception stack.
.errorDetailsStack table / .errorDetailsStack td	The exception stack that is shown as rows in a table.
.errorDetailsMessage	Text style for error message.

Sorting

Style name	Description
.ascending	Style for the list header class when the list is sorted by this column in ascending order.
.descending	Style for the list header class when the list is sorted by this column in descending order.
.unsorted	Style for the list header class when the list is not sorted by this column.

Customizing input and output forms:

Business Process Choreographer Explorer is a Web client for working on business processes and human tasks. The Web client provides default input and output forms for displaying and entering business data. You can use JSP documents to customize these default input and output forms.

To include user-defined JavaServer Pages (JSP) documents in the Web client, you must specify them when you model a human task in WebSphere Integration Developer. For example, you can provide JSP documents for a specific task and its input and output messages, and for a specific user role or all user roles. At run time, the user-defined JSP documents are included in the user interface to display output data and collect input data.

The customized forms are not self-contained Web pages; they are HTML fragments that Business Process Choreographer Explorer imbeds in an HTML form, for example, fragments for labels and input fields.

When a button is clicked on the page that contains the customized forms, the input is submitted and validated in Business Process Choreographer Explorer. The validation is based on the type of the properties provided and the locale used in the browser. If the input cannot be validated, the same page is shown again and information about the validation errors is provided in the messageValidationErrors request attribute.

To add customized forms to Business Process Choreographer Explorer, complete the following using WebSphere Integration Developer.

1. Create the customized forms.

   The user-defined JSP documents for the input and output forms used in the Web interface access message data. Use Java snippets or the JSP execution language to access the business data from the request context.

2. Assign the JSP documents to a task.

   Open the human task in the human task editor. In the client settings, specify the location of the user-defined JSP documents and the role to which the customized form applies, for example, administrator. The client settings for Business Process Choreographer Explorer are stored in the task template. At run time these settings are retrieved with the task template. For more information about adding JSP documents to human tasks, see the WebSphere Integration Developer information center.

3. Package the user-defined JSP documents in a Web archive (WAR file).

   You can either include the WAR file in the enterprise archive with the module that contains the tasks or deploy the WAR file separately.

The customized forms are rendered in Business Process Choreographer Explorer at run time.

User-defined JSP fragments:

The user-defined JSP fragments are imbedded in an HTML form tag. At run time, Business Process Choreographer Explorer includes these fragments in the rendered page.

The user-defined JSP fragment for the input message is imbedded before the JSP fragment for the output message.

```
<html....>
   ...
   <form...>
     Input JSP (display task input message)

     Output JSP (display task output message)

   </form>
   ...
</html>
```

Because the user-defined JSP fragments are embedded in an HTML form tag, you can add input elements. The name of the input element must match the XML Path Language (XPath) expression of the data element. It is important to prefix the name of the input element with the provided prefix value:

```
<input id="address"
       type="text"
       name="${prefix}/selectPromotionalGiftResponse/address"
       value="${messageMap['/selectPromotionalGiftResponse/address"]}
       size="60"
       align="left" />
```

The prefix value is provided as a request attribute. The attribute ensures that the input name is unique in the enclosing form. The prefix is generated by Business Process Choreographer Explorer and it should not be changed:

```
String prefix = (String)request.getAttribute("prefix");
```

The prefix element is set only if the message can be edited in the given context. Output data can be displayed in different ways depending on the state of the human task. For example, if the task is in the claimed state, the output data can be modified. However, if the task is in the finished state, the data can be displayed only. In your JSP fragment, you can test whether the prefix element exists and render the message accordingly. The following JSTL statement shows how you might test whether the prefix element is set.

```
...
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%
...
<c:choose>
  <c:when test="${not empty prefix}">
    <!--Read/write mode-->
  </c:when>
  <c:otherwise>
    <!--Read-only mode-->
  </c:otherwise>
</c:choose>
```

Accessing message data from the customized form:

You can access the request attributes in the message data either programmatically or by using an expression language.

The expression language uses expressions similar to XML Path Language (XPath) expressions to access the parts of the message data. If you use an expression language, you must add the corresponding tag libraries to the user-defined JavaServer Pages (JSP) document.

The following attributes are stored in the request context:

- message (Type: commonj.sdo.DataObject)

  The message attribute contains the message. The message can be a commonj.sdo.DataObject object or a primitive type.
- messageMap (Type: java.util.Map)

  This attribute represents the message content as a map. The keys of the message map are strings that represent XPath expressions to access the leaf nodes of the business object. The values in the map are the string representations of the data values of the business objects. When applicable, these values are the result of a locale sensitive conversion in Business Process Choreographer Explorer according to the locale that is specified in the browser. For example, the numerical value that represents the process state is converted into a string.
- messageValidationErrors

  This attribute accesses the validation errors from a submitted form. The value is a map that maps the XPath expression for accessing the incorrect property to the validation message.

1. Access message data.

   You can access the input and output data from the user-defined JSPs either programmatically with request attributes or by using an expression language.

   - Access the data using request attributes.

     You can obtain the message as a comonj.sdo.DataObject object or as the complete message map. The following example shows how to obtain the message as a data object.

     ```
     commonj.sdo.DataObject msg =
       (commonj.sdo.DataObject)request.getAttribute("message")
     ```

     The following example shows how to access the complete message.

```
java.util.Map msgMap =
  (jave.util.Map)request.getAttribute("messageMap")
```

- Access the data using an expression language statement.

```
...
${messageMap['/selectPromotionalGiftRequest/name']}
....
```

2. **Optional:** Access input data from the output JSP page.

   If the output JSP page requires access to the input message, you have to transport the data explicitly from the input JSP document to the output JSP document.

   You can use JSTL statements and the message map to pass the input message to the output JSP document as shown in the following example.

   a. Add the tag library to the JSP document.

   ```
   <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
   ```

   b. In the input JSP document, store the message map in the request.

   ```
   <c:set var="inputData" value="${messageMap}" scope="request" />
   ```

   c. In the output JSP document, access the request to get the complete message map.

   ```
   Map inputData = (Map)request.getAttribute("inputData");
   ```

Activating Business Process Choreographer:

After configuring the business process container and human task container, you must restart the servers where they were installed.

To activate Business Process Choreographer:

1. If you installed the business process container and human task container on a cluster of application servers, restart the cluster.
2. If you installed the business process container and human task container on one application server, restart the application server.
3. To verify that the business process container and human task container applications started successfully, make sure that no error messages exist in the `SystemOut.log` file for the application server. On a cluster, check the log for all application servers in the cluster.

Business Process Choreographer is running.

You are ready to verify that Business Process Choreographer is working.

Verifying that Business Process Choreographer works:

Run the Business Process Choreographer installation verification application.

The application server, database system, and messaging service must be running.

1. Using either the administrative console or the wsadmin command, install the application in *install_root*`/installableApps/bpcivt.ear`. Errors will occur at this stage if the Business Process Choreographer database cannot be accessed. These problems can be caused if the database system is not running, if any database clients are not correctly configured, or if errors were made defining the data source, for example, entering an invalid user ID or password. After the enterprise application is installed, it is in the state stopped, and any process and task templates that it contains are in the state started. No process or task instances can be created until the application is started.

2. Select the application BPCIVTApp and click **Start** to start the application. At this point, the input queues are read for the first time. Errors will occur at this stage if the queue manager is not running, or if any mistakes were made defining the JMS provider or JMS resources.
3. Verify that the application works. Using a Web browser, open the following page:

   `http://app_server_host:port_no/bpcivt`

   Where *app_server_host* is the network name for the host of the application server and *port_no* is the port number used by Business Process Choreographer Explorer. The port number depends on your system configuration. You should see a message indicating success.
4. **Optional:** Stop and remove the BPCIVTApp application.

Business Process Choreographer works.

Understanding the startup behavior of the business process container:

This topic explains why the business process container is unavailable until all enterprise applications are started.

When the business process container is started or restarted, no messages in the internal queue are processed until all enterprise applications are started. It is not possible to change this behavior. The time that business process container is unavailable during a restart depends on how long it takes until all enterprise applications are started. This behavior is necessary to prevent the business process engine from navigating processes with associated enterprise applications that are not running.

Starting to process messages in the internal queue before all applications are started would result in ClassNotFound exceptions.

Configuring the Common Event Infrastructure

You must configure the necessary resources and services before you can use the Common Event Infrastructure.

1. Configure the event database. See *"Configuring the event database" on page 213*.
2. Deploy the Common Event Infrastructure application. See *"Deploying the Common Event Infrastructure application" on page 220*.
3. Start the application server.
4. **Optional:** Deploy a message driven bean. You can deploy a message driven bean in one of the following ways:
   - Use the default messaging service and associate the message queue to the default emitter profile. See *"Configuring default event messaging" on page 223*.
   - Use an alternative messaging service. See *"Configuring event messaging using another JMS provider" on page 224*.

   The Common Event Infrastructure is installed and ready to use. By default, the Common Event Infrastructure service and the application events service are started when the application server starts.
5. **Optional:** Change the default configuration settings for services and resources. These services and settings include:
   - Common Event Infrastructure service. See *"Configuring the Common Event Infrastructure Service" on page 228*.
   - Application events service. See *"Configuring the Events service" on page 229*.

- Emitter factory profile. See *"Creating an emitter factory profile" on page 227*.
- Event group. See *"Creating an event group" on page 228*.

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 with a Cloudscape event database. Migration is not supported for a Cloudscape event database.

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 The type of message. This can be any of the following:
- Entry
- Exit
- Error
- Information
- Warning

Message
The text of the message.

Configuring a Cloudscape database:

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

You must configure the Cloudscape event database prior to 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 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 `sysdeflt`.
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 run-time 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=false`, you must complete the database configuration by manually running the generated scripts. The default value in the sample database response file is `EXECUTE_SCRIPTS=true`.

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=false, 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 run-time 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:
   - $WAS_HOME/event/dbscripts/db2zos/ddl/cr_db.db2
   - $WAS_HOME/event/dbscripts/db2zos/ddl/cr_db_catalog.db2
   - $WAS_HOME/event/dbscripts/db2zos/ddl/cr_tbl.db2
   - $WAS_HOME/event/dbscripts/db2zos/ddl/cr_tbl_catalog.db2
   - $WAS_HOME/event/dbscripts/db2zos/ddl/ins_metadata.db2
   - $WAS_HOME/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 run-time environment in which you are configuring the event database.

4. Run the $WAS_HOME/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.

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 invalid 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 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**.

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 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 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**.

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:

- 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.

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 Application Server runtime environment where you want to deploy the application. (Replace *profile_path* with the path to the directory containing the WebSphere Application 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 find out 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.

Deploying the Common Event Infrastructure application:

The event server enterprise application must be deployed in each WebSphere run-time 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 Application Server run-time environment where you want to deploy the application. (Replace *profile_path* with the path to the directory containing the WebSphere Application Server profile.)
2. Run the **event-application.jacl** script using the **wsadmin** command:

```
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 find out 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 run-time 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 $WAS_HOME/event/application directory for the profile defining the WebSphere Process Server run-time 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:

```
$WAS_HOME/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.

Run-time 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 documentation for more information about the administrative console and the wsadmin tool.

In most circumstances, only certain properties need to be configured. For complete 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 *"Run-time configuration" on page 225*.

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 223*.

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; usually, 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; usually, 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. Usually, 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. Usually, 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 223*.

Creating an emitter factory profile:

An emitter factory profile defines properties that are used for an emitter factory, which event sources use to create emitters.

The properties in an emitter factory profile affect the behavior of any emitter that is created using the associated emitter factory. You can use the default emitter factory profile or create additional profiles for your event sources to use. You

might want to create an additional emitter factory profile to specify a different transaction mode or synchronous transmission profile. For more information about how these options affect the behavior of the emitter, see *Creating an event source*.

To create an emitter factory profile, follow these steps:

1. In the WebSphere administrative console, click **Resources > Common Event Infrastructure Provider > Emitter Factory Profile > New**.
2. Specify the properties of the new profile. Refer to the online help for the Emitter Factory Profile Settings page for detailed information about these properties.
3. Click **OK** to save your changes and create the emitter factory profile.

Event sources can now use the configured emitter factory to obtain emitters.

Creating an event group:

An event group defines a logical collection of events based on the content of their property data.

You can use an event group to query events from the event server. You can also associate an event group with a Java Message Service (JMS) destination for asynchronous event distribution.

To create an event group, follow these steps:

1. **Optional:** Set up one or more JMS destinations for the event group. An event group can be associated with one JMS topic, and one or more JMS queues. Refer to the documentation for your JMS provider for information on how to create JMS destinations and connection factories and bind them into a Java Naming and Directory Interface (JNDI) namespace.
2. Create a new event group profile. In the WebSphere administrative console, click **Resources > Common Event Infrastructure > Provider > Event Group Profile List > *event_group_profile_list* > Event Group Profiles > New**.
3. Specify the properties of the event group profile. Refer to the online help for the Event Group Profile Settings page for detailed information about these properties.
4. Click **OK** to save your changes and create the event group profile.

Event consumers can now specify the event group when querying events. If event distribution is enabled in the event server profile, events belonging to the event group are also published to JMS destinations that are specified in the event group profile. Event consumers can then receive events asynchronously by subscribing to the appropriate destinations.

Configuring the Common Event Infrastructure Service:

Configure the Common Event Infrastructure Service to process events in WebSphere applications and processes.

This task assumes you have installed the Common Event Infrastructure server, deployed a Message Driven Bean application (WebSphere messaging service or a JMS messaging service), and that you are logged into the WebSphere Process Server administrative console.

You configure the Common Event Infrastructure Service, to enable or disable the event server, by modifying the **Enable service at server startup** property, which specifies whether or not the service is started automatically for a specific application server.

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. The properties for that Application server are displayed in the Configuration tab.
3. Look in the **Container Settings** table, and expand the **Container Services** menu.
4. Click on **Common Event Infrastructure Service** to display its properties.
5. Under General Properties, select or clear the **Enable service at server startup** property check box.

   Note: If you disable the Common Event Infrastructure Service from the administrative console, the enterprise application is also disabled automatically. If you use the `wsadmin` tool to disable the service from a script, you must disable the enterprise application separately.

   CAUTION:
   Disabling the Common Event Infrastructure 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.

   Selected
   [Default] The Common Event Infrastructure Service starts when the application server starts. This enables applications that generate events to run on the application server.

   Cleared
   The Common Event Infrastructure Service will not start when the application server starts. Applications that generate events will not start on this application server.

   Any attempt to start an application that uses the Common Event Infrastructure Service is rejected, and a message is issued. The server will continue to start without the application.
6. To save and apply your configuration, click the **OK** button.
7. Stop and restart the application server for the changes to take effect.

Configuring the Events service:

Perform these steps to switch the Events service on or off.

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, as well as 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 or not 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 **Events service**. 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, as well as 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. The name that is provided is a 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 JNDI name.
6. To save and apply your configuration, click the **OK** button.
7. Stop and restart the Application server for the changes to take effect.

Events service settings:

The Events service provides access to the events infrastructure for WebSphere applications and ensures that information about the WebSphere server is automatically included in each event passed to the events infrastructure.

To view this administrative console page, click **Servers > Application Servers >***server_name* then, under **Business Integration**, click on **Events service**.

Enable service at server startup

This setting specifies whether the server attempts to start the Events service automatically.

Selected (default)
When the application server starts, it attempts to start the Events service automatically.

Cleared
The server does not try to start the Events service. If applications that run on this server need to use the service, the system administrator must start the service manually or select this property, then restart the server.

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.

Events emitter infrastructure factory JNDI name

The JNDI name for the factory that is used to create event emitters. The Event Emitter Factory is defined as a resource of the Common Event Infrastructure Provider. By default, this will specify the default Event Emitter Factory JNDI value of `com/ibm/events/configuration/emitter/Default`.

The event submitter first passes the event to its listener registry and then (as a common base event) on to the event emitter. The event emitter filters the event and, if required, passes it on to be stored in the event data store and / or distributed to the consumers through JMS. Data type = String.

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 16. 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: • 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: • EventAccess.eventExists() • EventAccess.queryEventByGlobalInstanceId() • EventAccess.queryEventsByAssociation() • EventAccess.queryEventsByEventGroup() • eventquery.jacl
eventUpdater	Event consumers that need to update events stored in the event database. This role provides access to the following interfaces: • EventAccess.updateEvents() • EventAccess.eventExists() • EventAccess.queryEventByGlobalInstanceId() • EventAccess.queryEventsByAssociation() • EventAccess.queryEventsByEventGroup() • eventquery.jacl
eventCreator	Event sources that need to submit events to an emitter using synchronous EJB calls. This role provides access to the following interfaces: • Emitter.sendEvent() • Emitter.sendEvents() • emitevent.jacl **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.
catalogAdministrator	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.

Table 16. Security roles and user types (continued)

Security role	User types
catalogReader	Event catalog applications that need to retrieve event definitions from the event catalog. This role provides access to the following interfaces: • 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";
```

Applying product maintenance

Because WebSphere Process Server for z/OS is installed and configured into the WebSphere Application Server, the maintenance applied to WebSphere Process Server for z/OS is done so through the WebSphere Application Server product, using the WebSphere Application Server techniques for applying product maintenance.

Contact the IBM Software Support Center for information about preventive service planning (PSP) upgrades for the product. For more information about PSP upgrades for WebSphere Process Server for z/OS, see the *Program Directory for WebSphere Process Server for z/OS*. Although the Program Directory contains a list of required program temporary fixes (PTFs), the most current information is available from the IBM Software Support Center.

Use the following procedure whenever you want to apply a new service release to your system.

See *Applying product maintenance* in the WebSphere Application Server for z/OS information center for a description of how to apply product maintenance

You can maintain service to clients when upgrading the host cluster of WebSphere Application Server for z/OS.

Applying a service level or restoring to the previous accepted service level

Because WebSphere Process Server for z/OS is installed and configured into the WebSphere Application Server, the service level applied to WebSphere Process Server for z/OS is done so through the WebSphere Application Server product, using the WebSphere Application Server techniques for applying service level or restoring to the previous accepted service level

Service that is applied to the product data sets and product HFS occasionally requires corresponding changes to be made to the configuration HFS for existing application serving environments that configure at a lower service level. Most of these ″post-maintenance″ or ″post-install″ updates can be performed automatically. This is done by the post-installer. See *Applying a service level or restoring to the previous accepted service level* in the WebSphere Application Server for z/OS information center for a description of how to apply service

Uninstalling

The Uninstalling section describes how to uninstall WebSphere Process Server for z/OS, Version 6.0.1

Overview

Like the install on z/OS, the uninstall is run from a command line.

You run the uninstall command using **zSMPInstall.sh** with the appropriate command arguments and parameters. Running the command **zSMPInstall.sh** with the `-uninstall` with appropriate command arguments and parameters restores the WebSphere environment back to the level it was at prior to installation.

Note: The Common Event Infrastructure and Business Process Choreographer components are uninstalled separately. For information on how to uninstall Common Event Infrastructure and Business Process Choreographer, consult the appropriate help topics.

The uninstall process results in the following actions:

- Disables WebSphere Process Server features by running Configuration Manager scripted actions.

  This will remove any administrative console plug-in extensions.

- Removes profile augmentation using WSPROFILE scripted actions.
  This will unaugment the WebSphere Application Server default profile.
- Deletes the post install file and removes code base permissions

Run the install script to uninstall WebSphere Process Server for z/OS

Run **zSMPInstall.sh** with the `-uninstall` keyword to remove the WebSphere Process Server for z/OS definitions from the installed configuration.

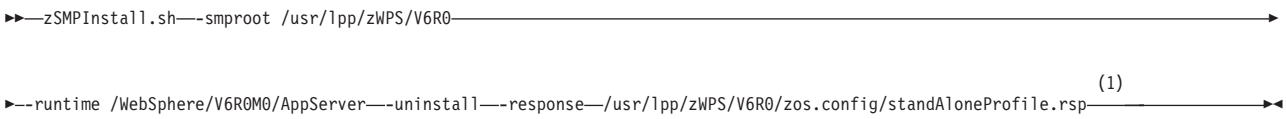
The WebSphere Process Server for z/OS install process assumes a working knowledge of z/OS UNIX System Services. You may want to have access to the following documentation:

- z/OS V1R7.0 UNIX System Services User's Guide
- z/OS V1R7.0 UNIX System Services Command Reference

Perform this task when your goal is to remove the WebSphere Process Server for z/OS definitions to restore your WebSphere environment to the level it was prior to installing and configuring WebSphere Process Server for z/OS.

1. Access the OS/390 UNIX command shell. Enter the TSO command OMVS at the ISPF Command Shell or TSO OMVS from any other ISPF panel.
   Once you're in the UNIX shell, a command prompt (usually a dollar ($) or pound (#) sign) indicates that the system is ready to accept input.
2. From the command prompt, run the install command. Use the following syntax diagram as guide on how run the command:

Uninstall command syntax diagram



Notes:

1 Type in one of the absolute path names of the response file according to your configuration. The path file names above assume the uninstaller is using the default response files. If the response file was customized, the path name would need to represent the absolute path of the customized file.

> **Note:** The instructions described above indicate how to run the install script from a OS/390 UNIX command shell. Alternatively you can run the install script from a prompt using Telnet. To run the install script using Telnet, type the following from the system prompt: **telnet TCPIP-Adress 2023** and run the command as indicated in the instructions above.
>
> The product should be uninstalled for the appropriate configuration depending on the response file properties used.

The following will be accomplished with the -uninstall option.

- Disable product features by running Configuration Manager Scripted actions that remove the associated Administrative console plug-in extensions
- Un-augments the default profile.

Check the associated log file and trace file if the uninstall was not successful.

- **Standard out messages**

Standard output messages display directly on the screen from which you ran the installer script to uninstall the product. You can choose to redirect these messages to a file by using redirect symbol and a file name at the end of the command line. For example, adding the syntax `>run.log` to the end of the install command redirect the standard output messages to the file named **run.log** in the present working directory. The standard out messages display as follows:

```
parsing command arguments...
parsing arguments complete
setting up configuration...
runtimeRootDirName is: /WebSphere/V6R0M0/AppServer
WAS_HOME is: /WebSphere/V6R0M0/AppServer
WBI_HOME is: /WebSphere/V6R0M0/AppServer
running Configuration Manager for uninstall...
Configuration Manager for uninstall complete
unaugmenting profile(s)...
unaugmenting profile(s) complete
```

- **Log file**

  Log messages are written to the **zSMPInstall.log** file in the run-time directory. Standard location for this file is /WebSphere/V6R0M0/AppServer/logs/wbi/zSMPInstall.log.

- **Trace file**

  Review the zSMPInstall.trace (ASCII) file in the run-time directory. Standard location for this file is /WebSphere/V6R0M0/AppServer/logs/wbi/zSMPInstall.trace..

  There should be no error messages (i.e. messages with an ″E″ suffix) in the trace file.

Perform other troubleshooting tasks as follows:

- **Review the actions of the Update Configuration Manager task.**

  These actions are recorded by writing to a log file (ASCII). The log file name is `cmtInstall.log`.

  Standard location for this file is in directory /WebSphere/V6R0M0/AppServer/logs/wbi.

  Search this Configuration Manager log for `>SEVERE<` or `>WARNING<` level messages to determine overall error in processing.

  Each Ant script run from the install directory writes to it's own log (ASCII).

  Default name for the install directory that contains the ant scripts is:/WebSphere/V6R0M0/AppServer/properties/version/install.wbi/6.0.0.0/config/full/uninstall.

  The resulting ant logs are written to the product log directory. The default name for this directory is /WebSphere/V6R0M0/AppServer/logs/wbi. Ant logs include the following (review these logs to determine errors in processing) :

  – 90SDeleteFirstStepsFilesWBI.ant.log
  – 90SRemoveJavaOptions.ant.log
  – 90SUninstallCEI.ant.log
  – 98SUndeployBPCAdminConsolePlugins.ant.log
  – 98SUndeployServerAdminConsolePlugins.ant.log
  – 99SUndeployCoreAdminConsolePlugins.ant.log

  Each of these logs should contain a Build Successful message.

- **Review the contents of the Unaugment Log**

  The unaugment profile task records it's actions by writing to a log file (ASCII). The log file name has the name **wasprofile_unaugment_default.log.** Standard location for this file is in directory /WebSphere/V6R0M0/AppServer/logs/

wasprofile. Search this WAS Profile Augment log for >SEVERE< or >WARNING< level messages to determine overall error in processing.

There should be no SEVERE messages.

Removing the Business Process Choreographer configuration

Use this task to remove the business process container, human task container, Business Process Choreographer Explorer, and the associated resources.

1. Ensure that all the stand-alone servers, the database, and the application server (or at least one application server per cluster) are running.
2. For each enterprise application that contains human tasks or business processes, stop and uninstall all human task and business process templates, then uninstall the application.
3. Perform one of the following actions:
   - To uninstall the business process container, human task container, Business Process Choreographer Explorer, and the associated resources, perform "Using a script to remove the Business Process Choreographer configuration."
   - If you want to reuse parts of the existing configuration, perform "Using the administrative console to remove the Business Process Choreographer configuration" on page 239.

The Business Process Choreographer configuration has been removed.

Using a script to remove the Business Process Choreographer configuration

Use this task to remove the business process container, task container, and Business Process Choreographer Explorer configuration, and the associated resources.

Before you can remove the Business Process Choreographer configuration, you must stop all process and task templates, delete all process and task instances, then stop and remove the configuration for all enterprise applications that contain business processes or human tasks.

1. Change to the Business Process Choreographer `sample` directory: Type the following:

   `cd` *`install_root`*`/ProcessChoreographer/sample`
2. Run the script `bpeunconfig.jacl`. Use the appropriate command in the following table. In the following cases, also specify the appropriate options:
   - For stand-alone servers, stop the application server and use the `conntype none` option. This step ensures that any Cloudscape databases are not locked and can be removed automatically.
   - In a Network Deployment (ND) environment, run the script, as follows
     – If the deployment manager is not running, run the script on the deployment manager, using the `conntype none` option.
     – If the deployment manager is running, stop the application server from which the configuration is to be removed, then run the script, omitting the `conntype none` option.

       When the script is running on the application server node from which the Business Process Choreographer configuration is to be removed, the script can automatically delete any Cloudscape databases.
   - If WebSphere security is enabled, specify also the user ID and password:

     `-userid` *`userID`* `-password` *`password`*

- If you are not configuring the default profile, specify also the profile name:
  `-profileName profileName`

Option	Description
For a single server on z/OS, use the command:	*install_root*\bin\wsadmin.bat -f bpeunconfig.jacl -server *Server* -node *Node*
For a cluster on z/OS, use the command:	*install_root*\bin\wsadmin.bat -f bpeunconfig.jacl -cluster *Cluster* [-*deleteDatabase*]

When removing the configuration for a single server with WebSphere security enabled, enter the command:

```
install_root/bin/wsadmin.sh -f bpeunconfig.jacl  -server Server -node Node
            -userid userID  -password password
```

When removing the configuration for a single server with WebSphere security disabled, enter the command:

```
install_root/bin/wsadmin.sh -f bpeunconfig.jacl  -server Server -node Node
```

When removing the configuration for a cluster with WebSphere security enabled, enter the command:

```
install_root/bin/wsadmin.sh -f bpeunconfig.jacl -cluster Cluster
            -userid userID  -password password
```

When removing the configuration for a cluster with WebSphere security disabled, enter the command:

```
install_root/bin/wsadmin.sh -f bpeunconfig.jacl -cluster Cluster
```

Where:

Server The name of the application server. If only one server exists, this parameter is optional.

Node The name of the node. This is optional. If the node is omitted, the local node is used.

Cluster
The name of the cluster.

If you omit a parameter, you are prompted for it.

3. **Optional:** Delete the database used by Business Process Choreographer.
4. **Required:** Delete the database used by WebSphere default messaging. This database cannot be reused in a new configuration. For both the Business Process Choreographer database and the messaging database the following apply:
   - The bpeunconfig.jacl script lists the databases that were used by the configuration that has been removed. You can then more easily identify the databases that are to be removed.
   - When Cloudscape is the messaging database, the bpeunconfig.jacl script optionally removes the database, unless it is locked by a running application server. (Stop the server, and use the `conntype none` option.) You can specify either of the following choices:
     – The `deleteDB` command line option, for which the choices are:
       - `yes` (the default)
       - `no`
     – To be prompted
5. **Optional:** Delete the queue manager used by Business Process Choreographer.

The Business Process Choreographer applications and associated resources (such as scheduler, data sources, listener ports, connection factories, queue destinations, activation specs, work area partition, mail session, and authentication aliases) have been removed.

Using the administrative console to remove the Business Process Choreographer configuration:

Use this task to remove part or all of the business process container, task container, and Business Process Choreographer Explorer configuration, and the associated resources.

Before you can remove the Business Process Choreographer configuration, you must stop all process and task templates, delete all tasks and process instances, then stop and uninstall all enterprise applications that contain business processes or human tasks.

1. Uninstall the Business Process Choreographer enterprise applications.
   a. Display the enterprise applications.

      In the administrative console, select **Applications → Enterprise Applications**.
   b. Identify the scope of the Business Process Choreographer installation.

      Look for applications named `BPEContainer_`*`scope`*, `TaskContainer_`*`scope`*, and `BPCExplorer_`*`scope`*.
      - If Business Process Choreographer was installed on an application server, *scope* has the value of *`nodeName_serverName`*.
      - If Business Process Choreographer was installed on a cluster, *scope* has the value of *`clusterName`*.
   c. **Optional:** If you installed the business process container, uninstall it.
      1) Select `BPEContainer_`*`Scope`*, then click **Stop**.
      2) Select the application again, then click **Uninstall → OK → Save → Save**.
   d. **Optional:** If you installed the human task container, uninstall it.
      1) Select `TaskContainer_`*`Scope`*, then click **Stop**.
      2) Select the application again, then click **Uninstall → OK → Save → Save**.
   e. **Optional:** If you installed Business Process Choreographer Explorer, uninstall it.
      1) Select `BPCExplorer_`*`Scope`*, then click **Stop**.
      2) Select the application again, then click **Uninstall → OK → Save → Save**.
2. Remove all or any of the following resources that you do not want to reuse:
   a. **Optional:** Find the Business Process Choreographer data source (the default name is `BPEDataSource`*`dbType`*) and note its associated authentication data alias (if any) and Java Naming and Directory Interface (JNDI) name before removing it (for a single server, the default name is `jdbc/BPEDB`).

      To find the data sources:
      1) Click **Resources → JDBC Providers**.
      2) If Business Process Choreographer was installed on an application server, select **Server**.
      3) If Business Process Choreographer was installed on a cluster, select the cell.
      4) Click **Apply**.
      5) Select the appropriate JDBC provider, then click **Data sources**.

6) If you are using an Oracle database management system, remove also a second data source: `BPEDataSourceOracleNonXA`.

b. **Optional:** For a database other than a Cloudscape database, remove the JDBC provider of the data source identified in step 2 on page 239, unless it contains further data sources that you still need.

c. **Optional:** Remove the appropriate connection factories and queues.

- For default messaging, before you remove the connection factories, note their associated authentication data aliases. Then remove the JMS connection factories and JMS queues.
  1) Click **Resources → JMS Providers → Default messaging**.
  2) On the Default messaging provider pane perform one of the following:
     – If you configured Business Process Choreographer on a cluster, select **Cluster** then click **Apply**.
     – If you configured Business Process Choreographer on a server, select **Server** then click **Apply**.
- For WebSphere MQ, remove the JMS queue connection factories and JMS queue destinations.
  1) Click **Resources → JMS Providers → WebSphere MQ**.
  2) On the WebSphere MQ messaging provider pane, select **Server**. Then click **Apply**.

  If you configured Business Process Choreographer on a cluster, you must repeat this for each server that is a member of the cluster.

For the business process container the JNDI names are normally as follows:

```
jms/BPECF
jms/BPECFC
jms/BPEIntQueue
jms/BPEApiQueue
jms/BPERetQueue
jms/BPEHldQueue
```

For the human task container the JNDI names are normally as follows:

```
jms/HTMCF
jms/HTMIntQueue
jms/HTMHldQueue
```

d. **Optional:** If you are using WebSphere default messaging as the JMS provider, remove the activation specifications.

1) Click **Resources → JMS Providers → Default messaging → JMS activation specification**.
2) Remove the following activation specifications:

```
BPEApiActivationSpec
BPEInternalActivationSpec
HTMInternalActivationSpec
```

e. **Optional:** If you are using WebSphere MQ as the JMS provider, remove the listener ports.

1) Click **Servers → Application servers →** *serverName*.
2) Under Communications, click **Messaging → Message Listener Service → Listener Ports**.

3) On the Application servers pane, remove the following listener ports:

   `BPEInternalListenerPort`

   `BPEApiListenerPort`

   `BPEHoldListenerPort`

   `HTMInternalListenerPort`

f. **Optional:** Delete the authentication data aliases.

- If the data source identified in step 2 on page 239 had an authentication data alias, remove that alias.

  Usually, the alias for the database is named *cellName*/`BPEAuthDataAlias`*dbType_Scope*, where:

  cellName
  The name of the cell

  dbType
  The database type

  Scope
  One of the values given in step 1b on page 239

- If any of the connection factories identified in step 2c on page 240 have an authentication data alias, remove the alias.

  Usually, the alias for the database is named *cellName*/`BPEAuthDataAliasJMS_`*Scope*, where:

  cellName
  The name of the cell

  Scope
  One of the values given in step 1b on page 239

The authentication data alias is in **Security** → **Global security** → **JAAS Configuration** → **J2C Authentication data**.

g. **Optional:** Remove the scheduler configuration for the data source JNDI name.

1) Click **Resources** → **Schedulers**.
2) Select **Server**. Then click **Apply**.
3) On the Schedulers pane, note the work manager name, then select and delete the scheduler `BPEScheduler`.

h. **Optional:** Remove the work manager.

1) Click **Resources** → **Asynchronous beans** → **Work managers**.
2) Select **Server**. Then click **Apply**.
3) On the Work managers pane, select and delete the work manager whose name you noted in step 2g.

i. **Optional:** Remove the work area partition.

1) Click **Servers** → **Application servers** → *serverName*.
2) Under Business Process Services, click **Work area partition service**.
3) On the Application servers pane, select and delete the work area partition `BPECompensation`.

j. **Optional:** Remove the mail session.

1) Click **Resources** → **Mail Providers**.
2) On the Mail Providers pane, select **Cell**. Then click **Apply**.
3) Click **Built-in Mail Provider**.
4) Under Additional Properties, select **Mail sessions**.

5) Select and delete `HTMMailSession_Scope`, where *Scope* is the scope identified in step 1b on page 239

k. In a cluster, repeat the removal of any server level resources for each cluster member.

l. Save your configuration changes.

m. Restart the application server.

3. **Optional:** If you use WebSphere default messaging for Business Process Choreographer, you can delete the bus member, bus, and data source:
   a. Click **Service integration** → **Buses** → **BPC.*cellName*.Bus** → **Messaging engines**.
   b. Select the messaging engine:
      - ***nodeName.serverName*-BPC.*cellName*.Bus** if you configured Business Process Choreographer on a server.
      - ***clusterName*-BPC.*cellName*.Bus** if you configured Business Process Choreographer in a cluster.

      Note: If you configured Business Process Choreographer to use a remote messaging engine, *clusterName* might not match the name of the cluster where you configured Business Process Choreographer.
   c. In *Additional Properties*, select *Data store*, and note the JNDI name for the data source.
   d. Go to **Service integration** → **Buses** → **BPC.*cellName*.Bus** → **Bus members** and remove the bus member identified by one of the following:
      - `Node=nodeName, Server=serverName` if you configured Business Process Choreographer on a server.
      - `Cluster=clusterName` if you configured Business Process Choreographer on a cluster.
   e. **Optional:** If you removed the last member of the bus BPC.*cellName*.Bus, you can also remove the bus.
   f. **Optional:** Remove the data source. Click **Resources** → **JDBC Providers** → **Server** → **Apply** → **Cloudscape JDBC Provider** → **Data Sources**, then delete the data source that you noted in step 3c.
4. **Optional:** If you configured Business Process Choreographer on a cluster, delete the BPC_REMOTE_DESTINATION_LOCATION variable: Click **Environment** → **WebSphere Variables** → **Cluster** → **Apply**. Select the variable named `BPC_REMOTE_DESTINATION_LOCATION`, then click **Delete**.
5. Click **Save** to save all your deletions in the master configuration.
6. **Optional:** Delete the Business Process Choreographer database.
7. **Optional:** If you are using WebSphere MQ, delete the queue manager used by Business Process Choreographer.
8. If you use WebSphere default messaging for Business Process Choreographer, delete the datastore for the message engine. If you use the default data store, you can delete the data store by deleting the following directory:
   - On Windows systems, delete

     install_root\profiles*profileName*\databases\com.ibm.ws.sib*nodeName.serverName*-BPC.*cellName*.Bus
   - On UNIX and Linux systems, delete

     install_root/profiles/*profileName*/databases/com.ibm.ws.sib/*nodeName.serverName*-BPC.*cellName*.Bus

The Business Process Choreographer configuration has been removed.

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.

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 find out the node name, run the *install_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.

scope
The scope of the Common Event Infrastructure configuration profile objects to be removed. This parameter is optional. If you specify a scope, a the JMS transmission profile and emitter factory profiles in the specified scope are removed. The valid values are `cell`, `node`, and `server`.

app_name
The name of the deployed messaging enterprise application you want to remove. This parameter is required.

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 *install_root*`/event/dbscripts` directory. The scripts for removing the JDBC provider are placed in database-specific subdirectories of the *install_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	z/OS (Windows script)	`rm_event_cloudscape.bat`	`rm_cloudscape_jdbc_provider.bat`

Type	Operating system	Database script	JDBC configuration script
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.)

Troubleshooting the install

If the product installation and configuration are not successful, use this information to help you assess and correct the problems.

This topic assumes that you have attempted to install and or configure but have not been successful.

You should be aware that the installer program records multiple indicators of success in the following ways:

- Standard output messages

  Standard output messages display directly on the screen from which you run the installer script (zSMPInstall.sh). You can choose to *redirect* these messages to a file by using redirect symbol and a file name at the end of the command line. For example, adding the syntax >run.log to the end of the install command redirect the standard output messages to the file named **run.log** in the present working directory.
- Log file messages

  Log messages for installation are written to the **zSMPInstall.log** file in the run-time directory. Standard location for this file is /WebSphere/V6R0M0/AppServer/logs/wbi/zSMPInstall.log.

  Log messages for configuration are written to the**zWPSConfig.log** or the **zWESBConfig.log** file in the run-time directory. Standard locations for these files are /WebSphere/V6R0M0/AppServer/logs/wbi/zWPSConfig.log and /WebSphere/V6R0M0/AppServer/logs/wbi/zWESBConfig.log respectively.
- Trace file messages

  The installation trace messages are written to the **zSMPInstall.trace** file in the run-time directory. Standard location for this file is /WebSphere/V6R0M0/AppServer/logs/wbi/zSMPInstall.trace.

Trace messages for configuration are written to the**zWPSConfig.trace** or the **zWESBConfig.trace** file in the run-time directory. Standard locations for these files are /WebSphere/V6R0M0/AppServer/logs/wbi/zWPSConfig.trace and /WebSphere/V6R0M0/AppServer/logs/wbi/zWESBConfig.trace respectively.

Make sure that you have installed and configured WebSphere Application Server for z/OS successfully. Refer to the installation troubleshooting information in the WebSphere Application Server for z/OS information center if you are having trouble installing and configuring WebSphere Application Server for z/OS.

For current information available from IBM Support on known problems and their resolution, see the WebSphere Process Server Support page.

1. Review the messages from Standard Out. There should be no error messages displayed. The standard output messages display on either on the screen from which you ran the install command or in a file that you specified by using the redirect (">") symbol on the command line.

   The following is an example of a successful execution of the install script with the **-install** option:

   ```
   parsing command arguments...
   parsing arguments complete
   setting up configuration...
   runtimeRootDirName is: /WebSphere/V6R0M0/AppServer
   WAS_HOME is: /WebSphere/V6R0M0/AppServer
   WBI_HOME is: /WebSphere/V6R0M0/AppServer
   set up configuration complete
   creating the symbolic links...
   invokeSymLink
   creation of symbolic links complete
   doing post install file updates...
   post install updates complete
   running Configuration Manager update...
   Configuration Manager update complete
   augmenting profile(s)...
   augmenting profile(s) complete
   ```

   Error messages indicate an unsuccessful install. Some errors as displayed in Standard Out will be self explanatory and can be easily corrected. If the install completed to the point where the log and trace files were created, continue with the following steps.

2. Review the zSMPInstall.log (ASCII) file in the run-time directory. Standard location for this file is /WebSphere/V6R0M0/AppServer/logs/wbi/zSMPInstall.log.

   Verify the response file values, including any -Z command line overrides, are correct.

   If there are error messages, try to determine which of the following tasks were in progress when the error occurred.

   - create symbolic links
   - create post install file
   - update codebase permissions
   - update Configuration Manager
   - augment profile(s)

   Knowing the task that was in progress at the time of an error will help you assess the information in the trace file.

3. Review the zSMPInstall.trace (ASCII) file in the run-time directory. Standard location for this file is /WebSphere/V6R0M0/AppServer/logs/wbi/zSMPInstall.trace.

On a successful install, only informational messages, for example messages that have a suffix of CWPIZ0044**I** should be listed in the trace file.

If warning messages (messages with a suffix of **W**) or error messages (messages with a suffix of **E**) are listed in the trace, further review is required.

If the warning or error occurred during the create symbolic links, create post install file, or update codebase permissions tasks, the trace message should contain information that will help you diagnose and correct the problem.

If the warning or error occurred in the update Configuration Manager task, proceed to the next step.

If the warning or error occurred in the augment profile(s) task, proceed to step 5.

4. Review the actions of the Update Configuration Manager task. These actions are recorded by writing to a log file (ASCII). The log file name is `cmtInstall.log`.

   Standard location for this file is in directory /WebSphere/V6R0M0/AppServer/logs/wbi.

   Search this Configuration Manager log for `>SEVERE<` or `>WARNING<` level messages to determine overall error in processing.

   Each Ant script run from the install directory writes to it's own log (ASCII).

   Default name for the install directory that contains the ant scripts is:/WebSphere/V6R0M0/AppServer/properties/version/install.wbi/6.0.0.0/config/full/install.

   The resulting ant logs are written to the product log directory. The default name for this directory is /WebSphere/V6R0M0/AppServer/logs/wbi. Ant logs include the following (review these logs to determine errors in processing) :

   - 90SConfigNoProfileFirstStepsWBI.ant.log
   - 90SConfigureWSProfileForWBI.ant.log
   - 90SConfigWBIMigrationScript.ant.log
   - 90SInstallCEI.ant.log
   - 90SUpdateJavaOptions.ant.log
   - 98SDeployBPCAdminConsolePlugins.ant.log
   - 98SDeployServerAdminConsolePlugins.ant.log
   - 99SDeployCoreAdminConsolePlugins.ant.log

   If there were no problems a **BUILD SUCCESSFUL** message displays at the end of the file.

5. Review the actions of the WebSphere Application Server profile augment task. The augment profile(s) task records it's actions by writing to a log file (ASCII). The log file name has the name **wasprofile_augment_default.log**. Standard location for this file is in the directory /WebSphere/V6R0M0/AppServer/logs/wasprofile.

   Search this Profile Augmentation log for `>SEVERE<` or `>WARNING<` level messages to determine overall error in processing.

   Individual ant action logs are located in /WebSphere/V6R0M0/AppServer/profiles/default/logs.

After troubleshooting the problems that caused the installation errors, and after you have run the installation script successfully, you should perform the following steps:

1. Start the application server.

2. Launch the Administrative Console and verify that the product components have been installed.

   For example, by installing WebSphere Process Server successfully, you should see evidence of Process Choreographer under the enterprise applications with names that start with `BPEContainer`, `BPCExplorer`, and `TaskContainer`.

Message reference for WebSphere Process Server for z/OS installation and configuration

The message reference for WebSphere Process Server for z/OS lists the message codes that may display while running the install script or when running the configuration script.

About the installation error messages

Use the data in the Explanation and User response fields to troubleshoot the WebSphere Process Server for z/OS message codes.

The message code displays as CWPIZyyyyz, where:

- CWPIZ = The WebSphere Process Server for z/OS message prefix
- yyyy = The numeric identifier assigned to the number
- z = Descriptor (E, I or W) for the type of message, where:
  - E = Error message
  - I = Informational message
  - W = Warning message

The WebSphere Process Server for z/OS installation error messages are documented in the information center under **Reference > Messages >CWPIZ**.

The WebSphere Process Server for z/OS installation error messages are written to the zSMPInstall.log file in the run-time directory. The standard default location for the log file is /WebSphere/V6R0M0/AppServer/logs/wbi/zSMPInstall.log.

The WebSphere Process Server for z/OS configuration error messages are written to the zWPSConfig.log file and the zWESBConfig.log file in the run-time directory. The standard default location for these log files are /WebSphere/V6R0M0/AppServer/logs/wbi/zWESBConfig.log and /WebSphere/V6R0M0/AppServer/logs/wbi/zWPSConfig.log respectively.

Log files

Various log files are created during the product install and configuration process.

Purpose

Consult the applicable logs if problems occur during the product install and configuration process.

Standard out messages redirected to log file

Standard out messages report high-level actions such as the starting and completing of the action that verifies the command line arguments.

By default, these messages display directly on the screen from which you run the product install script. However, you can *redirect* these messages to a file by using

redirect symbol and a file name at the end of the command line. For example, specifying `>run.log` at the end of the install command redirects the standard out messages to a file named `run.log` in the present working directory.

Standard out messages also report severe errors that occur prior to the Log and Trace File being opened. For instance, the following message block is display if the required keyword **-response** was not included in the install command.

```
parsing command arguments...
CWPIZ0101E -response keyword and value not specified on zSMPInstall.sh command line.
com.ibm.ws390.installer.InstallFailureException: -response keyword and value not specified
CWPIZ0017E  install task failed.
```

Log file

These messages include the messages written to Standard Out, but provide additional information and settings that were used by the installer program.

For instance, the following log portion shows the response properties and their values being used. It also shows the source and target directories being used during the creation of the symbolic links.

```
response property: profilePath=/WebSphere/V6R0M0/AppServer/profiles/default
response property: nodeName=SY1
response property: scaSecurityPassword=ibmuser
response property: dbType=Cloudscape
response property: ceiSampleJmsUser=ibmuser
response property: scaSecurityUserId=ibmuser
response property: configureScaSecurity=true
response property: mqUser=ibmuser
response property: serverName=server1
response property: adminBFMGroups=ibmuser
response property: profileName=default
response property: dbCreateNew=true
response property: ceiSampleJmsPwd=ibmuser
response property: cellName=SY1
response property: dbLocation=/WebSphere/V6R0M0/AppServer/cloudscape/databases/WBIDB
response property: mqPwd=ibmuser
response property: was.install.root=/WebSphere/V6R0M0/AppServer
response property: augment=
response property: ceiDbProduct=CLOUDSCAPE_V51_1
response property: wbi.install.root=/WebSphere/V6R0M0/AppServer
response property: ceiSampleServerName=server1
response property: templatePath=/WebSphere/V6R0M0/AppServer/profileTemplates/default.*
response property: dbName=WBIDB
set up configuration complete
creating the symbolic links...
Source=/usr/lpp/zWPS/V6R0M0

Target=/WebSphere/V6R0M0/AppServer
creation of symbolic links complete
doing post install file updates...
post install updates complete
running Configuration Manager update...
Configuration Manager update complete
```

Trace file

These messages are written to the **zSMPInstall.trace** file in the run-time directory.

The example below shows some preliminary informational messages and then a **CWPIZ0322E** error indicating that the required profileName property was not found in the response file that the user specified on the install script command line (nor was provided as a -Z override).

The subsequent **CWPIZ0017E** error message is a general message indicating the final outcome of the zSMPInstall.sh run.

```
[8/16/05 17:00:45:380 EDT] 0000000a ManagerAdmin  I   BBOO0222I:
TRAS0017I: The startup trace state is *=info.

[8/16/05 17:00:48:230 EDT] 0000000a WPSInstaller  I   BBOO0222I:
CWPIZ0044I Begin install task.

[8/16/05 17:00:48:273 EDT] 0000000a WPSInstaller  I   BBOO0222I:
CWPIZ0117I WPS installer log data will be written to
/WebSphere/V6R0M0/AppServer/logs/wbi/zSMPInstall.log.

[8/16/05 17:00:48:282 EDT] 0000000a WPSInstaller  I   BBOO0222I:
CWPIZ0024I WPS installer trace data will be written to
/WebSphere/V6R0M0/AppServer/logs/wbi/zSMPInstall.trace.

[8/16/05 17:00:48:292 EDT] 0000000a WPSInstaller  I   BBOO0222I:
CWPIZ0014I Trace specification is "*=all=disabled".

[8/16/05 17:00:48:298 EDT] 0000000a WPSInstaller  I   BBOO0222I:
CWPIZ0045I WPS SMP/E root directory is /zrockuser/wbi/Install.

[8/16/05 17:00:48:302 EDT] 0000000a WPSInstaller  I   BBOO0222I:
CWPIZ0052I WAS SMP/E root directory is /web/usr/lpp/zWebSphere/V6R0.

[8/16/05 17:00:48:307 EDT] 0000000a WPSInstaller  I   BBOO0222I:
CWPIZ0046I Destination application server root directory is
/WebSphere/V6R0M0/AppServer.

[8/16/05 17:00:48:314 EDT] 0000000a WPSInstaller  E   BBOO0220E:
CWPIZ0322E profileName property not specified in Response File.

[8/16/05 17:00:48:318 EDT] 0000000a WPSInstaller  E   BBOO0220E:
CWPIZ0017E install task failed.

A trace file from a zSMPInstall.sh executed with the trace specification argument
set to "*=all=enabled" provides additional debugging information. It may contain
information that is meaningful only to a developer. The following is a partial
trace using "*=all=enabled":

************ Start Display Current Environment ************
Host Operating System is z/OS, version 01.04.00
Java version = J2RE 1.4.2 IBM z/OS Persistent Reusable VM build cm142-20050623
(JIT enabled: jitc), Java
Compiler = jitc, Java VM name = Classic VM
was.install.root = /WebSphere/V6R0M0/AppServer
user.install.root = /WebSphere/V6R0M0/AppServer/profiles/default
Java Home = /web/usr/lpp/zWebSphere/V6R0/java/J1.4
ws.ext.dirs = /WebSphere/V6R0M0/AppServer/java/lib:/WebSphere/V6R0M0/AppServer/java/lib/
ext:/WebSphere/V6R0M0/AppServer/classes:/WebSphere/V6R0M0/AppServer/lib:/WebSphere/V6R0M0/AppServer/
installedChannels:/WebSphere/V6R0M0/AppServer/lib/ext:/WebSphere/V6R0M0/AppServer/deploytool/itp
/plugins/com.ibm.etools.ejbdeploy/runtime:/WebSphere/V6R0M0/AppServer/MQSeries/pubsubroot/lib
Classpath = /zrockuser/bbzconfig.jar:/WebSphere/V6R0M0/AppServer/lib/admin.jar:/WebSphere/V6R0M0
/AppServer/lib/ant.jar:/WebSphere/V6R0M0/AppServer/lib/bootstrapws390.jar:/WebSphere/V6R0M0
/AppServer/lib/bootstrap.jar:/WebSphere/V6R0M0/AppServer/lib/configmanager.jar:/WebSphere
/V6R0M0/AppServer/lib/emf.jar:/WebSphere/V6R0M0/AppServer/lib/ras.jar:/WebSphere/V6R0M0
/AppServer/lib/runtimefw.jar:/WebSphere/V6R0M0/AppServer/lib/utils.jar:/WebSphere/V6R0M0
/AppServer/lib/wasjmx.jar:/WebSphere/V6R0M0/AppServer/lib/wasproduct.jar:/WebSphere/V6R0M0
/AppServer/lib/wccm_base.jar:/WebSphere/V6R0M0/AppServer/lib/wjmxapp.jar:/WebSphere/V6R0M0
/AppServer/lib/wsanttasks.jar:/WebSphere/V6R0M0/AppServer/lib/wsexception.jar:/WebSphere
/V6R0M0/AppServer/lib/wsprofile.jar:/WebSphere/V6R0M0/AppServer/profiles/default/properties:
/WebSphere/V6R0M0/AppServer/properties:/WebSphere/V6R0M0/AppServer/lib/bootstrap.jar:/WebSphere
/V6R0M0/AppServer/lib/j2ee.jar:/WebSphere/V6R0M0/AppServer/lib/lmproxy.jar:/WebSphere/V6R0M0
/AppServer/lib/urlprotocols.jar:/WebSphere/V6R0M0/AppServer/lib/bootstrapws390.jar
Java Library path = /web/usr/lpp/zWebSphere/V6R0/java/J1.4/bin/classic/libjvm.so:/web/usr
/lpp/zWebSphere/V6R0/java/J1.4/bin/classic:/web/usr/lpp/zWebSphere/V6R0/java/J1.4/bin/:
/WebSphere/V6R0M0/AppServer/lib:/WebSphere/V6R0M0/AppServer/lib:/WebSphere/V6R0M0/AppServer
/MQSeries/pubsubroot/lib:/mqm/java/bin:/mqm/java/lib:/db2810/lib:/db2beta/db2710/lib:
/web/usr/lpp/WebSphere/lib:/lib:/usr/lib:/java/J1.3/bin:/java/J1.4/bin:/java/J5.0/bin:
/staf/lib:/WebSphere/V6R0M0/AppServer/lib:/usr/lib
Current trace specification = *=all
************ End Display Current Environment ************
[10/3/05 16:35:05:709 EDT] 0000000a ManagerAdmin  I   BBOO0222I: TRAS0017I:
The startup trace state is *=all.
[10/3/05 16:35:08:638 EDT] 0000000a WPSInstaller  >  setup Entry
                                /web/usr/wbi/zWebSphere/V6R0
                                APPSERVER
                                zSMPInstall.sh
                                -smproot
                                /web/usr/wbi/zWPS/V6R0
                                -runtime
                                /WebSphere/V6R0M0/AppServer
                                -response
                                /web/usr/wbi/zWPS/V6R0/zos.config/standAloneProfile.rsp
                                -prereqonly
                                -trace
                                *=all=enabled
[10/3/05 16:35:08:640 EDT] 0000000a WPSInstaller  3   logFileDeleted
                                true
[10/3/05 16:35:08:660 EDT] 0000000a WPSInstaller  I   BBOO0222I: CWPIZ0044I:
Begin install task.
[10/3/05 16:35:08:702 EDT] 0000000a WPSInstaller  I   BBOO0222I: CWPIZ0117I:
```

```
WPS installer log data will be written to /WebSphere/V6R0M0/AppServer/logs/wbi/zSMPInstall.log.
[10/3/05 16:35:08:712 EDT] 0000000a WPSInstaller  I   BBOO0222I: CWPIZ0024I:
WPS installer trace data will be written to /WebSphere/V6R0M0/AppServer/logs/wbi/zSMPInstall.trace.
[10/3/05 16:35:08:722 EDT] 0000000a WPSInstaller  I   BBOO0222I: CWPIZ0014I:
Trace specification is "*=all=enabled".
[10/3/05 16:35:08:726 EDT] 0000000a WPSInstaller  I   BBOO0222I: CWPIZ0052I:
WAS SMP/E root directory is /web/usr/lpp/zWebSphere/V6R0.
[10/3/05 16:35:08:730 EDT] 0000000a WPSInstaller  >  checkPathName Entry
                                /web/usr/wbi/zWPS/V6R0
[10/3/05 16:35:08:731 EDT] 0000000a WPSInstaller  <  checkPathName Exit
[10/3/05 16:35:08:732 EDT] 0000000a WPSInstaller  I   BBOO0222I: CWPIZ0045I:
WPS SMP/E root directory is /web/usr/wbi/zWPS/V6R0.
[10/3/05 16:35:08:736 EDT] 0000000a Symlink       >  isSymlink Entry
                                /web/usr/wbi/zWPS/V6R0
[10/3/05 16:35:08:737 EDT] 0000000a Symlink       3   absolute path
                                /web/usr/wbi/zWPS/V6R0
[10/3/05 16:35:08:737 EDT] 0000000a Symlink       3   canonical path
                                /web/usr/wbi/zWPS/V6R0
[10/3/05 16:35:08:738 EDT] 0000000a Symlink       <  isSymlink Exit
                                false
[10/3/05 16:35:08:738 EDT] 0000000a WPSInstaller  I   BBOO0222I: CWPIZ0046I:
Destination application server root directory is /WebSphere/V6R0M0/AppServer.
[10/3/05 16:35:08:744 EDT] 0000000a WPSInstaller  I   BBOO0222I: CWPIZ0247I:
Response file is /web/usr/wbi/zWPS/V6R0/zos.config/sample.rsp.
[10/3/05 16:35:08:764 EDT] 0000000a WPSInstaller  3   response property
                                profilePath=/WebSphere/V6R0M0/AppServer/profiles/default
[10/3/05 16:35:08:765 EDT] 0000000a WPSInstaller  3   response property
                                nodeName=SY1
```

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 is for planning purposes only. The information herein is subject to change before the products described become available.

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.

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

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries, or both: IBM, IBM (logo), AIX, CICS, Cloudscape, DB2, DB2 Connect, DB2 Universal Database, developerWorks, Domino, IMS, Informix, iSeries, Lotus, MQSeries, MVS, OS/390, Passport Advantage, pSeries, Rational, Redbooks, Tivoli, WebSphere, z/OS, zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

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

IBM Websphere Process Server for z/OS version 6.0.1

Printed in USA