

WebSphere Process Server for z/OS



Installing and Configuring WebSphere Process Server

Version 7.0.0

30 April 2010

This edition applies to version 7, release 0, modification 0 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, send an e-mail message to doc-comments@us.ibm.com. We look forward to hearing from you.

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

© **Copyright IBM Corporation 2006, 2010.**

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

Contents

Tables v

Installing WebSphere Process Server . . . 1

Preparing to install WebSphere Process Server	3
Overview of installation and configuration	4
Determining your skill needs	6
Stand-alone configuration	7
Network deployment configuration.	9
Using the zPMT tool	11
Installing WebSphere Application Server.	14
Loading the product code from the installation media on to z/OS	15
Using an IBM SystemPac or ServerPac	15
Using an IBM Custom-Build Product Delivery Option	16
Planning for product data sets	17
WebSphere Process Server file system directories	18
Installing WebSphere Process Server	19
Creating a configuration definition using zPMT	19
Running the installation jobs using zPMT generated JCL	21
Running the installation script from USS	22
Applying product maintenance.	23
Applying a service level or restoring to the previous accepted service level	24
About the upgrade process	24
Installing the JNDILookup Web Service application	28
Installing Message Service clients	29
Uninstalling	29
Uninstallation overview	29
Uninstalling WebSphere Process Server for z/OS	29
Uninstalling Business Process Choreographer	32
Troubleshooting the installation.	32
Message reference for WebSphere Process Server for z/OS installation and configuration	32
WebSphere Process Server log files	33

Installation information. 35

Differences between stand-alone and network deployment configurations	35
Installation media contents	35
Naming considerations for nodes, servers, hosts, and cells	38
WebSphere Process Server features	38
Product version and history information.	39
Product version information.	39
Response file values	41
Using the zPMT tool	51

Configuring WebSphere Process Server 55

Configuring WebSphere Process Server for z/OS	55
Designing the DB2 database objects	55
Creating a response file	58

Creating common configurations	63
Creating and configuring the databases	72
Choosing how to create your DB2 database.	72
Creating the messaging engine data stores	78
Granting table privileges to the JCA authentication alias user ID	79
Setting the correct schema name for the SIBs	80
Adding the DB2 libraries to the Servant and Adjunct JCL	80
Cleaning up the Derby JDBC resources	81
Verifying the installation with DB2	81
ConsolidateJAASAuthAliases.py script	82
Configuring the Business Process Management components	88
The deployment environment wizard versus configuring the components manually	88
Setting up deployment environments.	89
Configuring the Business Process Management components manually	117
Re-configuring a WebSphere Enterprise Service Bus with WebSphere Process Server functions	118
Configuring SCA support for a server or cluster	119
Considerations for Service Component Architecture support in servers and clusters	121
Configuring all REST services on the administrative console	122
Configuring REST services in a service provider	122
Configuring REST services for a server, cluster, or component	123
Configuring REST services using the command line.	124
Configuring Business Process Choreographer.	125
Configuring Business Space	125
Configuring Business Space using the Profile Management Tool	127
Configuring Business Space as part of the Deployment Environment Configuration wizard.	129
Configuring Business Space for network deployment environments	131
Setting up specific widgets to work in Business Space	159
Setting up security for Business Space	165
Commands (wsadmin scripting) for configuring Business Space	180
Configuring business rules and selectors	193
Configuring the business rule and selector audit log	193
Configuring business rule and selector auditing using commands	195
Considerations for installing the business rules manager	197
Configuring the relationship service.	202
DB2 setup required for the relationships function	203
Configuring extended messaging resources	204
Enabling the Extended Messaging Service	204

Configuring listener port extensions to handle late responses	205	Configuring WebSphere Process Server for Service Federation Management	227
Selecting extended messaging providers	206	Configuring the Service Connectivity Management connectivity server	227
Setting up the messaging server environment	212	Configuring the Service Connectivity Management connectivity provider	228
Configuring the JNDILookup Web Service.	212	Service Connectivity Management usage of Service Component Architecture modules	231
Configuring Common Event Infrastructure	213	Service Connectivity Management mapping to proxy gateways	231
Common Event Infrastructure components	213	Troubleshooting configuration	232
Configuring the Common Event Infrastructure using the administrative console	215	WebSphere Process Server errors	232
Deploying the Common Event Infrastructure application	217	Verification errors	241
Configuring event messaging	220	Message reference for WebSphere Process Server for z/OS installation and configuration.	245
Populating the event database.	223	Log files	246
Configuring WebSphere Business Integration Adapters	225		
Setting up administration of WebSphere Business Integration Adapters	225		

Tables

1.	Customization environments available using the zPMT tool.	12	6.	Stand-alone Database design restrictions for CEI component:	57
2.	Product version and history information links.	39	7.	JAAS authentication aliases	82
3.	Sample response files	41	8.	Deployment environment component relationships	100
4.	Alphabetic list of keywords in WebSphere Process Server for z/OS response files	42	9.	Monitoring	160
5.	Customization environments available using the zPMT tool.	52			

Installing WebSphere Process Server

This section contains information about preparing for and installing WebSphere® Process Server for z/OS®.

WebSphere Process Server documentation (in PDF format)

Related information



PDF documentation

WebSphere Process Server documentation (in PDF format)



Information roadmaps

Business Process Management information roadmaps on IBM developerWorks organize information about WebSphere Process Server, WebSphere ESB, and the other products in the portfolio.



IBM Education Assistant

Multimedia educational modules about WebSphere Process Server, provided by IBM Education Assistant.



Technotes

WebSphere Process Server Support > Install. Have questions about installing your WebSphere Process Server product? These resources can help lead you through your product installation and setup.



Overview

Overview tab, on product library Web page. Use this page to access announcements, data sheets, and other general library documents related to WebSphere ESB.

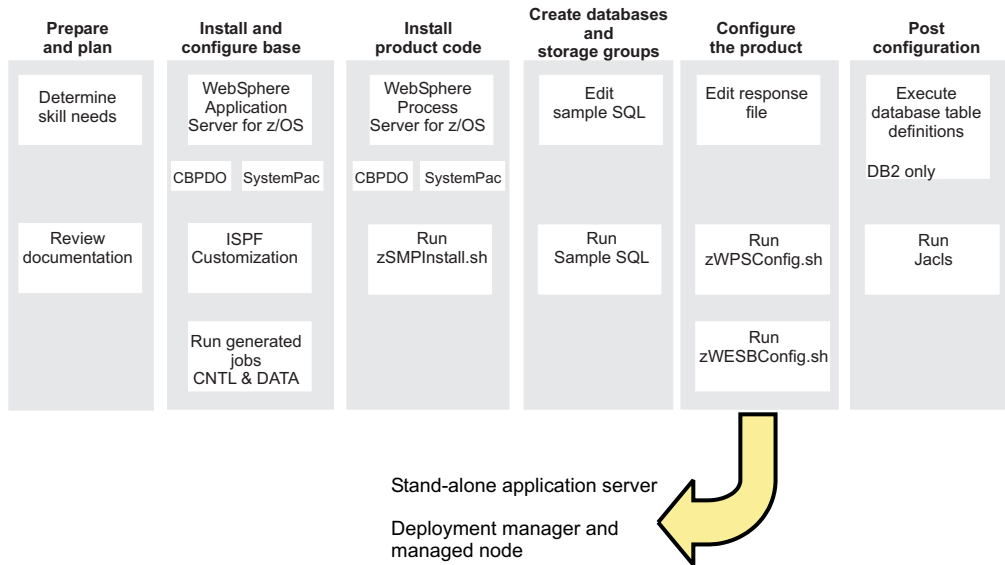
Task overview: installation and configuration

This article introduces the WebSphere Process Server for z/OS installation and configuration task flow for the supported configurations.

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 must complete the following steps:

1. Install the product binaries
2. Create WebSphere Process Server definitions
3. Augment your user profile as needed
4. Start your server.

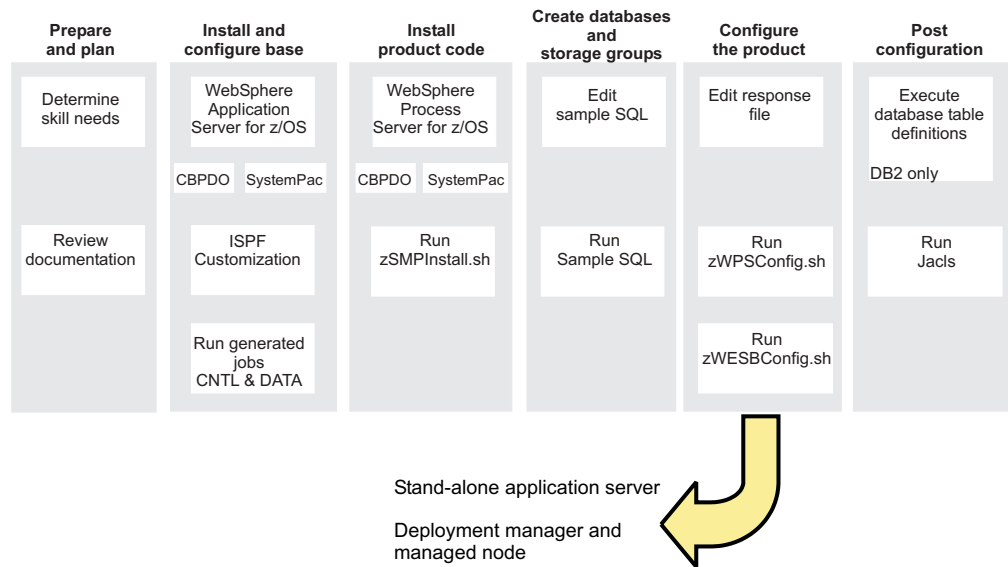
Task overview: installation and configuration

This article introduces the WebSphere Process Server for z/OS installation and configuration task flow for the supported configurations.

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 must complete the following steps:

1. Install the product binaries
2. Create WebSphere Process Server definitions
3. Augment your user profile as needed
4. Start your server.

Preparing to install WebSphere Process Server

A WebSphere Process Server for z/OS installation and configuration includes planning activities for WebSphere Process Server as well as WebSphere Application Server for z/OS.

About this task

The sections below list the tasks that you need to perform and provide references to the documentation resources that can help you install and configure WebSphere Process Server.

You must have made the following considerations to implement your plan:

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 <i>WebSphere Process Server 7.0.</i> WebSphere Process Server for z/OS installs on top of WebSphere Application Server for z/OS. For a list of system requirements see, Hardware and software requirements in the <i>WebSphere Application Server for z/OS Information Center.</i>
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 <i>WebSphere Application Server for z/OS Information Center.</i>

Task description	Information center resource
Implement Workload Management in goal mode on each z/OS system if necessary.	See Workload management (WLM) plan strategy in the <i>WebSphere Application Server for z/OS Information Center</i> .
Implement Resource Recovery Services (if not already implemented) on each z/OS system.	See Preparing Resource Recovery Services in the <i>WebSphere Application Server for z/OS Information Center</i> .
Plan for your performance and monitoring systems.	See Monitoring end user response time in the <i>WebSphere Application Server for z/OS Information Center</i> .
Plan and define your problem diagnosis procedures.	See Problem diagnostic plan strategy in the <i>WebSphere Application Server for z/OS Information Center</i> .
Consider automatic restart management before you install WebSphere Application Server for z/OS.	See Automatic restart management in the <i>WebSphere Application Server for z/OS Information Center</i> .
Perform planning tasks in preparation for loading the program materials from the installation media onto z/OS.	For information about planning tasks associated with unloading the WebSphere Application Server for z/OS installation media, see Planning for installation in the <i>WebSphere Application Server for z/OS Information Center</i> .

Depending on environment configuration variables and how you configured your response file, you may need to perform additional configuration tasks to complete the WebSphere Process Server for z/OS configuration.

Overview of installation and configuration

The WebSphere Process Server for z/OS installation and configuration is tightly integrated with, and dependent on, the installation and configuration of WebSphere Application Server for z/OS. The task is, therefore, a multiphase process that can span multiple roles.

In order to create a WebSphere Process Server server, the default profile that is created when WebSphere Application Server for z/OS is installed must be augmented into a WebSphere Process Server profile. This process adds WebSphere Process Server functionality to the existing WebSphere Application Server functionality.

Unlike many products that are installed on z/OS, you do not use WebSphere Customization Tool (WCT) to install and configure WebSphere Process Server. Instead, you run two shell scripts that perform tasks such as creating symbolic links between the product installation file system and the product configuration file system.

Installation

There are two main phases when installing WebSphere Process Server for z/OS:

Phase 1

The contents of the installation media are loaded on to the z/OS system.

This first phase is the responsibility of a system programmer who plans, maintains, and controls the use of the operating system to improve the overall productivity of an installation.

The result of completing the first phase of the installation is a read-only installation file system, which can be HFS (Hierarchical File Structure) or zFS (z-Series File System).

Phase 2

The installation script `zWPSInstall.sh` is run to create the required definitions that prepare the product for use.

This second phase is the responsibility of a product administrator.

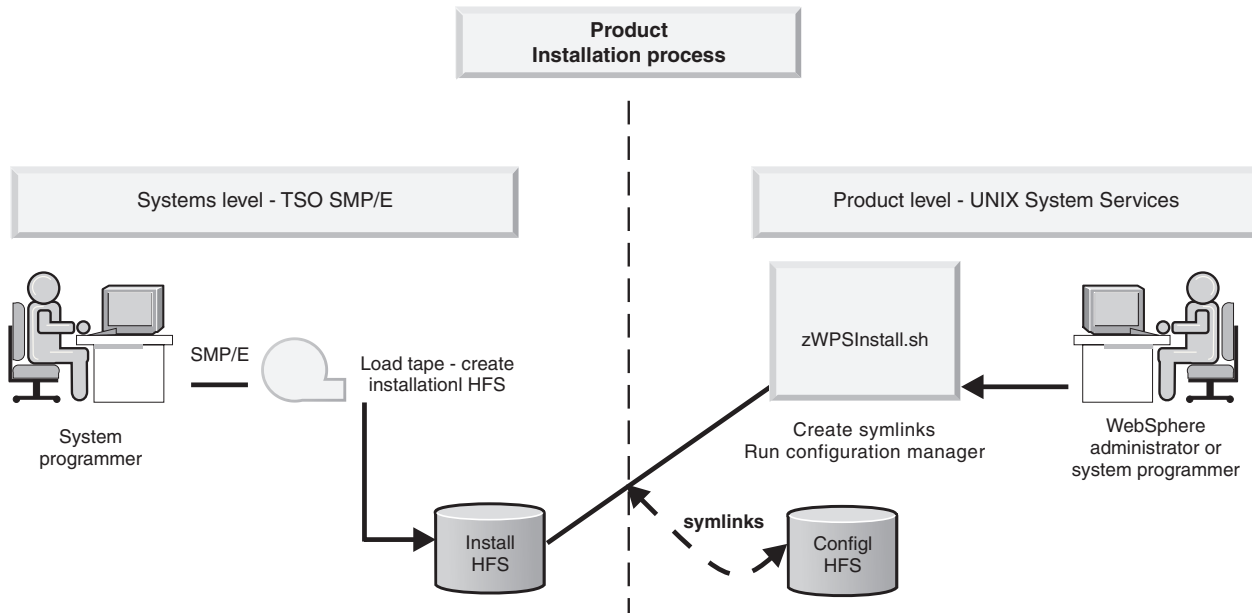
When this phase of the installation is complete, the following events have taken place:

- Symbolic links have been created from the WebSphere Application Server for z/OS configuration file system to the WebSphere Process Server for z/OS installation file system.
- Apache Ant script actions have been invoked that update the administrative console with the WebSphere Process Server for z/OS function.

The configuration file system is a writable file system that holds customized configuration documents and files for the configured product installation. The configuration file system contains the application server root directories (`/AppServer` and `/DeploymentManager`) and certain symbolic links that are associated with them. Each different node type under the configuration file system requires its own cataloged procedures under z/OS. By default the configuration file system is mounted at `/WebSphere/V7R0`.

The configuration file system has symbolic links to the WebSphere Process Server installation file system (by default, `/usr/lpp/zWPS/V7R0`). The symbolic links point to read-only files from the installation file system, such as JAR files and shell scripts.

The relationship between the people involved in the installation, the system programmer and the product administrator, and the installation phases is illustrated in the following diagram:



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

After completing the product installation, the WebSphere administrator can begin to configure the product for use.

Configuration

After completing both stages of the installation process, you can configure WebSphere Process Server for z/OS. Before configuring WebSphere Process Server, 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 “Creating the DB2 databases and storage groups using SPUFI, DSNTEP2, or DButility.sh” on page 75.

The product configuration script zWPSConfig.sh 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.

Determining your skill needs

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

In assembling your project team, you should consider the skills you need to implement WebSphere Application Server for z/OS. This article discusses the recommended skill set necessary to support the following configurations:

- Basic configurations
- Production environments

Documentation to support the z/OS skills described here can be found at this web site: z/OS Internet Library.

For basic configurations:

Below are the recommended skills necessary to support a basic configuration:

- z/OS UNIX[®] System Services and the hierarchical file system (HFS) to set up a functional HFS and UNIX[®] environment.
- eNetwork Communications Server (TCP/IP) or equivalent to configure connectivity for WebSphere Application Server for z/OS clients and servers.
- Resource recovery services (RRS) to implement resource recovery services and to support two-phase commit transactions.
- Security Server (RACF[®]), or the security product you use to authenticate WebSphere Application Server for z/OS clients and servers, and authorize access to resources.
- Secure Sockets Layer (SSL) to enable security if desired (recommended).
- SMP/E and JCL.
- System logger to set up log streams for RRS and the WebSphere Application Server for z/OS error log.
- Webserver to support HTTP clients if desired.
- Workload management (WLM).
- Java[™] and WebSphere Application Server tooling to support application development and deployment.

Depending on the needs of the applications you deploy, you might also need skills to configure the resource managers your applications require, such skills might include CICS[®], DB2[®], IMS[™], and MQ.

For production environments:

As you move your system toward a production environment, you must have the following system skills available:

- Application Response Measurement (ARM).
- System Automation, if you have it installed, or whichever automation you prefer to use.
- Sysplex, if you plan to use WebSphere Application Server for z/OS in a cell that spans systems.
- Sysplex Distributor (part of eNetwork Communications Server), if you plan to create a high availability environment.
- RMF[™] or other performance measurement systems.

Stand-alone configuration

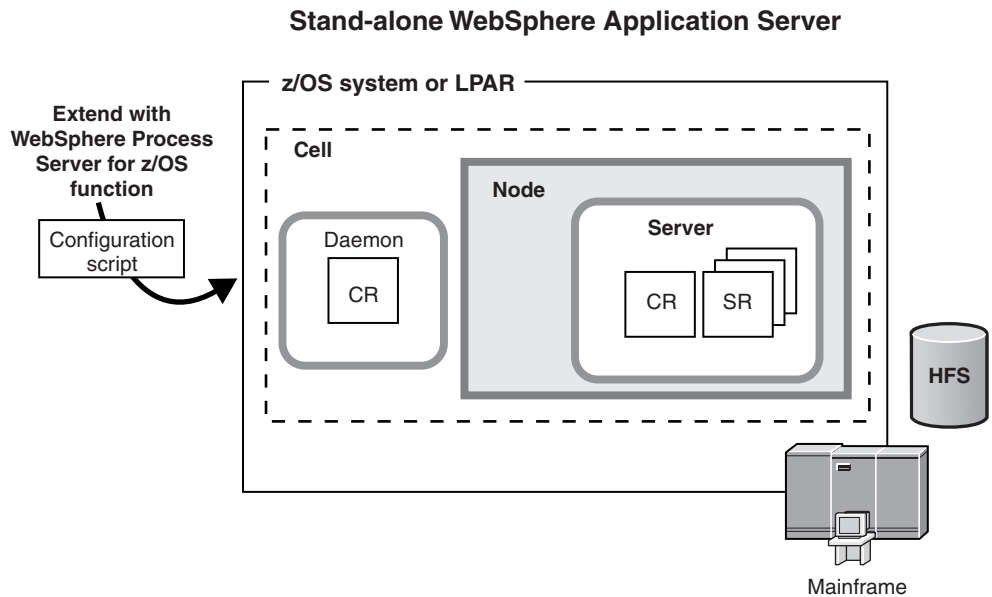
A stand-alone configuration, also known as a base configuration, is the simplest configuration you can use to deploy and run WebSphere Process Server for z/OS.

A stand-alone configuration of WebSphere Process Server for z/OS consists of a single node running an application server and one daemon server in a single z/OS[®] system or LPAR.

The application server runs the administrative console, which you can use to deploy and run additional applications. The application server is managed independently from other servers.

The daemon server is a unique server that runs constantly and has one controller region, which distributes server workload.

The following figure shows a stand-alone configuration which is made up of a single node running an application server and one daemon server in a single z/OS® system or LPAR:



The following figure shows a stand-alone configuration which is made up of a single node running an application server and one daemon server in a single z/OS® system or LPAR:

You must have the following configuration in place before you can create a stand-alone configuration:

- WebSphere Application Server for z/OS must be installed as the stand-alone server.
- Your UNIX user ID must have permission to access the UNIX shell to run the installation and augmentation scripts from inside the shell. Gaining permission to access the shell involves making modifications to your RACF® profile and creating a home directory in the UNIX shell. The home directory is where you begin a UNIX session, and where you store environment variable files that are required to run programs. You can also use the home directory as the main directory for storing data.
- The WebSphere Process Server for z/OS product code must be loaded onto the system from tape, so that you can use it to install and configure WebSphere Process Server for z/OS.

Advantages of a stand-alone configuration

Use a stand-alone configuration if you want to isolate the test and production systems in your company. Isolating test and production systems is important because, if the systems are not isolated, applications that are being tested in the test system might cause errors that can affect other applications on which your company depends.

You might choose to use a stand-alone configuration in the following scenarios:

- Your company is very large and you want to give each of your test groups an LPAR in which to run applications.
- Your company runs production and test environments on one zSeries® system. For example, you might give the test group access to one LPAR and give the production group access to all of the other LPARs.
- Your company is deciding whether to use WebSphere Process Server for z/OS and you want to provide a small amount of resource for feasibility testing.

The other type of configuration that you can use for WebSphere Process Server for z/OS is a network deployment configuration, which consists of multiple servers. A network deployment configuration is intended for a more sophisticated environment, and improves scalability and security throughout the system.

Network deployment configuration

An initial network deployment configuration consists of a deployment manager server that has a daemon for the z/OS® system on which the deployment manager runs. After a network deployment cell is created, you can add application server nodes by creating and federating new empty managed nodes, or by federating a stand-alone application server node into the network deployment cell.

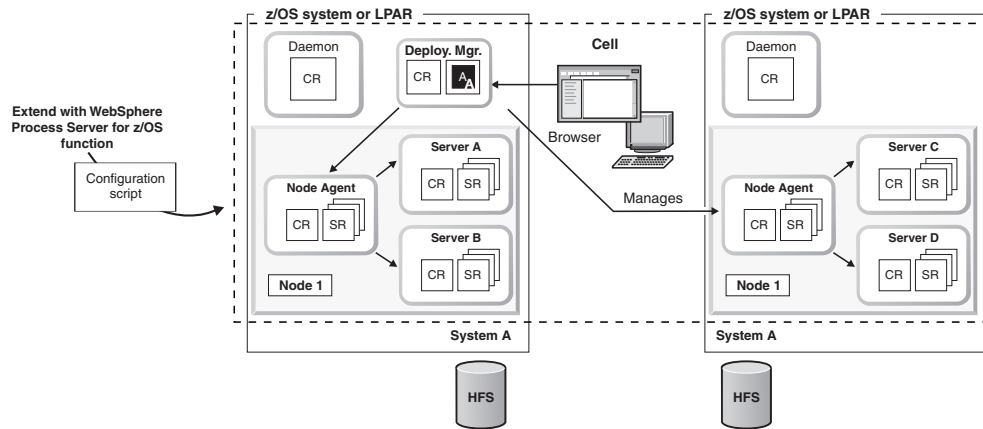
To install WebSphere Process Server into a network deployment environment, you must configure both a deployment manager node and an empty managed node before federation. Then to perform federation you run the job BBOWMNAN. When you federate an empty node to the deployment manager it becomes a managed node because it is being administered by the deployment manager. The managed node contains a node agent but no application servers. You can add an application server or cluster to the node with the administrative console.

The deployment manager runs the administrative console applications and is used for centralized administration tasks, such as managing the configuration of all of the managed nodes in its cell and deploying applications to selected servers and clusters in the cell. The deployment manager runs on one node, and application servers run in different nodes.

A basic network deployment configuration is made up of the following components:

- A deployment manager server running in a separate node that runs the administrative console application, which you can use to deploy applications.
- One or more application server nodes on each z/OS target system hosting portions of the cell. Each node consists of a node agent and some number of application servers. Each node must be federated into the deployment manager cell.
- A single location service daemon on each z/OS system. One daemon server must exist for each cell, which runs constantly, checking and distributing server workload.
- A DB2® database. DB2® Universal Database™ version 8.1 is the default database type for network deployment environments.

The following figure shows a network deployment configuration that consists of a deployment manager server that has a daemon for the z/OS system on which the deployment manager runs. The deployment manager is administering four servers, A,B,C, and D via two node agents:



You must have the following configured before you can create a network deployment topology:

- WebSphere Application Server for z/OS must be configured on the server with a deployment manager node and an empty managed node that has not been federated to the deployment manager. It is important that the empty managed node is not federated before running the installation script. You will federate the node after you have augmented it to contain WebSphere Process Server for z/OS configuration data.
- Your UNIX[®] user ID must have permission to access the UNIX shell because this is where you run the installation and augmentation scripts. Providing access involves making modifications to your RACF[®] (security) profile and creating a home directory within the UNIX shell. The home directory is where you begin a UNIX session, and where you store environmental variable files that are required to run programs. You can also use the home directory as the main directory for keeping your work data.
- The WebSphere Process Server for z/OS product code must have been loaded onto the system from tape, so you can use it to install and configure WebSphere Process Server for z/OS.

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

- When allocating target data sets it is possible, though not recommended, to use the same target data sets that you may have used for a stand-alone application server node. The job names for each configuration are very similar to one another, and if you use the same target data sets, you may find it difficult to keep the two sets of jobs separate. Therefore, create a new set of target data sets for the deployment manager node and keep the two sets of jobs separate from one another.
- Share the root of the file system across all processors apart from the configuration of the deployment manager in a file system on a generic system mount point.

It is very important that you plan your WebSphere Process Server for z/OS configuration before you start, especially when configuring a network deployment cell. There are many choices and you must understand the factors that influence these choices to make the correct decisions during the installation process.

Advantages of a network deployment configuration

- One of the main advantages of a network deployment configuration is availability. When the configuration contains multiple LPARs, you can reduce single points of failure and maintain availability during planned and unplanned outages.
- You can configure how messages are delivered. For example, you can specify secure assured delivery where messages are assured not to get lost and are transported securely, or best-effort where messages might get lost in case of a system failure.
- You can set up a network deployment cell to have several servers that host mediation modules. Mediation modules provide scalability (the ability to handle more client connections) and greater message throughput.
- You can create server clusters. With server clusters you can manage a group of servers together and enable those servers to participate in workload management.
- Your bus environment might be made up of several stand-alone and deployment manager profiles, to provide separate administrative domains for different departments or to separate test and production facilities. Each profile has its own SCA.SYSTEM service integration bus.

Related reference

zWPSInstall.sh script

Use the zWPSInstall.sh script to modify a WebSphere Application Server profile either to install or uninstall WebSphere Process Server. Run the script on each node in your configuration, including the deployment manager.

“Response file values” on page 41

When you have run the installation jobs which install the WebSphere Process Server for z/OS product definitions, sample response files are installed into the installation file system. Copy and edit these response files according to the configuration that you want to achieve, and pass in the response file when you run the augment script.

zWPSConfig.sh script

Use this script to configure and augment the WebSphere Process Server for z/OS installation. Run the script on each node in your configuration, including the deployment manager.

JCL for installing WebSphere Process Server into a deployment manager node

JCL for augmenting a deployment manager node

JCL for installing WebSphere Process Server into an empty node

JCL for augmenting an empty node

Using the zPMT tool

zPMT stands for z/OS Profile Management Tool. The zPMT tool is a workstation based tool that captures information and generates customized JCL batch jobs and configuration response files that you can use to create your WebSphere Process Server for z/OS system.

The essential features of the zPMT tool include its ability to perform the following tasks:

- Capturing information.
- Generating customized JCL batch jobs and configuration response files.

- Uploading the JCL batch jobs and configuration response files to the z/OS system.

The zPMT tool is part of WebSphere Customization Tools (WCT), an Eclipse based tool. To use the zPMT tool you have to install the WCT on your workstation and start it, then start the zPMT tool from there. For information about how to install WebSphere Customization Tools, see *Installing WebSphere Customization Tools*.

You can run the zPMT tool on your workstation and enter information in a series of windows. The information you provide is stored in the customization location of WCT which is a directory structure on your hard drive. The generated JCL jobs and a response file that create your WebSphere Process Server for z/OS servers are stored in this directory structure.

The JCL files are sent to a z/OS system using the process feature of the zPMT tool. The files are placed in two partitioned data sets named CNTL and DATA.

- The CNTL data set contains the generated JCL jobs you can use to install and configure WebSphere Process Server for z/OS.
- The DATA data set contains the configuration response file that is passed to the augmentation scripts.

You can use the zPMT tool to allocate the data sets as part of the upload.

The process of planning a cell is the same with the zPMT tool as it was with the ISPF panels. The zPMT tool prompts you for the same information as the ISPF panels, but in a few cases in a slightly different sequence. Naming conventions and port allocation charts are the same with the zPMT tool as they were with the ISPF panels.

Types of configuration that you can build with the zPMT tool

You can build three types of configuration with the zPMT tool. The following table cross-references the zPMT tool terms with the ISPF terms.

Table 1. Customization environments available using the zPMT tool

zPMT tool option	Equivalent ISPF option
WebSphere Process Server for z/OS Process Server	Stand alone Process Server node
WebSphere Process Server for z/OS deployment manager	Network deployment cell
WebSphere Process Server for z/OS managed (custom) node	Empty managed node

The zPMT tool creates a saved configuration definition in the workspace. The upload feature allows you to FTP the files to a z/OS system.

The variables you enter into the zPMT tool are saved as part of the workspace on your hard drive. That workspace is where the tool maintains all its information about your projects. The variables entered for one configuration are put into a zPMT tool response file which is a flat file with the variables and your values. This response file is similar in concept, but not identical in format, to the SAVECFG files for the ISPF dialogs.

The zPMT tool response file is different to the response file used to augment. Extra variables are used for customizing jobs, for example 4 variables for the JCL header to configure jobs. The response file used to augment is obtained when you click **process**, and is used to upload customized jobs to a z/OS system or local directory.

The name of this response file follows the naming format BPZRSPX where X can be:

- A = stand alone node.
- M = deployment manager node.
- N = managed node.

Installing WebSphere Customization Tools

You can create customized jobs and augment response files with WebSphere Customization Tools for tailoring your WebSphere Process Server installation in a graphical environment. You can also use WebSphere Customization Tools to share your configurations with others, make small updates to your augment response files quickly and easily, and upload your customized scripts to a target z/OS system.

About this task

After you install WebSphere Customization Tools you can use the zPMT tool to customize your installation of WebSphere Process Server.

Procedure

1. Go to the appropriate WebSphere Customization Tools Version 7.0.x Web site:
 - WebSphere Customization Tools Version 7.0 for Windows
 - WebSphere Customization Tools Version 7.0 for Linux
2. Under **Download package**, click **FTP** or **DD** to download the WCT installation file. Follow the instructions under the **Installation Instructions** heading to install WCT.
3. Navigate to the WebSphere Process Server installation file system root; if the default paths have been used this will be /usr/lpp/zWPS/V7R0. When you are at the installation file system root, navigate to util/WCT.
4. Download **zWBI.wct** to your workstation. This is a binary file download.
5. Start the WebSphere Customization Tools application on your workstation.
6. Click **Help > Software Updates > Install Extension**.
7. In the WebSphere Customization Tools Extension Locations panel, click **Install new extension location**.
8. In the Source Archive File panel, click **Browse**, navigate to the **zWBI.wct** file, then click **Open**.
9. In the Source Archive File panel, click **Next**. A summary of the installation details is displayed.
10. In the Extension Location Summary panel, click **Next**. It takes a few moments for the next panel to display.
11. In the Install Successful panel, read the instructions and make a note of the location displayed in the **Location** field.

Note: The text in the **Location** field is usually considerable in length and you must scroll right or enlarge the panel to read it all.

12. In the Install Successful panel, click **Finish**. The Product Configuration panel opens.

13. Click **Add an Extension Location**.
14. Browse to the location you noted in step 11.
15. Click **OK** . When you are prompted to restart the WebSphere Customization Tools application, click **Yes** and wait for the tool to restart.
16. Click **Welcome**.
17. Under **List of provided tools**, click **Profile Management Tool**. The following text is displayed: The tool also creates customization definitions for Websphere Process Server for z/OS. This text verifies that the plug-ins have installed successfully.

What to do next

You can now use the WebSphere Customization Tools application to configure WebSphere Process Server profiles. For further information about using WebSphere Customization Tools see Techdoc PRS3357.

Installing WebSphere Application Server

Before you can install WebSphere Process Server for z/OS, you must install and configure WebSphere Application Server for z/OS. If you have an existing WebSphere Application Server for z/OS network deployment cell, create a new empty WebSphere Application Server node on which to install WebSphere Process Server; after configuring the Deployment Manager to support WebSphere Process Server, you can then federate the node into the existing cell.

Before you begin

As a prerequisite for WebSphere Process Server for z/OS you must also install the following WebSphere Application Server feature packs as referred to in the WebSphere Process Server for z/OS program directory:

- The XML feature pack
- The SCA feature pack, including the optional SDO feature which is mandatory for WebSphere Process Server.

About this task

The procedure for installing WebSphere Application Server differs depending on the type of installation that you choose.

Stand alone server configuration

Creating a WebSphere Application Server stand alone server configuration provides you with the default profile that you can augment with WebSphere Process Server configuration data. See *Creating a stand-alone application server cell* in the WebSphere Application Server information center for more details. See the *WebSphere Application Server, Version 7.0 Information Center* for more information.

See *Building a practice WebSphere Application Server for z/OS cell* for information about creating a practice stand-alone configuration.

Network deployment server configuration

Creating a WebSphere Application Server deployment manager creates the default WebSphere Application Server profile that you can augment with WebSphere Process Server functions. See *Creating a Network Deployment cell* in the WebSphere Application Server information center for more information. See the *WebSphere Application Server, Version 7.0 Information Center* for more information.

What to do next

Next, install the WebSphere Process Server product code on top of WebSphere Application Server. See “Loading the product code from the installation media on to z/OS.”

Loading the product code from the installation media on to z/OS

The product code for WebSphere Process Server for z/OS is installed using either an IBM SystemPac[®] or ServerPac or an IBM Custom-Built Product Delivery Option (CBPDO).

About this task

In a z/OS environment, loading the product code from installation media on to the system is usually the responsibility of a system programmer.

- *Using an IBM SystemPac or ServerPac.* An IBM SystemPac or ServerPac consists of loadable product libraries and corresponding SMP/E data sets. The instructions for how to load the product code using a SystemPac or ServerPac are supplied with the SystemPac or ServerPac.
- *Using an IBM Custom-Built Product Delivery Offering.* A CBPDO contains SMP/E relative files and maintenance for one or more products. The instructions for how to load the product code using a CBPDO are supplied in the *WebSphere Process Server Program Directory*.

What to do next

When you have loaded the product code from the installation media and the product data sets are defined on the system, the system is ready for the product administrator to run the installation script `zWPSInstall.sh`.

Using an IBM SystemPac or ServerPac

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

About this task

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 installing a SystemPac or ServerPac.

For further information, see the following information sources:

- Web site at http://www.ibm.com/software/webserver/appserv/zos_os390/support
- PSP buckets at <http://www14.software.ibm.com/webapp/set2/psp/srchBroker>
- IBM Software Support Center Web site at <http://www-306.ibm.com/software/support/>.

When you install from an IBM SystemPac or ServerPac, you must ensure that the following requirements have been met:

Procedure

- 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” on page 17 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 or systems and the WebSphere Process Server installation file system is mounted at `/usr/lpp/zWPS/V7R0` or a similar mount point of your choice on each target system.

Using an IBM Custom-Build Product Delivery Option

An IBM Custom-Build Product Delivery Option (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.

About this task

If you use CBPDO, follow the instructions in the copy of *WebSphere Process Server for z/OS: Program Directory* (GI11-2880-00) which ships with your order.

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/>.

The Program Directory includes instructions on how to load the product code from the installation media on to your z/OS system. The Program Directory also provides instructions about how to set up the installation file system (using HFS or zFS) so that you can run the WebSphere Process Server installation script `zWPSInstall.sh`.

For further information, see the following information sources:

- WebSphere Application Server for z/OS server product support Web site at http://www.ibm.com/software/webserver/appserv/zos_os390/support
- PSP buckets at <http://www14.software.ibm.com/webapp/set2/psp/srchBroker>
- IBM Software Support Center web site at <http://www-306.ibm.com/software/support/>.

When you install from an IBM SystemPac or ServerPac, you must ensure that the following requirements have been met:

Procedure

- 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 and the WebSphere Process Server installation file system is mounted at /usr/lpp/zWPS/V7R0 or a similar mount point of your choice on each target system.

Planning for product data sets

WebSphere Process Server for z/OS product code resides in partitioned data sets (which contain the product data sets) and hierarchical file system directories (which contain the product directory and its subdirectories). The default high-level qualifier for the product data sets is BPZ.

Product data set contents

The WebSphere Process Server product data sets are divided into target data sets (used during product customization and execution) and distribution data sets (used to "back off" maintenance if necessary). In the following information, *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.

WebSphere Process Server has the following target data sets:

wps_hlq.**SBPZEXEC**
REXX execution scripts

wps_hlq.**SBPJCL**
JCL for installation jobs

WebSphere Process Server has the following distribution data sets:

wps_hlq.**ABPZANT**
installation file system files (ASCII)

wps_hlq.**ABPZEBCD**
installation file system files (EBCDIC)

wps_hlq.**ABPZEXEC**
REXX executables

wps_hlq.**ABPJCL**
JCL for installation jobs

See the WebSphere Process Server for z/OS Program Directory (GI11-2880-00) for allocation information about each target data set and distribution data set. Updates to this information are included in the Preventive Service Planning (PSP) bucket for each release of WebSphere Process Server for z/OS. You can find the Preventive Service Planning (PSP) bucket at <http://www14.software.ibm.com/webapp/set2/psp/srchBroker>.

Product data set naming convention

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.

Alternatively, 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.

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 production or test distinction; for example "BPZ.V6PROD.*" or "BPZ.V6TEST.*". Alternatively, the middle-level qualifier can include specific service level information; for example "WPS.W610FP1.*" or "WPS.W610FP2.*".

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

WebSphere Process Server file system directories

WebSphere Process Server for z/OS product and installation code resides in z/OS partitioned data sets (the product data sets) and z/OS USS file system directories (either HFS or zFS file systems).

WebSphere Process Server installation file system

All WebSphere Process Server for z/OS product files reside in the installation file system directory and its subdirectories. The installation file system for WebSphere Process Server is defined when you unload the product code from the installation media (Phase 1 of the installation process). The installation file system is typically mounted read-only.

The default location of the WebSphere Process Server installation file system is `/usr/lpp/zWPS/V7R0`.

The location of the WebSphere Process Server installation file system is different from the location of the WebSphere Application Server installation file system; the default location of the WebSphere Application Server installation file system is `/usr/lpp/zWebSphere/V7R0`.

The product directory and all of its subdirectories must reside in the same hierarchical file system (HFS) or zSeries file system (zFS) data set. The installation jobs and program directory assume that a data set is allocated to be used for WebSphere Process Server, separate from the z/OS root or version data set. The sample jobs to create either a zFS or HFS file system are supplied in hlq.SBPZJCL, members BPZALZFS and BPZALHFS. One of these two jobs is run during installation by your system programmer.

Configuration file system

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 Application Server configuration directories. These configuration directories contain symbolic links to files in the product directory.

The default location of the configuration directories is `/WebSphere/VR70`, so the location of the deployment manager configuration files might be `/WebSphere/VR70/DeploymentManager/config/bin`.

Related information

 [z/OS basic skills information center](#)

Installing WebSphere Process Server

The installation script creates the WebSphere Process Server for z/OS definitions that enable the product for use.

Related reference

 [zWPSInstall.sh script](#)

Use the `zWPSInstall.sh` script to modify a WebSphere Application Server profile either to install or uninstall WebSphere Process Server. Run the script on each node in your configuration, including the deployment manager.

 [JCL for installing WebSphere Process Server into a deployment manager node](#)

 [JCL for installing WebSphere Process Server into an empty node](#)

Creating a configuration definition using zPMT

You can use the Profile Management Tool to configure WebSphere Process Server. You access the Profile Management Tool by running the WebSphere Customization Tools application. This topic describes how to create a configuration definition using the zPMT tool. Alternatively, you can configure WebSphere Process Server by customizing sample response files that are shipped with the product.

Before you begin

- Install and configure WebSphere Application Server which is used as a base for your WebSphere Process Server for z/OS configuration.
- Install the WebSphere Customization Tools and perform any planning indicated for the customization task you have chosen. See [Installing WebSphere Customization Tools](#) for more information.

About this task

- See [zPMT tool](#) for conceptual information about the zPMT tool.
- See [“Response file values”](#) on page 41 for details of the response file keywords that you can use to configure WebSphere Process Server for z/OS.

Procedure

1. Launch the WebSphere Customization Tools. Click **Start > All Programs > IBM WebSphere > WebSphere Customization Tools V7.0 > WebSphere Customization Tools**. The welcome screen is displayed where you can select between **zPMT** and **zMMT**. Select **zPMT**.
 - The zPMT tool is opened as indicated by the **Profile Management Tool** button on the top toolbar. The zPMT tool contains two sections, customization locations and Customization Definitions:

- A customization location is a folder structure for all the sub-directories and files that the customization tools and the zPMT will create. The customization location folder can be a local or a network drive, and the operating system must recognize the drive and have access to it. You can create different locations for different configurations, and a workspace can contain multiple configurations.
 - A Customization Definition is a set of files on your workstation that you upload to z/OS to create the jobs that you run to configure WebSphere Process Server for z/OS.
2. Open an existing customization location if one exists, or create a new one. In both cases you must specify the file path.
 - You can add a previously created location to your working set, which is useful if you are sharing customization locations across installations of the WebSphere Customization Tools. In this situation a development department might send their configurations to test who will modify them. Adding a previously created location to your working set is also useful for backup purposes. For example, you could make a backup could be made before making any modifications to a configuration. You could also back up your configurations to a disaster recovery system in case the servers are down for any reason; having a backup greatly decreases the time needed to re-create the configuration.
 - The version number of a customization location refers to the base WebSphere Application Server to be targeted.
 - You must create customization locations with a version of 7.0 to augment for WebSphere Process Server for z/OS.
 3. Click **Augment**. The Environment Selection panel will display.
 4. Choose the environment that you want. You can select **Management, Application server, or Managed (custom) node**. Highlight your choice and click **Next**. The Augment Selection panel will display.
 5. Highlight the **WebSphere Process Server** option and click **Next**. After a short time, the Customization Definition Name panel will display.
 6. You can accept the default or enter a name for the zPMT Customization Definition. You can use more than 8 characters, but you cannot include space characters. A name is needed so that you can identify different definitions and keep them separate. The name is not used on the z/OS system, only by the zPMT tool. The Customization Definition name must be unique within your Customization Location.
 7. If you have an existing response file containing parameters and values that you would like to re-use to create this Customization Definition, optionally specify the path to that file in the **Response file path name** field. This field is optional. If an existing zPMT or configuration response file is specified here the zPMT reads the values it contains and uses them to populate the configuration panels that follow.
 8. Click **Next**. The Target Data Sets panel will display. Enter a high level qualifier for the target data set. This value will be used as a target data set to upload generated jobs, response files and instructions, as well as being used to customize the generated jobs with the target location of the JCL to run.
 9. Continue to progress through the customization panels, specifying the values for the parameters that will be used to create your WebSphere Process Server for z/OS system and pressing **Next** to go to the next panel.
 10. After you have completed the required panels, the Customization Summary panel will display. Click **Augment** to generate the customized jobs and finish the definition. The Customization Definition Summary will display with a

successful creation message. Click **Finish** to return to the Profile Management tool, where your new Customization Definition will be highlighted in the Customization Definitions view. The Customization Summary view now contains basic information about your highlighted definition, including its name, type and paths to the instruction file and response file. The Customization Instructions view contains the information that you require to run the jobs on your system to create your required type of configuration. Finally, the Customization Response File view displays the name-value pairs of parameters generated by WCT to describe your Customization Definition.

11. Upload the definitions to the z/OS system by selecting the Customization Definitions tab, highlighting the required Customization Definition, clicking **Process** on the Customization Definition window and selecting **Upload to target z/OS system**.
12. Enter the host, ID, and password used to access z/OS. The folder in which the files will be stored is disabled because this value must always be the same as the target data sets as specified in the configuration, so you are therefore unable to edit the location of this folder without regenerating the Customization Definition.
13. The target data sets might already exist on your z/OS system, or you can use the Profile Management Tool to allocate them.
14. The customized jobs and augmentation response file are uploaded to your z/OS system. The augmentation response file follows the naming format **BPZRSP x** where x can be:
 - **A** for a stand alone application server node
 - **D** for a deployment manager node
 - **N** for a managed node

You can either use the uploaded response file to configure the system or use the generated install and configuration jobs to create your WebSphere Process Server for z/OS configuration.

Results

You have now completed the customization of WebSphere Process Server for z/OS. You have created a zPMT response file using the zPMT tool, and uploaded the jobs and augmentation response file it generates to z/OS.

Running the installation jobs using zPMT generated JCL

The installation may be done by submitting the installation job which has been generated by zPMT or by running the zWPSInstall.sh comand. This topics describes running the installation job.

Before you begin

You must have created a configuration definition using the zPMT tool and processed it to transfer the generated JCL to the z/OS system.

About this task

Refer to the "customization Instructions" tab in the zPMT tool for the WebSphere Process Server profile which you have configured. These instructions will provide details of the sequence of actions and jobs to submit.

The installation job is called BPZINST and runs the install executable. Perform the pre-requisite actions and submit the job to run the installation. Refer to zWPSInstall.sh script for more information about the command which the installation job executes.

Once the installation job has run successfully, you may wish to move on to “Configuring WebSphere Process Server” on page 55 before running the augment job. The DbDesignGenerator.sh tool will now be available to design your DB2 database configuration and create a database design file. The database design file may be input into the augmentation process to provide WebSphere Process Server resource definitions with database configuration values. Refer to “Designing the DB2 database objects” on page 55 for more information.

Running the installation script from USS

Running the installation script, zWPSInstall.sh creates the code definitions needed to run WebSphere Process Server for z/OS on a WebSphere Application Server for z/OS server. If you are creating a network deployment configuration, you must run the installation script on each of the nodes that will belong to the network deployment cell. You can use JCL scripts to install the WebSphere Process Server product definitions or run the zWPSInstall.sh script manually. See JCL for installing WebSphere Process Server WebSphere Enterprise Service Bus into a stand-alone server, JCL for installing WebSphere Process Server into an empty node and JCL for installing WebSphere Process Server into a deployment manager node.

Before you begin

Before you install the server, you must perform the following steps:

1. Install and customize WebSphere Application Server for z/OS on the node. Ensure that the application server is stopped. For more information, see Starting and stopping the server.
2. Back up the file system on the server, so that you can restore it if necessary. For more information, see Backing up the WebSphere Application Server for z/OS system.
3. Load the product code from the installation media on to the z/OS system.
4. Ensure that you have administrator authority on the system so that you can run the installation script.
5. If you are running the installation script from TSO OMVS, ensure that the region size for the TSO user ID is large enough to run Java™ (typically 150M).
6. Optional: **Optional:** You might need to increase the OMVS time limit to allow the product configuration script time to complete. To increase the OMVS time limit so the session will not time out, enter the following system command:

```
SETOMVS MAXCPU=86400
```

For more information about the installation shell script, see zSMPInstall.sh script.

Procedure

1. In OMVS, switch to the administrator user ID. For example:

```
su wsadmin
```
2. Change to the directory in which you want to run the installation script: For example:

```
cd /usr/lpp/zWPS/V7R0/zos.config/bin
```
3. Add the current directory to your PATH so that you can run the installation script:

```
export PATH=.:$PATH
```

4. From the command prompt, run the installation script. For example:

```
zWPSInstall.sh -smproot /usr/lpp/zWPS/V7R0 -runtime  
/usr/lpp/zWebSphere/V7R0/AppServer -install
```

Optionally, you can redirect the standard out messages to a file instead of displaying it in the console but these instructions assume that you are viewing the standard out messages while the script runs.

If you are uncertain about the progress of the running script, refresh the console display and check the last few messages.

When the installation script has finished running, the following messages (or similar) are displayed before you are returned to the shell prompt:

```
CWPIZ0256I: set up configuration complete  
CWPIZ0257I: creating the symbolic links...  
CWPIZ0259I: creation of symbolic links complete  
CWPIZ0260I: doing post install file updates...  
CWPIZ0262I: post install updates complete  
CWPIZ0263I: running Configuration Manager update...  
Oct 19, 2007 4:28:16 PM com.ibm.ws390.installer.WPSInstaller  
INFO: BBZWI218  
Oct 19, 2007 4:38:16 PM com.ibm.ws390.installer.WPSInstaller  
WARNING: BBZWI221  
CWPIZ0264I: Configuration Manager update complete
```

When the script has finished running, review the messages that are displayed in the console, checking that there are no error messages displayed.

If the software did not install successfully, see [Troubleshooting the installation](#) for information about how to assess installation problems.

Note: If you are creating a network deployment configuration, do not run the job BBOWMNAN to federate the node into the deployment manager at this time. You will federate the node at a later stage, after completing the configuration steps.

Results

The WebSphere Application Server configuration now contains symbolic links to the WebSphere Process Server installation. The WebSphere Application Server administrative console is updated so that you can use it to administer WebSphere Process Server.

What to do next

After you have run the installation script successfully, you are ready to configure the server with WebSphere Process Server for z/OS functions. For more information, see [“Configuring WebSphere Process Server for z/OS”](#) on page 55.

Applying product maintenance

WebSphere Process Server for z/OS is installed and configured into a WebSphere Application Server. Consequently, you apply product maintenance to WebSphere Process Server for z/OS using the WebSphere Application Server product and techniques.

Before you begin

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.

About this task

Use the following procedure when you want to apply a new service release to your system.

Procedure

See *Applying product maintenance* in the WebSphere Application Server for z/OS information center for a description of how to apply product maintenance

What to do next

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

About this task

Service that is applied to the product data sets and product file system occasionally requires corresponding changes to be made to the configuration file system 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

About the upgrade process

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

You can upgrade the WebSphere Process Server for z/OS product using one of these methods:

- Overlaying an existing installation file system with a newer product version
- Running the upgrade script to update an older version to a newer installed product version

Scenario 1: Overlaying an existing installation file system with a newer version

In this scenario, you upgrade WebSphere Process Server for z/OS by using SMP/E to load the newer version of the product over the existing installation file system (Figure 1).

Note: A configuration has access only to its configuration file system. Symbolic links in the configuration file system provide access to the code in the installation file system.

Subsequently, the *applyPTE.sh* script runs when the server controller starts. The script checks the level of the configuration file system against the level of the installation file system. If the two file systems are at the same maintenance level, the server starts. If the configuration file system is at a lower level than the installation file system, the *applyPTE.sh* script makes changes to the configuration file system specified by the maintenance level of the installation file system and starts the server.

Note: If the configuration file system is at a higher level than the installation file system, for example when maintenance is backed off, the server cannot start.

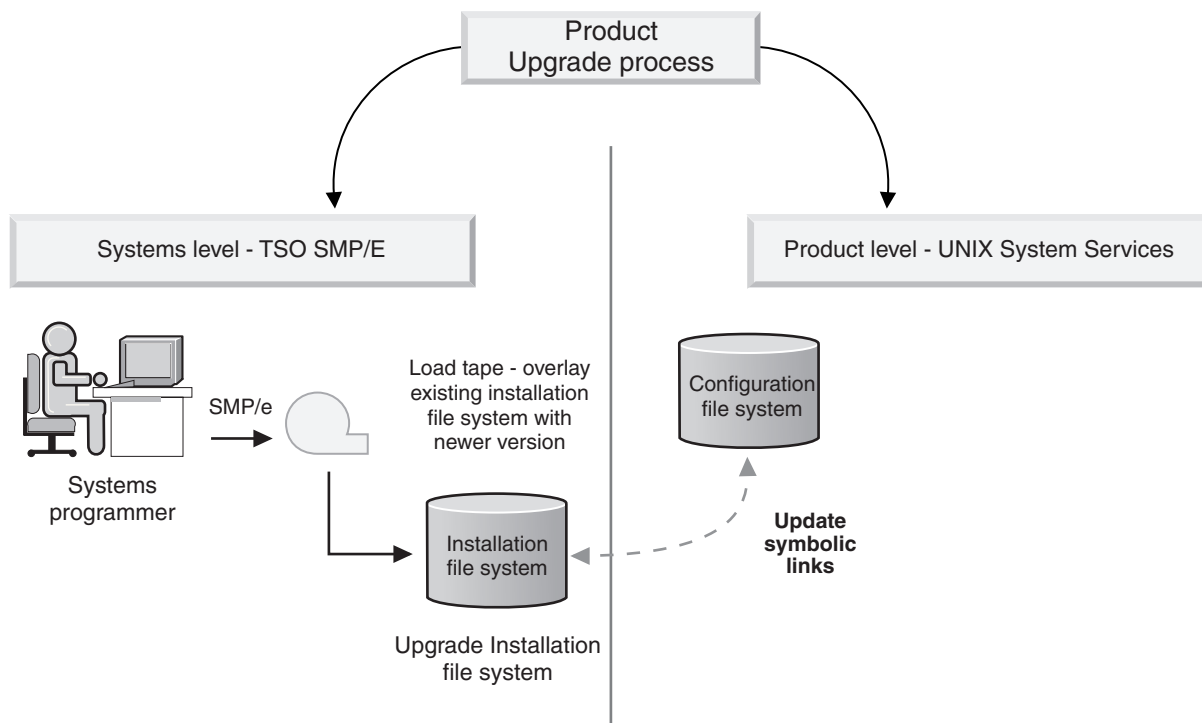


Figure 1. Upgrading WebSphere Process Server for z/OS product by overlaying the existing Installation file system

Scenario 2: Using the upgrade script

In this scenario, you upgrade WebSphere Process Server for z/OS by using SMP/E to load the newer version of the product separate from the existing installation file system (Figure 2).

You can then run the upgrade script for each application server that you want to upgrade. The upgrade process changes the service level symbolic link from the existing installation file system to the newer installation file system for the application server.

Subsequently, the *applyPTE.sh* script is run when the server controller starts. The script checks the level of the configuration file system against the level of the

installation file system. In this case, the configuration file system is at a *lower* level than the installation file system, and the *applyPTF.sh* script performs the changes to the configuration file system specified by the maintenance level of the installation file system and starts the server.

In Figure 2, the newer version of the WebSphere Process Server for z/OS has been installed and configured to run on application server A. To upgrade application server B, the systems programmer can run the upgrade script. The script updates the application server B Configuration file system by pointing its service level symbolic link to the new installation file system associated with application server A. The installation upgrade is completed when the *applyPTF.sh* script is run.

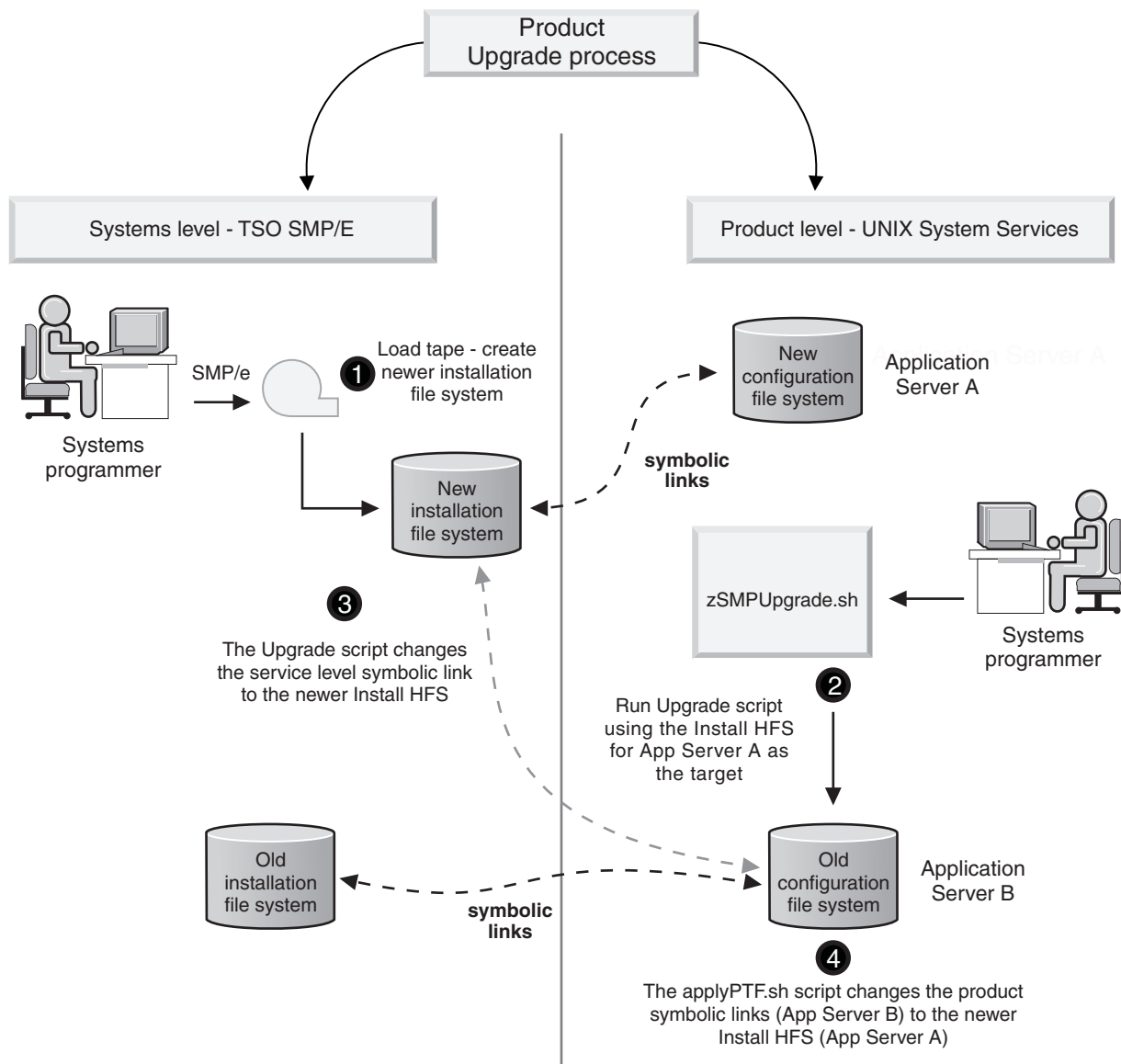


Figure 2. Upgrading the WebSphere Process Server for z/OS product using the upgrade script

Using the upgrade script with intermediate symbolic links

This example is similar to scenario in Figure 2, except that rather than a direct symbolic link between the installation file system and configuration file system, an intermediate symbolic link is used. The intermediate symbolic link is a standard symbolic link that points to the installation file system and the configuration file system points to the intermediate symbolic link. Changing a node to another service level involves changing the single intermediate symbolic link.

You can run the upgrade script for each application server that you want to upgrade. The script uses the newer version of the installed file system to update the symbolic links for the configuration file system that you want to update. The configuration file system points to the intermediate symbolic link, which is an additional layer of indirection.

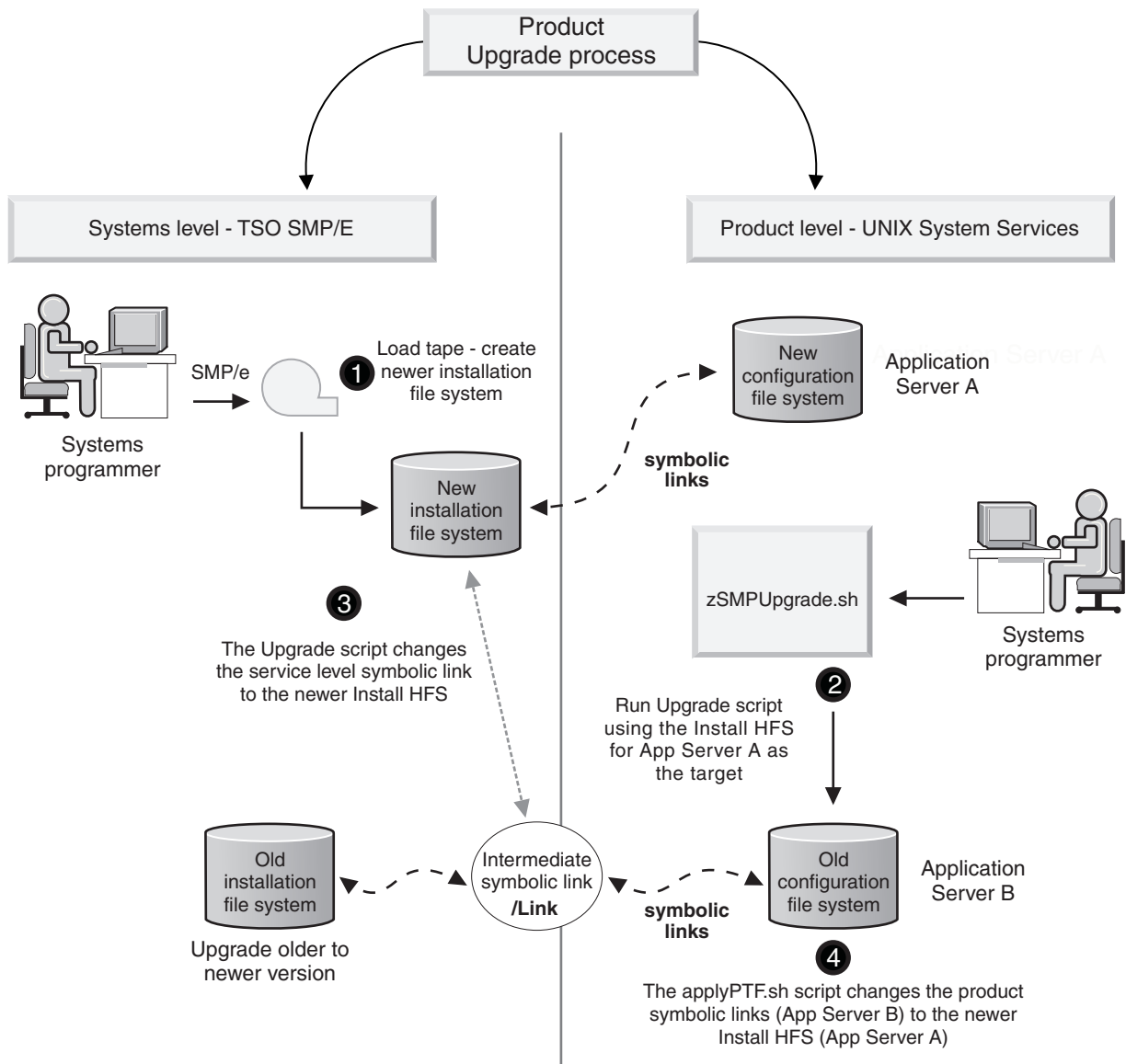


Figure 3. Upgrading the WebSphere Process Server for z/OS product using the upgrade script

Installing the JNDILookup Web Service application

WebSphere Process Server maintains administered JMS objects which cannot be interpreted by non-Java clients. To allow non-Java clients to access administered objects, WebSphere Process Server provides a JNDILookup Web Service. This Web service acts as a proxy to allow non-Java clients to retrieve JMS administered objects.

Before you begin

Before installing the JNDILookup Web Service application you must ensure you have a running installation of WebSphere Process Server on your system.

About this task

If your WebSphere Process Server installation is going to be accessed by non-Java clients, you need to install the JNDILookup Web Service. This application can be installed using the administrative console as described below.

Important: After you start performing the steps below, click **Cancel** to exit if you decide not to install the application. Do not simply move to another administrative console page without first clicking **Cancel** on an application installation page.

Procedure

1. Click **Applications** → **New Application** in the console navigation tree.
The first of two Preparing for application installation pages is displayed.
2. On the first Preparing for the application installation page, specify the path to the new application.
 - a. Browse to the *install_root/installableApps* directory, and select *SIBXJndiLookupEAR.ear*.
 - b. Click **Next**.
3. On the second Preparing for application installation page:
 - a. Select whether to generate default bindings and mappings.
Using the default bindings causes any incomplete bindings in the application to be completed with default values. Existing bindings are not altered. You can customize default values used in generating default bindings.
 - b. Click **Next**.
The Install New Application pages are displayed.
4. On Step 1: Select installation options panel, select **Deploy Web services**.
5. Click **Step 5: Summary** to go to the Summary panel.
6. On the Summary panel click **Finish**.

What to do next

Examine the application installation progress messages. If the application installs successfully, save your changes to the Master Configuration. You can now see **SIBXJndiLookup** in the list of deployed applications on the Enterprise Applications page accessed by clicking **Applications** → **Enterprise Applications** in the console navigation tree.

To start the application from the Enterprise Applications page, select **SIBXJndiLookup** and click **Start**.

Installing Message Service clients

If you want to enable C, C++, or .NET applications to participate in interactions with WebSphere Process Server, you can use the Message Service clients.

About this task

The steps that you need to complete to install message service clients depends on the type of client and the type of installation you choose to use. The steps are described in the documentation for the type of client.

Procedure

- Installing Message Service Client for .NET
- Installing Message Service Client for C/C++

Uninstalling

The Uninstalling section describes how to uninstall WebSphere Process Server for z/OS.

Uninstallation overview

You can uninstall WebSphere Process Server for z/OS by running the install script `zWPSInstall.sh` with the `uninstall` parameter from a command line.

Review the latest technote information before running the uninstall script.

Running the command `zWPSInstall.sh` with the `-uninstall` command argument restores the WebSphere environment back to the level it was at before installation.

Note: You must uninstall the Common Event Infrastructure and Business Process Choreographer components separately. Consult the appropriate help topics for information.

The uninstall process results in the following actions:

- WebSphere Process Server features are disabled by running Configuration Manager scripted actions. All administrative console plug-in extensions are removed.
- `WSPROFILE` scripted actions reverse the augmentation process on the default WebSphere Application Server profile.
- The post install file is deleted and code base permissions are removed.

Uninstalling WebSphere Process Server for z/OS

Uninstall WebSphere Process Server for z/OS by running the shell script `zWPSInstall.sh` with the keyword `-uninstall`. These instructions describe how to run the install script with the `uninstall` keyword from a `OS/390`® UNIX command shell. Alternatively you can run the script and keyword from a prompt using telnet. These instructions assume you are using TSO, if you are using telnet use the equivalent commands. These instructions also assume that you have the default key mappings using the CTRL key to enter input. If you have changed this configuration, use whichever key you have mapped to enter information into TSO.

Before you begin

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

Read the latest technote information on uninstalling WebSphere Process Server for z/OS before running the uninstall script.

The WebSphere Process Server for z/OS install process assumes that you have a working knowledge of z/OS UNIX System Services. Refer to the following documentation if you need more information:

- z/OS V1.11 UNIX System Services User's Guide
- z/OS V1.11 UNIX System Services Command Reference

Procedure

1. Access the OS/390® UNIX® command shell. telnet into USS directly, or from TSO enter the TSO command OMVS at the ISPF Command Option 6 or TSO OMVS from any other ISPF panel. Once you are in the UNIX shell, a command prompt (usually a dollar (\$) or pound (#) sign) indicates that the system is ready to accept input.

2. Switch to the administrator user ID:

```
su wsadmin
```

3. Change to zos.config/bin in the product installation file system directory:

```
cd /usr/zWPS/V7R0/zos.config/bin
```

4. Add the current directory to the path:

```
export PATH=.:$PATH
```

5. From the command prompt, run the installation script with the uninstall command, for example:

```
zWPSInstall.sh -smproot /wps/pathofWPScode/zWPS/V7R0 -runtime  
/WebSphere/V7R0/AppServer -uninstall -response  
/yourdrivename/yourfoldername/responsefilename.rsp
```

Type in the absolute path name of the response file according to your system configuration. The path file names above assume the uninstaller is using the default response files. If the response file was customized, the path name must represent the absolute path of the customized file.

6. Wait for the 'Configuration Manager for uninstall complete' message, and for the command prompt to return.

Results

Running the install script with the -uninstall parameter results in the following actions:

- WebSphere Process Server for z/OS product features are disabled when the configuration manager runs its scripted actions. The administrative console plug-in extensions are removed when the WebSphere Process Server for z/OS product features are disabled.
- The WSPROFILE script reverses the augment process on the default WebSphere Application Server profile which removes all WebSphere Process Server functionality. You are warned that the 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 you can no longer use other augmented profiles.
- The post install file is deleted.
- Code base permissions are removed.

If any shared common components are being used by other applications the command line prompting warns that uninstalling the product may cause other applications to no longer function correctly.

If the uninstall command was not successful check the associated log file and trace files:

- **Standard out messages**

Standard output messages display directly on the screen. 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/V7R0/AppServer
WAS_HOME is: /WebSphere/V7R0/AppServer
WBI_HOME is: /WebSphere/V7R0/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 `zWPSInstall.log` file in the run-time directory. The default location of this file is `/WebSphere/V7R0/AppServer/logs/wbi/zWPSInstall.log`.

- **Trace file**

Review the `zWPSInstall.trace` (ASCII) file in the run-time directory. The default location for this file is `/WebSphere/V7R0/AppServer/logs/wbi/zWPSInstall.trace`. There must be no error messages that have an "E" suffix..

You can also carry out the following troubleshooting actions:

- **Review the actions of the Update Configuration Manager task.** The actions are written to the log file `cmtInstall.log`, which is in ASCII format. The default location for this file is `/WebSphere/V7R0/AppServer/logs/wbi`. Search in this log for the text `>SEVERE<` or `>WARNING<` level messages to determine which errors have occurred. Each Apache Ant script that runs from the install directory has a log that it writes to, which is in ASCII format.

The default name of the directory that contains the Ant scripts is `/WebSphere/V7R0/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/V7R0/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 must contain a 'build successful' message when the uninstall has been successful.

- **Review the contents of the Unaugment Log.** The unaugment profile task records the actions it has made by writing to a log file (ASCII). The log file name is `wasprofile_unaugment_default.log`. The standard location for this file is `/WebSphere/V7R0/AppServer/logs/wasprofile`. Search this WebSphere Application Server profile augment log for `>SEVERE<` or `>WARNING<` level messages to determine overall error in processing. There should be no SEVERE messages.

What to do next

After you have removed WebSphere Process Server for z/OS from your system you must remove the Business Process Choreographer configuration.

Uninstalling Business Process Choreographer

For information on how to remove Business Process Choreographer from a WebSphere Process Server installation, go to the WebSphere Process Server for z/OS, version 7.0, information center and review the topics under **Installing WebSphere Process Server > Uninstalling the software > Removing the Business Process Choreographer configuration**. You can also find this information in the *Business Process Choreographer PDF*.

Troubleshooting the installation

You can diagnose problems when the installation of WebSphere Process Server is unsuccessful.

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 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

For a listing of the WebSphere Process Server for z/OS installation error messages, see `CWPIZ` in the Messages portion of the Reference documentation.

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/V7R0/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/V7R0/AppServer/logs/wbi/zWESBConfig.log and /WebSphere/V7R0/AppServer/logs/wbi/zWPSConfig.log respectively.

Related reference

“Failure in loading T2 native library db2jcc2zos” on page 232

This error can occur when using the DB2 Universal Driver connector, and WebSphere Application Server cannot load some external DB2 modules from SDSNLOAD or SDSNLOAD2.

“DataSource has a null RelationalResourceAdapter property” on page 234

The error shown in the example below is caused by a redundant datasource that is left behind after running the augment script zWPSConfig.sh. You can safely delete this datasource using the WebSphere administrative console. Be careful not to delete the JDBC provider that has a very similar name.

“SQLCODE = -471” on page 235

This error can occur when the Universal Driver has not been properly configured in a DB2 system.

“SQL code -204 and -516” on page 236

This error can be caused if the currentSchema property does not match the schema name of the tables and indexes that you created. The error messages show the JCA authentication alias that is being used.

“Repeated SIB messages about acquiring and losing locks” on page 237

This error can occur after correcting the DB2 Universal Driver configuration and restarting the server. The error messages are repeated continuously in the adjunct region.

WebSphere Process Server log files

There are two distinct groups of log files in the installed product. Logs detailing the product installation, product updates, and profile management are one group. Logs detailing the characteristics and runtime activities of individual profiles make up the second group.

Various log files are created during the installation and uninstallation of WebSphere Process Server and during profile creation, augmentation, and deletion. Examine these logs when problems occur during the product installation and configuration process. The log files and their locations within the product installation are detailed in the "Installation and profile creation log files" topic.

There are also a number of log files that are created for each profile. Some of these logs describe the parameters used for the creation of the profile. These types of log files generally remain unchanged when the profile is fully configured. Other profile-specific logs are continually updated to capture error, warning, and information messages emitted during runtime. Some of these log files are also used to capture a Common Base Event (that may include business object data) that is selected for monitoring. This set of logs is described in "Profile-specific log files" topic.

Installation information

This reference section contains subtasks and supporting conceptual and reference information related to installing WebSphere Process Server.

Differences between stand-alone and network deployment configurations

A stand-alone application server cell, also known as a base application server node, is the simplest configuration you can use to deploy and run WebSphere Process Server for z/OS applications. A stand-alone configuration provides a framework for a quick start or development environment and if you are configuring for a small environment you may find that a single server topology is all you need to meet your requirements. However, you are probably more likely to use a network deployment configuration because it can administer application servers that may be dispersed among multiple sysplexes in a network environment. A network deployment configuration is intended to support multiple application servers which can be clustered to provide high availability and reliability.

The main differences between a stand-alone and a Network Deployment configuration are:

Stand-alone configuration	Network deployment configuration
<p>A minimum of four address spaces are allocated for the following processes:</p> <ul style="list-style-type: none">• Location service daemon• Controller• Servant• Control region adjunct <p>The maximum amount of address spaces is limited only by resources</p>	<p>A minimum of seven address spaces are allocated for the following processes:</p> <ul style="list-style-type: none">• Location service daemon• Application server controller• Application server servant• Application server control region adjunct• Deployment manager controller• Deployment manager servant• Node agent <p>The maximum amount of address spaces is limited only by resources</p>
<p>Each server node is in a separate administrative domain.</p>	<p>All nodes are in the same administrative domain.</p>
<p>You can start and stop servers independently. Each server has an independent, unshared JNDI namespace.</p>	<p>You can start and stop servers independently, however the JNDI namespace is shared among all servers in the cell.</p>
<p>Multiple servants are allowed.</p>	<p>Multiple servants are allowed.</p>
<p>Clustering is not available .</p>	<p>Clustering is available.</p>

Installation media contents

The WebSphere Process Server for z/OS components that are to be installed on z/OS systems are supplied on a single tape or as a download package; components that are to be installed on non-z/OS systems are supplied on CDs.

Software supplied with WebSphere Process Server for z/OS

The WebSphere Process Server for z/OS package contains all the software that you need to install and configure WebSphere Process Server for z/OS, and to assemble and deploy applications.

The tape, or download package, includes WebSphere Application Server for z/OS, which needs to be installed and configured before installing WebSphere Process Server for z/OS. The following table lists the software that is supplied on the tape or in the download package.

Software	Description
WebSphere Process Server for z/OS	Provides the main WebSphere Process Server for z/OS software.
WebSphere Application Server for z/OS	Provides the WebSphere Application Server for z/OS software which you must install and configure before you install the WebSphere Process Server for z/OS software.
WebSphere Application Server Application Clients	Provides a stand-alone client run-time environment for your client applications so that you do not need to install a full instance of WebSphere Application Server for z/OS. The application client module is a Java Archive (JAR) file that contains a client for accessing a Java application.
DataDirect Java ^(TM) Database Connectivity (JDBC) drivers	Provides the two JDBC drivers that are produced by DataDirect Technologies for enabling connectivity to Microsoft ^(R) SQL Server. These drivers are the SequeLink and Connect JDBC drivers.

The CDs contain optional supplemental software that provides tool support for your production and development environments. The following table lists the software that is supplied on the CDs.

Software	Description
WebSphere Process Server Clients	<p>Provides the WebSphere Application Server for z/OS configuration with SOA Core and Business Process Choreographer functionality without needing a full WebSphere Process Server installation.</p> <ul style="list-style-type: none"> • WebSphere Process Server 7.0 Client for Windows® • WebSphere Process Server 7.0 Client for AIX® • WebSphere Process Server 7.0 Client for Solaris on SPARC • WebSphere Process Server 7.0 Client for Solaris on x86 64-bit • WebSphere Process Server 7.0 Client for HP-UX • WebSphere Process Server 7.0 Client for HP-UX on Itanium® 64-bit • WebSphere Process Server 7.0 Client for Linux® on x86 32-bit • WebSphere Process Server 7.0 Client for Linux on x86 64-bit • WebSphere Process Server 7.0 Client for Linux on POWER® • WebSphere Process Server 7.0 Client for Linux on System z® 31-bit • WebSphere Process Server 7.0 Client for Linux on System z 64-bit
IBM Message Service Clients	<p>Provides messaging and Web services capabilities in non-Java environments. Extends interaction between applications and WebSphere Process Server by using the provided clients:</p> <ul style="list-style-type: none"> • 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.
WebSphere Application Server Edge Components	<p>Provides load balancing, caching, and centralized security. See the WebSphere Application Server Network Deployment Edge Components Web page for more information.</p>
WebSphere Application Server Network Deployment V7.0 Supplements	<p>Provides WebSphere Application Server Network Deployment functions, licensed for use with WebSphere Process Server.</p>
WebSphere Application Server interim fix 6.2.0.13-WS-WAS-IFPK56164	<p>Interim fix for WebSphere Application Server Network Deployment 6.2.</p>
IBM Tivoli® Access Manager Servers	<p>Provides authentication and authorization APIs and integration to help you to secure access to business-critical applications and data that is spread across the extended enterprise. See IBM Tivoli Access Manager for e-business for more information.</p>

Software	Description
Rational® Agent Controller 6.2.5 for Windows	The Rational Agent Controller provides a technology for problem determination. The Rational Agent Controller is an enhancement to the Agent Controller from the Autonomic Computing Toolkit. Install the Rational Agent Controller on one of the supported z/OS system environments. Another z/OS system (perhaps an administrator workstation) can use the Log and Trace Analyzer to establish communication with the agent on the remote z/OS system. The Analyzer can request a specific log file from the Rational Agent Controller. The agent uses the appropriate log file parser to normalize the native log to a Common Base Event format and transfers the log file to the Log and Trace Analyzer. Create log file parsers for the Rational Agent Controller with the Common Base Event model builder, which is included in the Autonomic Computing Toolkit.

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.

Naming considerations for nodes, servers, hosts, and cells

There are certain considerations that you must take into account when customizing the WebSphere® Application Server installation. See Setting the customization variables: Stand-alone application server cell for more information.

WebSphere Process Server features

This topic describes the WebSphere Process Server features available for installation in the Installation Manager.

Sample Applications

Selecting the WebSphere Process Server **Sample Application** feature in Installation Manager determines whether the sample applications for both WebSphere Process Server and WebSphere Application Server are included in your installation. Sample applications include both source code files and integrated enterprise applications that demonstrate some of the latest Java Platform, Enterprise Edition (Java EE) and WebSphere technologies.

For more information about sample applications, see Installing and accessing the Samples Gallery.

For better performance in a production environment, do not install the Sample Applications.

WebSphere Process Server - Client

Selecting **WebSphere Process Server - Client** on the features panel installs the WebSphere Process Server Client and WebSphere Process Server. To install just the WebSphere Process Server Client, clear the check box for WebSphere Process Server.

Stand-alone WebSphere Process Server or WebSphere Enterprise Service Bus development profile

The Installation Manager includes an optional feature to create stand-alone development profiles for both WebSphere Process Server and WebSphere Enterprise Service Bus. These profiles will not work in a production environment. They are intended for users to gain familiarity with WebSphere Process Server or WebSphere Enterprise Service Bus without having to create working production profiles. Creating these profiles requires you to supply your administrator security ID and password credentials.

Product version and history information

Information and links to product version and history information.

The WBI.product file in the properties/version directory contains information such as product, version, build date, and build level. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE product SYSTEM "product.dtd">
<product name="IBM WebSphere Process Server">
<id>WBI</id>
<version>7.0.0.0</version>
<build-info date="8/31/09" level="of0935.02"/>
</product>
```

Click the following links for appropriate product version and history information:

Table 2. Product version and history information links.

Links
Product version information
genVersionReport command
versionInfo command
historyInfo command
genHistoryReport command

Product version information

The properties/version directory in the WebSphere Process Server for z/OS installation file system 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 historyInfo command 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 configuration_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 root of the installation file system, 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

The existence of this file 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

The existence of this file 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="IBM WebSphere Application Server - ND">
<id>ND</id>
<version>6.0.0</version>
<build-info date="02/03/05" level="s0461.18"/>
</product>
```

WBI.product

The existence of this file indicates the particular WebSphere Process Server for z/OS 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

An example of the file follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE product SYSTEM "product.dtd">
<product name="IBM WebSphere Process Server">
<id>WBI</id>
<version>6.2.0.0</version>
<build-info date="11/15/08" level="o0845.22"/>
</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.

Response file values

When you have run the installation jobs which install the WebSphere Process Server for z/OS product definitions, sample response files are installed into the installation file system. Copy and edit these response files according to the configuration that you want to achieve, and pass in the response file when you run the augment script.

Purpose

Response files contain keywords that you can use to configure WebSphere Process Server. Take a copy of the response file and make the file writeable before you start tailoring it to match the needs of the configuration you are creating. Once you have finished, save your changes.

Use the following rules and guidelines when you are working with response files:

- Do not leave any extra unreadable characters at the end of the keywords.
- Use TSO oedit or vi to edit the response file.
- Start comments on a new line. Do not start the comments to the right of the keyword.
- Start comments with a number sign (#) in column 1.
- The response file must be in EBCDIC format. If you prefer to edit the file using your workstation, use the text format during the FTP download and upload.

There are global variables at the beginning of the response file that are referenced by other variables in the file with the dollar sign (\$). The serverName variable is an example of a variable that references a global variable.

The following table lists the names of the four sample response files that you can use to create different WebSphere Process Server for z/OS configurations.

Table 3. Sample response files

standAloneProfile.rsp	Use this file to create a stand-alone configuration using a Derby™ database. A stand-alone configuration has a single node running an application server and one daemon server in a single z/OS® system or LPAR. Use a Derby database only for a test system.
standAloneProfileDB2.rsp	Use this file to create a stand-alone configuration using a DB2® database. A stand-alone configuration has a single node running an application server and one daemon server in a single z/OS® system or LPAR. Use a DB2® database for a production system.

Table 3. Sample response files (continued)

DmgrDB2.rsp	Use this file to create a deployment manager server with a network deployment configuration using a DB2 database. A basic network deployment configuration contains a deployment manager server in one node, and an application server in another node. The application server is then federated into the deployment manager cell which allows the application server to be managed by the deployment manager. In a network deployment configuration, both nodes are augmented with WebSphere Process Server functions.
ManagedDB2.rsp	Use this file to create a node with a network deployment configuration using a DB2 database. A basic network deployment configuration contains a deployment manager server in one node, and an application server in another node. The application server is then federated into the deployment manager cell which allows the application server to be managed by the deployment manager. In a network deployment configuration, both nodes are augmented with WebSphere Process Server functions.

The following table lists the default response file keywords that you can use to create a WebSphere Process Server for z/OS configuration in alphabetic order. Many of these keywords default to appropriate values if not set by a response file, and you do not need include all of these keywords in a response file to produce a working installation of WebSphere Process Server for z/OS.

Table 4. Alphabetic list of keywords in WebSphere Process Server for z/OS response files

-adminPassword	The password for the administrative security user ID specified with the -adminUserName keyword. This keyword is required when augmenting an existing profile that has administrative security enabled.
-adminUserName	The user ID that is used for administrative security. This keyword is required when augmenting an existing profile that has administrative security enabled.
-augment	The augment parameter uses an augmentation template to configure a node with WebSphere Process Server for z/OS functions.
-bpcDbConnectionLocation	The database location for the configuration of the Business Process Choreographer. This value is used in place of the database name on the WebSphere Process Server data source definition. For example with a stand-alone configuration DB2=LOC1.
-bpcDbHostName	The database server host name or IP address for the configuration of the Business Process Container database.
-bpcDbJDBCClasspath	The location of JDBC driver files.
-bpcDbName	The database name for the configuration of the Business Process Container database.

Table 4. Alphabetic list of keywords in WebSphere Process Server for z/OS response files (continued)

-bpcDbProduct	The database location for the configuration of the Business Process Container. For example, with a stand-alone configuration DB2=DB2UDBOS390_V8_1. Possible values for this keyword are: <ul style="list-style-type: none"> • DERBY_EMBEDDED • DERBY_NETWORKSERVER • DB2_UNIVERSAL • DB2UDBOS390_V7_1 • DB2UDBOS390_V8_1 • DB2UDBOS390_V9_1 • DB2UDBISERIES_NATIVE • DB2UDBISERIES_TOOLBOX • INFORMIX • MSSQLSERVER_EMBEDDED • MSSQLSERVER_DATADIRECT • ORACLE9I, ORACLE10G
-bpcDbPassword	The password for authenticating the JDBC resources for the Business Process Container.
-bpcDbSqlId	The DB2 Schema qualifier used by the Business Process Container. This value is substituted in the install produced ddl/sql.
-bpcDbUser	The DB2 user ID that has privileges to create and drop the Business Process Container databases.
-bpcDbServerPort	The port number of the Business Process Container database.
-bpcDbStorageGroup	The DB2 schema qualifier used by the Business Process Container. This value is substituted in the ddl/sql that is produced when installing. For example, with a stand-alone configuration DB2=BPEDBSTO.
-bpcmqPassword	The password for authenticating the MQ resources for the Business Process Container.
-bpcmqUser	The user name for the configuration of resources for the Business Process Container which uses WebSphere Process Server MQ.
-bspaceAlreadyConfigured	A boolean keyword used to specify if Business Space has already been configured for the profile.
-bspaceAlreadyDeployed	A boolean keyword used to specify if Business Space has already been deployed to the server.
-bspaceSchemaName	The DB2 Schema qualifier used for the Business Space tables. This keyword is substituted in the install produced ddl/sql.
-cbeServerName	The name of the server where the Common Base Event Browser is configured to run.

Table 4. Alphabetic list of keywords in WebSphere Process Server for z/OS response files (continued)

-cdbSchemaName	<p>The schema name of the Common Data Base. This parameter valid for all dbTypes except for Oracle/Informix/MSSQLSERVER_DataDirect/MSSQLSERVER_MICROSOFT Do not use this keyword for dbTypes DB2UDBOS390_V8_1/DB2UDBOS390_V9_1/DB2UDBISERIES_TOOLBOX if dbSchemaName is set. The following are the default schema values for the databases:</p> <ul style="list-style-type: none"> • Derby_Embedded/Derby_Embedded40 is "APP" • Derby_NetwrokServer/Derby_NetwrokServer40 is "dbUserId" • DB2_Universal/DB2_dataServer is "dbUserId" • DB2UDBOS390_V8_1/DB2UDBOS390_V9_1 is "dbUserId" • DB2UDBISERIES_TOOLBOX is "dbUserId" <p>Note: You can specify a different schema for supported databases, but Recovery table is always created using default schema if dbTypes are Derby_Embedded/ Derby_Embedded40/DB2_Universal/DB2_dataServer/ Derby_NetwrokServer/Derby_NetwrokServer40. For example: if cdSchemaName is "mySchema" , dbUserId "myUser" for derby_embedded, recovery tables schema is APP not mySchema for derby_networkservr, recovery tables schema is dbUserId not mySchema for db2_universal/ db2_dataserver, recovery tables schema is dbUserId not mySchema.</p>
-ceiAlreadyDeployed	A boolean keyword used to specify if the Common Event Infrastructure has already been deployed to the server.
-ceiBufferPool4k	The name of the 4K buffer pool for the Common Event Infrastructure. This buffer pool must be active before the database DDL scripts can be run.
-ceiBufferPool8k	The name of the 8K buffer pool for the Common Event Infrastructure. This buffer pool must be active before the database DDL scripts can be run.
-ceiBufferPool16k	The name of the 16K buffer pool for the Common Event Infrastructure. This buffer pool must be active before the database DDL scripts can be run.
-ceiCacheSizeInMB	The size of the cache in MB that will be used for transaction logs.
-ceiCreateLogin	An optional parameter, which if set to true results in the creation of the login user ID that will own the Event Service Sybase tables for the Common Event Infrastructure.
-ceiDbInstallDir	The directory where the database is installed for the Common Event Infrastructure.
-ceiDbName	The DB2 name for the Common Event Infrastructure.
-ceiDiskSizeInMB	The database size in MB to be created for the Event Service. For DB2 for z/OS, the default value is 10. If you want to specify another value it must be greater than or equal to 10.
-ceiEventCatalogDBName	The name of the event catalog database.
-ceiInstancePrefix	The prefix used for the CEI instance.

Table 4. Alphabetic list of keywords in WebSphere Process Server for z/OS response files (continued)

-ceiFindDeviceNumber	The event database creates six devices for the Common Event Infrastructure. This keyword identifies the value of the first device number that should be assigned to the new devices. The default value is 10 if not specified.
-ceiOverrideDataSource	The keyword that indicates whether or not to remove any existing Common Event Infrastructure service data source at the specified scope before creating a new one. When this keyword is set to true, the command removes any existing Common Event Infrastructure service data source at the specified scope before creating a new one. When this keyword is set to false, the command does not create an event service data source at the specified scope if another event service data source is found at the same scope. The default value is false if not specified.
-ceiOracleHome	(Deprecated) The directory of the ORACLE_HOME.
-ceiSaUser	The Microsoft SQL Server ID that has privileges to create tables, devices, and caches for the Common Event Infrastructure.
-ceiSaPassword	The password for the Microsoft SQL Server ID that has privileges to create tables, devices, and caches for the Common Event Infrastructure. This keyword is required if you specify a value for the ceiSaUser keyword, unless the sa user ID does not have a password.
-cellName	The cell name of the profile. This must match the cell name of the WebSphere Application Server profile that you want to augment. The value for this keyword must not contain spaces, commas or any characters that are not valid such as the following: /, \, *, :, ;, =, +, ?, , <, >, &, %, ', " ,]>, #, \$, ^, {, or }. A period (.) is not valid if it is the first character.
-configureBPC	The keyword that determines whether the Business Process Choreographer sample configuration is created. If you set this keyword to true, the -adminUserName and -adminPassword keywords also must be set. The default for this keyword is the same value as the -enableAdminSecurity keyword. Note: The Business Process Choreographer sample configuration does not use the common database (WPRCSDB). It always uses a Derby database, which is not supported in a network deployment environment. If you plan to federate this stand-alone server later, set -configureBPC to false.
-configureBRM	The keyword that determines whether or not to configure the business rules manager. The default value is false.
-configureBSpace	The keyword that determines whether or not to configure the Business Space. The default value is false.
-createDefaultProfileForMigration	A boolean keyword used to specify that the server will be the target of a version to version migration. Set this value to true when you are augmenting a standalone server to be used as a migration target. This keyword will prevent the creation of a duplicate CEI messaging engine.
-dbAlreadyConfigured	A boolean keyword used to specify if the database has already been configured.

Table 4. Alphabetic list of keywords in WebSphere Process Server for z/OS response files (continued)

-dbAppMeUserId	The user ID required for authentication if messaging engines use the common DB.
-dbAppMePassword	The password required for authentication if messaging engines use the common DB.
-dbBPCMeUserId	The user ID required for Business Process Choreographer Messaging Engine object creation. This keyword is only valid if -configureBPC is set to true.
-dbBPCMePassword	The password required for Business Process Choreographer Messaging Engine object creation. This keyword is only valid if -configureBPC is set to true.
-dbCeiMeUserId	The user ID required for authentication if the CEI messaging engines use the common DB.
-dbCeiMePassword	The password required for authentication if CEI messaging engines use the common DB.
-dbCommonForME	The keyword that indicates whether to use the common database for the messaging engine tables. The default value is false. If you set this keyword to true the common database is used for the messaging engine tables which must be manually created in the common database. See "Creating the messaging engine data stores" on page 78 for more information.
-dbConnectionLocation	The location of DB2 for z/OS database.
-dbCeiUserId	The user ID required for authentication of the CEI database.
-dbCommonUserId	The user ID required for authentication of the common database.
-dbCeiPassword	The password required for authentication of the CEI database.
-dbCommonPassword	The password required for authentication of the common database.
-dbCreateNew	A boolean flag to indicate whether to create a new database in which to store the Common DB, or whether to use an existing database.
-dbDelayConfig	The keyword that indicates whether to postpone table creation until after the profile is created. Valid values are true or false. This keyword is set to false by default. Whether you set this keyword to true or false depends on your configuration. If you have all the necessary databases set up and ready you can set -dbDelayConfig to true. If you do not have all the necessary databases set up and ready you can set -dbDelayConfig to false and set up the databases after augment.
-dbDriverType	The database driver type. For Oracle databases, valid values are THIN or OCI. For DB2 databases, valid values are 2 or 4.
-dbHostName	The database server host name or IP address. The default value is localhost.
-dbJDBCClasspath	The JDBC class path required for the common DB.
-dbInstance	The database instance name for Informix databases.
-dbLocation	The location of the database server (database product installation root).

Table 4. Alphabetic list of keywords in WebSphere Process Server for z/OS response files (continued)

-dbName	The name of the WebSphere Process Server database. The default value is WPRCSDB.
-dbOutputScriptDir	The directory location of the output script.
-dbPassword	The password required for database authentication. This keyword is required for all databases except Derby embedded.
-dbProviderType	An optional parameter that specifies the provider type for the current dbType. Currently only applicable to the Informix dbType.
-dbSchemaName	The name of the database schema. This keyword is valid for dbTypes DB2UDBOS390_V8_1/DB2UDBOS390_V9_1/DB2UDBISERIES_TOOLBOX and is deprecated in version 7, use the cdbSchemaName keyword instead.
-dbServerPort	The database server port number. Depending on the database you are using, you can specify a different port number instead of the default port number which is 446.
-dbStorageGroup	The name of the storage group for the DB2 for z/OS databases.
-dbSysMeUserId	The database system messaging engine user ID.
-dbSysMePassword	The database system messaging engine password.
-dbSysUserId	The user ID to gain access to the database.
-dbSysPassword	The password to gain access to the database.
-dbType	The database type. Set one of the following values for the type of database product you are using with the WebSphere Process Server database: <ul style="list-style-type: none"> • DERBY_EMBEDDED for a Derby Embedded database • DB2UDBOS390_V8_1 for a DB2 for z/OS v8 database • DB2UDBOS390_V9_1 for a DB2 for z/OS v9 database
-dbUserId	The user ID that is required for database authentication. This keyword is required for all databases except Derby embedded.
-dmgrAdminPassword	The password that gains the user administration access rights to the system in which the deployment manager is running.
-dmgrAdminUserName	The user name that gains the user administration access rights to the system in which the deployment manager is running.

Table 4. Alphabetic list of keywords in WebSphere Process Server for z/OS response files (continued)

-dmgrHost	<p>The keyword that identifies the system in which the deployment manager is running. Specify this keyword with the dmgrPort keyword to federate an empty node when it is created. The host name can be the long or short DNS name or the IP address of the deployment manager system. Specifying this optional keyword directs the configuration process to attempt to federate the empty node into the deployment manager cell when it creates the empty node.</p> <p>This keyword is ignored when creating a deployment manager or an application server. If you federate an empty node when the deployment manager is not running or is not available because of security being enabled or for other reasons, the installation indicator in the logs is INSTCONFFAIL to indicate a complete failure. The resulting empty node is unusable. You must move the profile directory of the empty node out of the profile repository (the profiles installation root directory) before creating another empty node with the same name.</p> <p>If you have enabled security or changed the default JMX connector type, you cannot federate during the configuration process. Use the addNode command instead. The default value for this keyword is localhost. The value for this keyword must be a properly formed host name and must not contain spaces or characters that are not valid such as the following: *, ?, ", \, <, >, ,, /, \, , A connection to the deployment manager must also be available in conjunction with the dmgrPort keyword. For example dmgr_host_name.</p>
-dmgrPort	<p>The keyword that identifies the SOAP port of the deployment manager. Specify this keyword with the dmgrHost keyword to federate an empty node when it is created. The deployment manager must be running and accessible. If you have enabled security or changed the default JMX connector type, you cannot federate during the configuration process. Use the addNode command instead. The default value for this keyword is 8879. The port that you indicate must be a positive integer and a connection to the deployment manager must be available in conjunction with the dmgrHost keyword. For example dmgr_port_number.</p>
-enableAdminSecurity	<p>The keyword that enables administrative security. Possible values are:</p> <ul style="list-style-type: none"> • True - when enableAdminSecurity is set to true you must also specify the keywords -adminUserName and -adminPassword along with the values for these keywords. • False - when enableAdminSecurity is set to false, you do not have to set any keywords. <p>The default value is false.</p>
-fileStoreForME	<p>The location of the file store for messaging engines. This keyword applies to stand-alone server configurations only. Default value is false, cannot be set to true when -dbCommonForME is also set to true.</p>

Table 4. Alphabetic list of keywords in WebSphere Process Server for z/OS response files (continued)

-isDeveloperServer	Specifies whether the server is intended for development purposes only.
-portsFile	An optional keyword that specifies the path to a file that defines port settings for the new profile.
-profileName	The keyword used to augment a profile. The profile is always named default in each of the configurations. The profile must not have already been federated. When augmenting the profile the server must not be running.
-profilePath	The default profile root directory for WebSphere Application Server for z/OS. This keyword is mandatory. The server configuration for WebSphere Application Server for z/OS resides in a directory structure under a profiles directory. The profile path contains the files that define the runtime environment, such as commands, configuration files, and log files. Specify the full path to avoid an Apache Ant scripting limitation which can cause a failure when federating the profile into a cell. If not specified, the augment procedure looks up the path where the profile resides in the WebSphere Application Server for z/OS configuration, for example: <ul style="list-style-type: none"> • Stand-alone Derby=/WebSphere/V7R0/AppServer/profiles/default • Stand-alone DB2=/WebSphere/V7R0/AppServer/profiles/default • Network Deployment=/WebSphere/V7R0/DeploymentManager/profiles/default • Managed node in Network Deployment=/WebSphere/V7R0/AppServer/profiles/default.
-serverNameplatform	The name of the platform in which the server is running.
-serverType	The keyword that specifies the type of management profile. Specify DMGR for a management profile. This keyword is required when you create a management profile.
-soaCoreAugmentType	The keyword that specifies the type of SOA core augment information.
-topologyPattern	The keyword that determines the topology patterns for your deployment manager: none (default value), CondensedSync, CondensedAsync or Reference.
-topologyRole	The keyword that indicates the function the profile will play in the deployment environment, when you are federating a profile that has been created. Valid values are ADT for a deployment target, Messaging for host messaging or Support for supporting services. You can indicate one value or more than one value, each separated by a space, for example ADT Messaging Support or Messaging or ADT Support.
-wbidbDesign	The keyword that is used to specify a single design file for all of the components during profile creation. You must specify the fully qualified path to the design_file

Using response files

- 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 keyword 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/
```

Global properties

The start 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 displayed in multiple places in the response file because they are used by multiple components. The **GLOBAL Properties** section provides a central location for these common values to help you editing these values. For example, you can use a global property to set all the server property instances to **server1** instead of searching the response file for all instances of the property that you want to change.

Use the GLOBAL property by specifying the global keyword prefixed with the \$ symbol as the value for the subsequent property. For example, `serverName=$CONFIGSERVER`.

Global properties must physically appear in the response file before they are referenced by the \$ symbol.

When a global property is used, it must constitute the whole value of a property. The following example is NOT allowed:

```
templatePath=/usr/$USERPATH/dir
```

Global properties simplify the use of the override argument (-Z) in the product configuration command by reducing the amount of syntax on the command line.

Comments for a property display above the property, the property name and the default value are displayed in bold text. The following is an example of a property:

```
#####  
#  
# 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
```

Related concepts

“Network deployment configuration” on page 9

An initial network deployment configuration consists of a deployment manager server that has a daemon for the z/OS[®] system on which the deployment manager runs. After a network deployment cell is created, you can add application server nodes by creating and federating new empty managed nodes, or by federating a stand-alone application server node into the network deployment cell.

Related tasks

 Creating a stand-alone configuration with a Derby database

A stand-alone configuration is the simplest configuration type in WebSphere Process Server for z/OS. A stand-alone configuration has a single node running an application server and one daemon server in a single z/OS[®] system or LPAR. Use a Derby database only for a test system. Use a DB2 database for a production system.

“Creating a network deployment configuration with WebSphere Process Server” on page 68

A basic network deployment configuration contains a deployment manager server in one node, and an application server in another node. In a WebSphere Process Server network deployment configuration, both nodes are augmented with WebSphere Process Server functions. The application server is then federated into the deployment manager cell, which allows the application server to be administered by the deployment manager. The application server must be augmented with WebSphere Process Server functions before it is federated into the deployment manager cell.

“Creating a stand-alone configuration” on page 66

A stand-alone configuration is the simplest configuration type in WebSphere Process Server for z/OS. A stand-alone configuration has a single node running an application server and one daemon server in a single z/OS system or LPAR. You can create a stand-alone configuration that uses either a Derby or DB2 type database. Use a Derby database only for a test system. Use a DB2 database for a production system. When you use a DB2 database you must make considerable changes to the response file, and you must set up and configure the database. If you want to quickly set up a stand-alone configuration for evaluation or demonstration purposes, you might prefer to use a Derby database instead.

Using the zPMT tool

zPMT stands for z/OS Profile Management Tool. The zPMT tool is a workstation based tool that captures information and generates customized JCL batch jobs and configuration response files that you can use to create your WebSphere Process Server for z/OS system.

The essential features of the zPMT tool include its ability to perform the following tasks:

- Capturing information.
- Generating customized JCL batch jobs and configuration response files.
- Uploading the JCL batch jobs and configuration response files to the z/OS system.

The zPMT tool is part of WebSphere Customization Tools (WCT), an Eclipse based tool. To use the zPMT tool you have to install the WCT on your workstation and start it, then start the zPMT tool from there. For information about how to install WebSphere Customization Tools, see *Installing WebSphere Customization Tools*.

You can run the zPMT tool on your workstation and enter information in a series of windows. The information you provide is stored in the customization location of WCT which is a directory structure on your hard drive. The generated JCL jobs and a response file that create your WebSphere Process Server for z/OS servers are stored in this directory structure.

The JCL files are sent to a z/OS system using the process feature of the zPMT tool. The files are placed in two partitioned data sets named CNTL and DATA.

- The CNTL data set contains the generated JCL jobs you can use to install and configure WebSphere Process Server for z/OS.
- The DATA data set contains the configuration response file that is passed to the augmentation scripts.

You can use the zPMT tool to allocate the data sets as part of the upload.

The process of planning a cell is the same with the zPMT tool as it was with the ISPF panels. The zPMT tool prompts you for the same information as the ISPF panels, but in a few cases in a slightly different sequence. Naming conventions and port allocation charts are the same with the zPMT tool as they were with the ISPF panels.

Types of configuration that you can build with the zPMT tool

You can build three types of configuration with the zPMT tool. The following table cross-references the zPMT tool terms with the ISPF terms.

Table 5. Customization environments available using the zPMT tool

zPMT tool option	Equivalent ISPF option
WebSphere Process Server for z/OS Process Server	Stand alone Process Server node
WebSphere Process Server for z/OS deployment manager	Network deployment cell
WebSphere Process Server for z/OS managed (custom) node	Empty managed node

The zPMT tool creates a saved configuration definition in the workspace. The upload feature allows you to FTP the files to a z/OS system.

The variables you enter into the zPMT tool are saved as part of the workspace on your hard drive. That workspace is where the tool maintains all its information about your projects. The variables entered for one configuration are put into a

zPMT tool response file which is a flat file with the variables and your values. This response file is similar in concept, but not identical in format, to the SAVECFG files for the ISPF dialogs.

The zPMT tool response file is different to the response file used to augment. Extra variables are used for customizing jobs, for example 4 variables for the JCL header to configure jobs. The response file used to augment is obtained when you click **process**, and is used to upload customized jobs to a z/OS system or local directory.

The name of this response file follows the naming format BPZRSPX where X can be:

- A = stand alone node.
- M = deployment manager node.
- N = managed node.

Configuring WebSphere Process Server

After you have installed WebSphere Process Server, you must complete additional configuration tasks to fully prepare your runtime environment.

Configuring WebSphere Process Server for z/OS

A configuration script augments a node to add support for WebSphere Process Server for z/OS functions. You must decide on the topology of your WebSphere Process Server for z/OS cell and then define and augment all the nodes that will run WebSphere Process Server for z/OS functions.

About this task

You set values for various parameters in a response file that is passed as input to the configuration shell scripts. There are two ways to create a response file: you can use the zPMT tool to generate the response file, or you can edit sample response files that are supplied with the product. In Creating common configurations you will find a high-level task list that guides you through planning and implementing different WebSphere Process Server topologies.

Designing the DB2 database objects

There are various ways to organise the DB2 for z/OS objects which are required to support a WebSphere Process Server instance. The decisions may be influenced by your organisation's standards or conventions, or by practices which make management of the DB2 subsystem easier. Work with the DB2 subsystem administrator to design the approach for naming conventions and use of databases, storage groups, buffer pools and other DB2 resources, and use the DbDesignGenerator.sh tool to implement the database design.

The DbDesignGenerator.sh tool

Use the DbDesignGenerator.sh tool to specify the database configuration values for a set of WebSphere Process Server database objects. The DbDesignGenerator.sh tool produces two things:

1. The database design file, which is used to store the values entered into the DbDesignGenerator.sh tool. This file may be referred to in the augmentation response file `wbidbDesign` parameter to populate the attribute values of WebSphere Process Server resource definitions. Refer to “Creating a response file” on page 58 for information about how to create the augmentation response file.
2. The DDL which should be run to create the DB2 database objects for the various WebSphere Process Server components. Refer to “Creating and configuring the databases” on page 72 for instructions on how to execute the DDL to create the database objects.

DB2 decisions

There are a number of decisions that you must make when configuring WebSphere Process Server for z/OS.

Deciding on a naming convention for the databases:

If you have only one server configured with WebSphere Process Server for z/OS using a DB2 subsystem, you can use the default database names as provided in the sample response files. For multiple WebSphere Process Server for z/OS-configured servers (in the same cell, or different cells) using the same DB2 subsystem or same data-sharing group, you must plan for a naming convention to isolate the following DB2 components:

- Database names
- Storage Group names
- Schema-qualifiers for your tables
- VCATs, which stands for VSAM catalog name. VCATs are the high level qualifiers used to prefix DB2 table spaces and tables for a DB2 sub system
- Volumes or SMS storage groups for these data sets

The following example is a database naming convention (for one database) in which the cell name is S5CELL:

Database	Dbase Name	Storage Group	Schema -Owner	VCAT DSN-h1q
WPS:	S5CELLDB	S5DBSTO	S5CELL	DSN810PP
BPC:	S5CELLDB	S5DBSTO	S5CELL	DSN810PP
APP SIB:	S5CELLDB	S5DBSTO	S5S1A	DSN810PP
BPC SIB:	S5CELLDB	S5DBSTO	S5S1B	DSN810PP
CEI SIB:	S5CELLDB	S5DBSTO	S5S1C	DSN810PP
SCA SIB:	S5CELLDB	S5DBSTO	S5S1S	DSN810PP
CEI:	S5CELLDB	S5DBSTO	S5CELL	DSN810PP
BSPACE:	S5CELLDB	S5DBSTO	S5CELL	DSN810PP

The following example is a database naming convention (for nine databases) in which the WebSphere Application Server cell name is B6CELL:

Database	Dbase Name	Storage Group	Schema -Owner	VCAT DSN-h1q
WPS:	B6WPSDB	B6WPSSTO	B6CELL	B6WPS
BPE:	B6BPEDB	B6BPESTO	B6CELL	B6WPS
SIBs:	B6SIBAPP	B6SIBSTO	B6CELLA	B6WPS
	B6SIBSCA	B6SIBSTO	B6CELLS	B6WPS
	B6SIBBPC	B6SIBSTO	B6CELLB	B6WPS
	B6SIBCEI	B6SIBSTO	B6CELLC	B6WPS
CEI:	B6EVTDB	B6EVTSTO	B6CELL	B6WPS
BSPACE:	B6BSPACE	B6BSPSTO	B6CELL	B6WPS

Deciding on schema names and SQL IDs:

It is important that you choose unique names for the schema names and SQL IDs in the DB2 tables.

The DB2 tables must have uniquely-qualified schema names to coexist with other WebSphere Process Server for z/OS or WebSphere Business Integration Server cells in the same DB2 subsystem. You can set the current schema or SQL ID values using the administrative console, in the Data source > Custom properties definitions.

The DB2 table and index names can be prefixed with these names in the DDL definitions, or you can insert a SET CURRENT SQLID statement in front of the DDL used to create the tables. For example:

```
SET CURRENT SQLID = 'B6CELL';
```

Each of the SIB databases requires unique schema names for its tables, because the same table names are used for all Buses. These same schema names must also be set in the SIB Messaging Engine Data Store properties. See Messaging engines in the WebSphere Application Server for z/OS information center.

Creating the database design file using the database design tool

Use the database design tool (DDT) to generate a design file that is used to create the database tables required by WebSphere Process Server. The DDT generates the design file from a user specified properties file or user interactive input. The resulting design file is then used by the DDT to create the database scripts that are used to create the database tables. Additionally, the design file can be used as input during profile creation and during deployment environment configuration to specify the database configuration properties.

Before you begin

You must have a list of all database requirements and schema names.

About this task

The following steps describe how to use the DDT to generate the design file and database scripts. The input for the DDT is either a user specified properties file or user interactive input.

The **DbDesignGenerator** command has the following options.

-? , -help
display help info.

-e db_design_file_name
edit the specified database design file (e.g. *.dbDesign, *.properties).

-v db_design_file | db_scripts_output_directory
when a db_design_file is given, validation will be done on the specified database design file based on the database specs.

When a db_scripts_output_directory is given, the database scripts in the specified directory will be validated. Currently only scripts generated from template ddl generator can be validated.

-g db_design_file [-d output_directory] [db_design_file2] [-d output_directory2] ... [db_design_fileN] [-d output_directoryN]
generate the database scripts from the specified design files in batch mode.

The generated scripts will be put in the corresponding output directories or the default locations if output directories are absent.

Note: The DDT does not support the generation of database scripts for Common Event Infrastructure.

Note: The following restrictions apply to stand-alone database design for Common Event Infrastructure components.

Table 6. Stand-alone Database design restrictions for CEI component:

Database type	CEI restrictions
DB2 Distributed	The database name must not be the same as the commonDB name. Edit the CEI database design and choose a different name.

Table 6. Stand-alone Database design restrictions for CEI component: (continued)

Database type	CEI restrictions
SQL Server	<ul style="list-style-type: none"> • dbServerName cannot be empty. Edit the CEI database design and provide the database server name. • dbUser cannot be the same as the CommonDB user ID. Edit the CEI database design and provide a different user ID. • sysUser and sysPassword cannot be empty. Edit the CEI database design and provide the system user ID and system password.
Oracle	<ul style="list-style-type: none"> • dbUser cannot be the same as the CommonDB user ID. Edit the CEI database design and provide a different user ID. • sysUser and sysPassword cannot be empty. Edit the CEI database design and provide the system user ID and system password.

Procedure

Generate the design file and database scripts using the **DbDesignGenerator** command, which is located in:

DbDesignGenerator.bat - for Windows

DbDesignGenerator.sh - for Unix and z/OS

Returns the main menu:

```
[info] running DbDesignGenerator in interactive mode...
```

```
[info] Enter 'q' to quit without saving; '-' for back to previous menu; '?' for help at any time.
```

```
[info] To accept the given default values, simply press the 'Enter' key.[info] Please pick one of the following [design option(s)] :
```

```
(1)Create a database design for Standalone profile or Deployment Environment
```

```
(2)Create a database design for a single component (e.g. BPC, CEI etc)
```

```
(3)Edit an existing database design
```

```
(4)Generate database scripts from a database design
```

```
(5)exit [q]
```

Creating a response file

There are two ways to create a response file that contains all the details of how you want to configure your system. You can either use the zPMT tool to generate the response file, or you can edit sample response files that are supplied with the product. See “Creating a response file using the zPMT tool” and “Creating a response file using the samples” on page 59 for more information.

Creating a response file using the zPMT tool

zPMT is the z/OS Profile Management Tool. The zPMT tool is a workstation based tool that captures information and generates customized JCL batch jobs and configuration response files that you can use to create your WebSphere Process Server for z/OS system.

You may already have used zPMT to create your response file. If you used zPMT to generate your installation job, then the response file will have been produced by that process. If you ran the installation from a USS command rather than as a batch job, then you may not have used zPMT to create a WebSphere Process Server configuration definition and the associated augmentation response file.

If you want to create the response file using zPMT, refer to “Using the zPMT tool” on page 11 for instructions. Alternatively you can customize one of the supplied sample response files, refer to “Creating a response file using the samples” for more information.

Creating a response file using the samples

WebSphere Process Server for z/OS is shipped with four sample response files that you can use to create a response file that contains all the details of how you want to configure your system.

Before you begin

- Install and configure WebSphere Application Server which is used as a base for your WebSphere Process Server configuration.

About this task

You can create a response file that you can use to create your WebSphere Process Server configuration. The response file you create is based on one of the following four sample response files:

standAloneProfile.rsp

Create a stand-alone configuration using a Derby™ database.

standAloneProfileDB2.rsp

Create a stand-alone configuration using a DB2® database.

DmgrDB2.rsp

Create a deployment manager server with a network deployment configuration using a DB2 database.

ManagedDB2.rsp

Create a node with a network deployment configuration using a DB2 database.

For more information about the keywords contained in these sample response files which you can tailor to meet the needs of your configuration, see Response file values.

Creating a response file for a stand-alone configuration with a Derby database:

You can create a response file to use when configuring WebSphere Process Server for z/OS in a stand-alone configuration with a Derby database. A stand-alone configuration is the simplest configuration type in WebSphere Process Server for z/OS. A stand-alone configuration has a single node running an application server and one daemon server in a single z/OS® system or LPAR. Use a Derby database only for a test system. Use a DB2® database for a production system.

Before you begin

Before you can create a response file for a stand-alone configuration of WebSphere Process Server with a Derby database, you must complete the following tasks:

- Install WebSphere Process Server

Procedure

1. Create a response file to provide input to the configuration script `zWPSConfig.sh`. A sample response file for a stand-alone server with a Derby database is supplied in `/usr/lpp/zWPS/V7R0/zos.config/standAloneProfile.rsp`.
 - a. Copy the sample response file, `standAloneProfile.rsp`, to your working directory. For example:

```
cp /usr/lpp/zWPS/V7R0/zos.config/standAloneProfile.rsp /u/work/
```
 - b. Use the **chmod** command to assign the appropriate permissions to the copy of the response file. For example:

```
chmod 755 standAloneProfile.rsp
```
 - c. Edit the parameters in the response file as appropriate to your configuration.
 - d. Save the edited response file.
2. Stop the server if it is started.
3. Increase the OMVS time limit to allow the product configuration script time to complete. In an MVS™ console enter the following command:

```
SETOMVS MAXCPU=86400
```
4. Access the USS command shell, then switch to the administrator user ID. For example:

```
su wsadmin
```
5. Change directory to the application server bin directory:

```
cd /WebSphere/V7R0/AppServer/bin
```

Results

You are now ready to configure your WebSphere Process Server installation. See “Creating common configurations” on page 63 for more information.

Creating a response file for a stand-alone configuration with a DB2 database:

You can create a response file to use when configuring WebSphere Process Server for z/OS in a stand-alone configuration with a DB2 database. A stand-alone configuration has a single node running an application server and one daemon server in a single z/OS® system or LPAR. When you use a DB2 database you must make considerable changes to the response file, and you must set up and configure the database. If you want to quickly set up a stand-alone configuration for evaluation or demonstration purposes, you might prefer to use a Derby database instead.

Procedure

1. Create and customize a `DB2JccConfiguration.properties` file to provide WebSphere Application Server with information about the DB2 sub system:
 - a. Create the file in a suitable directory. For example `/etc/db2cfg`.
 - b. Set the permissions on the directory so that the WebSphere Control Region user ID and the user ID that runs your WebSphere Process Server configuration jobs can read the properties file.
 - c. Make sure the `DB2JccConfiguration.properties` file contains the following line:

```
db2.jcc.ssid=DB15
```

Where *dbn* is the SSID of your installation, for example DB15. For details of all the other possible properties you can change in the DB2JccConfiguration.properties files refer to the manual *DB2 for z/OS V8.1 Application Programming Guide and Reference for Java* (SC18-741).

2. Create a response file to provide input to the configuration script zWPSCfg.sh. A sample response file for a stand-alone server with DB2 is supplied in `usr/lpp/zWPS/V7R0/zos.config/standaloneProfileDB2.rsp`.
 - a. Copy the sample response file, `standaloneProfileDB2.rsp`, to your working directory. For example:

```
cp /usr/lpp/zWPS/V7R0/zos.config/standaloneProfileDB2.rsp /u/work/
```
 - b. Use the **chmod** command to assign the appropriate permissions to the copy of the response file. For example:

```
chmod 755 standaloneProfileDB2.rsp
```
 - c. Set the `-dbDelayConfig` parameter to true to prevent the configuration script automatically running the DDL scripts to create the database objects. You will run the DDL scripts later.
 - d. Edit the other parameters in the response file as appropriate to your configuration.
 - e. Save the edited response file.

Results

You are now ready to configure your WebSphere Process Server installation. See “Creating common configurations” on page 63 for more information.

Creating a response file for a network deployment manager configuration:

A basic network deployment configuration contains a deployment manager server in one node, and an application server in another node. In a WebSphere Process Server network deployment configuration, both nodes are augmented with WebSphere Process Server functions. The application server is then federated into the deployment manager cell, which allows the application server to be administered by the deployment manager. The application server must be augmented with WebSphere Process Server functionality before it is federated into the deployment manager cell.

Procedure

1. Create and customize a DB2JccConfiguration.properties file to provide WebSphere Application Server with information about the DB2 sub-system:
 - a. Create the file in a suitable directory. For example `/etc/db2cfg/directory`.
 - b. Set the permissions on the directory so that the WebSphere Controller Region user ID and the user ID that runs your WebSphere Process Server configuration jobs can read the properties file.
 - c. Make sure the DB2JccConfiguration.properties file contains the following line:

```
db2.jcc.ssid=DB15
```

Where DB15 is the SSID of your installation. For details of all the other possible properties you can change in the DB2JccConfiguration.properties files refer to the manual *DB2 for z/OS V8.1 Application Programming Guide and Reference for Java* (SC18-741).

2. Create a response file to provide input to the configuration script `zWPSConfig.sh`. A sample response file for a stand-alone server with DB2 is supplied in `/usr/lpp/zWPS/V7R0/zos.config/DmgrDB2.rsp`.
 - a. Copy the sample response file, `DmgrDB2.rsp`, to your working directory. For example:


```
cp /usr/lpp/zWPS/V7R0/zos.config/DmgrDB2.rsp /u/work/
```
 - b. Use the **chmod** command to assign the appropriate permissions to the copy of the response file. For example:


```
chmod 755 DmgrDB2.rsp
```
 - c. Set the `-dbDelayConfig` parameter to true to prevent the configuration script automatically running the DDL scripts to create the database objects. You will run the DDL scripts later.
 - d. Edit the other parameters in the response as appropriate to your system.
 - e. Save the edited response file.

What to do next

You are now ready to configure your WebSphere Process Server installation. See “Creating common configurations” on page 63 for more information.

Creating a response file for an empty node:

A basic network deployment configuration contains a deployment manager server in one node, and another managed node in which application servers may be created. In a WebSphere Process Server network deployment configuration, both nodes are augmented with WebSphere Process Server functions. The managed node is then federated into the deployment manager cell, which allows the application servers to be created and administered by the deployment manager. The managed node must be augmented with WebSphere Process Server functionality before it is federated into the deployment manager cell.

Procedure

1. Create and customize a `DB2JccConfiguration.properties` file to provide WebSphere Application Server with information about the DB2 subsystem:
 - a. Create the file in a suitable directory. For example `/etc/db2cfg/directory`.
 - b. Set the permissions on the directory so that the WebSphere Controller Region user ID and the user ID that runs your WebSphere Process Server configuration jobs can read the properties file.
 - c. Make sure the `DB2JccConfiguration.properties` file contains the following line:


```
db2.jcc.ssid=DB15
```

Where DB15 is the SSID of your installation. For details of all the other possible properties you can change in the `DB2JccConfiguration.properties` files refer to the manual *DB2 for z/OS V8.1 Application Programming Guide and Reference for Java* (SC18-7414).
2. Create a response file to provide input to the configuration script `zWPSConfig.sh`. A sample response file for an empty managed node is supplied in `/usr/lpp/zWPS/V7R0/zos.config/ManagedDB2.rsp`.
 - a. Copy the sample response file, `ManagedDB2.rsp`, to your working directory. For example:


```
cp /usr/lpp/zWPS/V7R0/zos.config/ManagedDB2.rsp /u/work/
```

- b. Use the **chmod** command to assign the appropriate permissions to the copy of the response file. For example:


```
chmod 755 ManagedDB2.rsp
```
 - c. Edit the other parameters in the response file as appropriate to your configuration. See “Response file values” on page 41.
 - d. Save the edited response file.
3. Increase the OMVS time limit to allow the product configuration script time to complete. In an MVS console enter the following command:


```
SETOMVS MAXCPU=86400
```
 4. Access the USS command shell, then switch to the administrator user ID. For example:


```
su wsadmin
```
 5. Change directory to the application server bin directory:


```
cd /WebSphere/V7R0/AppServer/bin
```

What to do next

You are now ready to configure your WebSphere Process Server installation. See “Creating common configurations” for more information..

Creating common configurations

This section describes some of the common configurations that you can create in WebSphere Process Server for z/OS.

Augmenting with WebSphere Process Server

After installation of the WebSphere Process Server code into your WebSphere Application Server configuration, your existing WebSphere Application Server profile must be augmented into a WebSphere Process Server profile to enable WebSphere Process Server functionality.

The WebSphere Process Server for z/OS server configuration is stored in a directory structure under the WebSphere Application Server for z/OS profiles directory: `/WebSphere/V7R0/AppServer/profiles/default` or `/WebSphere/V7R0/DeploymentManager/profiles/default`. This directory contains all of the directories and files specific to the server. It is also used to store the server and FFDC logs.

In WebSphere Application Server for z/OS, all runtime environments are created using a profile name of default. The `manageprofiles` command, and the `-profile` option on other administrative commands, are not used with WebSphere Application Server for z/OS.

Before you begin

Determine which database you will use to support the WebSphere Process Server implementation. The choice of database you use depends on the implementation topology:

- If you want to configure a network deployment environment, you must use a DB2 database
- If you want to configure a stand-alone server, you can use either a Derby or a DB2 database.
- If you want to configure clustering, your DB2 z/OS system must be running in data-sharing mode.

If this is your first WebSphere Process Server for z/OS installation, start with a stand-alone configuration with a Derby database. After installing a stand-alone configuration with a Derby database, you can configure a stand-alone server with a DB2 database to become familiar with using DB2. You can then create a network deployment configuration that uses a DB2 type database, which is a more complex procedure.

If you have used zPMT to plan your configuration, there are a series of JCL jobs that have been generated to perform augmentation. See “Augmenting profiles with the zPMT generated augment job” on page 65 for more details, or refer to the zPMT generated instructions.

If you have not used zPMT, you will first need to construct a response file (or modify an existing or sample file) containing the profile-specific parameters used during the augment process to configure your system. See “Creating a stand-alone configuration” on page 66 or “Creating a network deployment configuration with WebSphere Process Server” on page 68 for details on that process. When you have your response file, you must use a USS script provided by WebSphere Process Server to perform augmentation. See “Augmenting profiles with the USS command” for more details.

Augmenting profiles with the USS command:

Augmentation is the ability to change an existing profile with an augmentation template. You can augment existing WebSphere Application Server or WebSphere Application Server network deployment profiles into WebSphere Process Server profiles.

Before you begin

Before using this procedure, make sure that you have completed the following tasks:

- Review the prerequisites for creating or augmenting a profile.
- Shut down any servers associated with the profile that you want to augment.
- Make sure that you know the type of profile that you want to augment.
- Create a response file.

About this task

You can augment a profile from USS using the zWPSCConfig.sh or zWESBConfig.sh scripts, depending on the type of profile that you want to configure. To perform the augmentation of a profile from USS, complete the following steps:

Procedure

1. Stop the server
2. Increase the OMVS time limit to allow the product configuration script time to complete. In an MVS console enter the following command:
`SETOMVS MAXCPUIME=86400`
3. Access the USS command shell, the switch to the administrator user ID. For example:
`su wsadmin`
4. Change directory to the server bin directory. For example:
`cd /WebSphere/V7R0/AppServer/bin`

or

```
cd /WebSphere/V7R0/DeploymentManager/bin
```

5. Run the zWPSConfig.sh configuration script with the absolute path of your edited response file. For example:

```
zWPSConfig.sh -augment -response /working_directory_path/standAloneProfileDB2.rsp
```

This will augment for a DB2 database, where `working_directory_path` is the location of the `standAloneProfileDB2.rsp` file that you have edited. For more information about the command, see `zWPSConfig.sh` script.

6. Wait for the configuration to run.

What to do next

When the script has finished running, review the messages that are written to the console. You can check that your profile augmentation completed successfully if you receive the following message:

INSTCONFSUCCESS: Profile augmentation succeeded.

You can check the following log file:

```
install_root/logs/manageprofiles/profile_name_augment.log
```

Augmenting profiles with the zPMT generated augment job:

Once a WebSphere Process Server zPMT customization definition has been processed, several JCL jobs will be generated to govern installation and augmentation of your existing WebSphere Application Server configuration into a WebSphere Process Server configuration. This topic covers the augmentation phase of this process. Fully customized instructions are also available in the customization instructions tab of your zPMT customization definition.

About this task

Before augmenting profiles with the zPMT generated job you must have completed the following tasks:

- Completed your configuration planning in zPMT, processed your customization definition and uploaded the resulting generated dataset members to your target z/OS system.
- Ensure that your target WebSphere Application Server configuration is fully configured with WebSphere Application Server, and has been installed with WebSphere Process Server code. See “Installing WebSphere Process Server” on page 19 for more information.

Procedure

1. Stop the target configuration and take a back up.
2. Navigate to the location of the customization jobs that were uploaded by zPMT.
3. Run the augmentation job or jobs required to augment your target WebSphere Application Server profile with WebSphere Process Server functionality. You can find the names of the jobs to run in the customization instructions tab of your zPMT customization definition.
 - a. If you would like to do this in one job, follow the instructions for MVS Single-job Augmentation Steps.

- b. If you would like to do this in a series of jobs, follow the instructions for MVS Step by Step Augmentation Steps.
4. Restart your target configuration.

Results

Your target WebSphere Application Server configuration has now been successfully augmented to a WebSphere Process Server configuration.

Creating a stand-alone configuration

A stand-alone configuration is the simplest configuration type in WebSphere Process Server for z/OS. A stand-alone configuration has a single node running an application server and one daemon server in a single z/OS system or LPAR. You can create a stand-alone configuration that uses either a Derby or DB2 type database. Use a Derby database only for a test system. Use a DB2 database for a production system. When you use a DB2 database you must make considerable changes to the response file, and you must set up and configure the database. If you want to quickly set up a stand-alone configuration for evaluation or demonstration purposes, you might prefer to use a Derby database instead.

Before you begin

Before you configure WebSphere Process Server, you must complete the following tasks:

- Install WebSphere Process Server. See *Installing WebSphere Process Server* for more information.
- Create a response file for a stand-alone configuration that uses either a Derby or DB2 database, either by using the sample files or the zPMT tool.

Procedure

1. Stop the server.
2. Run the zWPSConfig.sh script from the command line. For example:

```
install_root/zWPSConfig.sh -augment -response responseFile
```

The command displays status as it runs. Wait for it to finish.
3. Run the augment job using either a USS command or JCL generated by zPMT. Refer to “Augmenting profiles with the USS command” on page 64 or “Augmenting profiles with the zPMT generated augment job” on page 65 for more information.
4. Wait for the configuration to run.
When the script has finished running, review the messages that are written to the console. If the script has run successfully, no error messages are displayed and the informational messages state augmenting profile(s) complete.
5. Start the server. See *Starting stand-alone servers* for more information.
6. Verify that WebSphere Process Server information appears in the administrative console:
 - a. Open the administrative console by opening a browser window and typing the URL of the server that you want to view. For example:

```
http://server_name.domain_name:port_number/admin
```
 - b. Log in to the administrative console.
 - c. Verify that you can see WebSphere Process Server on the Welcome page. You can click on WebSphere Process Server for more information.

- d. Navigate around the administrative console to check applications and messaging engines are started.
7. Back up the data sets that contain the stand-alone server configuration.

Results

For a configuration that uses a Derby database, if you specified values for the Business Process Choreographer Sample Configuration in the stand-alone configuration response file, you have created a sample configuration that includes the business process container, the human task container, and the Business Process Choreographer Explorer. This sample configuration is 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.

A Derby sample configuration is not suitable for a production system. You can have only one Business Process Choreographer configuration, therefore you must remove the sample configuration, as described in Removing the Business Process Choreographer configuration before you can continue configuring Business Process Choreographer.

For a configuration that uses a DB2 database, the stand-alone server has now been augmented with WebSphere Process Server functions and the DDL scripts that you need to run to create the database objects have been generated.

What to do next

For a configuration that uses a Derby database, you can now deploy applications to the stand-alone server.

For a configuration that uses a DB2 database, you must now run the DDL scripts to create the DB2 database objects. See Create and configure the DB2 database objects. See “Creating and configuring the databases” on page 72 for more information.

Related reference

“Response file values” on page 41

When you have run the installation jobs which install the WebSphere Process Server for z/OS product definitions, sample response files are installed into the installation file system. Copy and edit these response files according to the configuration that you want to achieve, and pass in the response file when you run the augment script.

 zWPSConfig.sh script

Use this script to configure and augment the WebSphere Process Server for z/OS installation. Run the script on each node in your configuration, including the deployment manager.

 JCL for installing WebSphere Process Server into a deployment manager node

 JCL for augmenting a deployment manager node

 JCL for installing WebSphere Process Server into an empty node

 JCL for augmenting an empty node

 zWPSInstall.sh script

Use the zWPSInstall.sh script to modify a WebSphere Application Server profile either to install or uninstall WebSphere Process Server. Run the script on each node in your configuration, including the deployment manager.

Creating a network deployment configuration with WebSphere Process Server

A basic network deployment configuration contains a deployment manager server in one node, and an application server in another node. In a WebSphere Process Server network deployment configuration, both nodes are augmented with WebSphere Process Server functions. The application server is then federated into the deployment manager cell, which allows the application server to be administered by the deployment manager. The application server must be augmented with WebSphere Process Server functions before it is federated into the deployment manager cell.

Before you begin

Before you create a network deployment configuration you must complete the following tasks:

- Install WebSphere Process Server on each node that is in the deployment manager cell.
- Decide on a naming convention for the databases, storage groups, and schema (or SQL ID). Although you create and configure the DB2 database objects after configuration, you need the names of these database objects when you edit the response files that provide input to the configuration script.

Procedure

1. Create the deployment manager.
2. Configure an empty node.
3. Federate the empty node into the deployment manager cell.
4. Create a cluster.
5. Configure SCA and optionally other Business Process Management components as described in the Configuring the software section.

6. Create one or more DB2 databases for the configuration. See *Creating and configuring the databases*.

Related concepts

“Network deployment configuration” on page 9

An initial network deployment configuration consists of a deployment manager server that has a daemon for the z/OS® system on which the deployment manager runs. After a network deployment cell is created, you can add application server nodes by creating and federating new empty managed nodes, or by federating a stand-alone application server node into the network deployment cell.

Related reference


“Response file values” on page 41

When you have run the installation jobs which install the WebSphere Process Server for z/OS product definitions, sample response files are installed into the installation file system. Copy and edit these response files according to the configuration that you want to achieve, and pass in the response file when you run the augment script.

 zWPSConfig.sh script

Use this script to configure and augment the WebSphere Process Server for z/OS installation. Run the script on each node in your configuration, including the deployment manager.

 JCL for installing WebSphere Process Server into a deployment manager node

 JCL for augmenting a deployment manager node

 JCL for installing WebSphere Process Server into an empty node

 JCL for augmenting an empty node

 zWPSInstall.sh script

Use the zWPSInstall.sh script to modify a WebSphere Application Server profile either to install or uninstall WebSphere Process Server. Run the script on each node in your configuration, including the deployment manager.

Configuring the deployment manager with WebSphere Process Server:

The deployment manager manages the network deployment cell. All nodes in a network deployment configuration use DB2 database objects.

Before you begin

Before you configure the deployment manager node you must have completed the following tasks:

- Create a deployment manager node in WebSphere Application Server. See *Creating a network deployment cell*.
- Install WebSphere Process Server on the deployment manager node. See *“Installing WebSphere Process Server”* on page 1.
- Back up the data sets that contain the deployment manager configuration.
- Create a response file for a deployment manager with a DB2 database, either by using the sample files or the zPMT tool.

Procedure

1. Stop the server.

2. Increase the OMVS time limit to allow the product configuration script time to complete. In an MVS console enter the following command:

```
SETOMVS MAXCPUIME=86400
```

3. Access the USS command shell, then switch to the administrator user ID. For example:

```
su wsadmin
```

4. Change directory to the deployment manager bin directory:

```
cd /WebSphere/V7R0/DeploymentManager/bin
```

5. Run thezWPSCConfig.sh configuration script.

You can use JCL to run the configuration on the deployment manager node; see JCL for augmenting a deployment manager node for an example JCL script.

Alternatively, you can run the configuration script directly from USS with the absolute path of your edited response file. For example:

```
zWPSCConfig.sh -augment -response /working_directory_path/DmgrDB2.rsp
```

where *working_directory_path* is the location of the DmgrDB2.rsp file that you have edited.

6. Wait for the configuration to run.

When the script has finished running, review the messages that are written to the console. If the script has run successfully, no error messages are displayed and the informational messages state augmenting profile(s) complete.

7. Back up the data sets that contain the deployment manager configuration.

Results

The deployment manager is configured with WebSphere Process Server.

What to do next

You can now create the DB2 common database using one of the following methods:

- Run the supplied createDB.sh script to create a single common database in which all the database objects for the configuration are created. The createDB.sh script creates the database, then populates the database with the database objects that are required by features such as Business Process Choreographer, Common Event Infrastructure, and SCA. See “Creating DB2 database objects using the createDB.sh script” on page 73.
- Run the DDL scripts that were generated by the configuration script using SPUFI, DSNTEP2, or DBUtility.sh This method is more complex because you create the database using one DDL script, then later populate it by running other DDL scripts. This method does, however, enable you to specify that the database objects are created in multiple databases, instead of them all being created in the single common database. See “Creating the DB2 databases and storage groups using SPUFI, DSNTEP2, or DButility.sh” on page 75.

Configuring an empty node with WebSphere Process Server:

You must configure an empty WebSphere Application Server node with WebSphere Process Server for z/OS functions before you federate the empty node to the deployment manager.

Before you begin

Important: Do not federate the managed node to the deployment manager at this stage.

Before you configure the empty node you must have completed the following steps:

- Create an empty node using WebSphere Application Server. See Adding, managing, and removing nodes. Do not federate the managed node at this stage.
- Install WebSphere Process Server on the empty node.
- Back up the data sets that contain the deployment manager configuration.
- Create a response file for an empty node with a DB2 database. You can do this by either editing the sample files or using the zPMTtool..

Procedure

1. Run the zWPSConfig.sh configuration script.

You can use JCL to run the configuration on the empty node; see JCL for augmenting an empty node for an example JCL script.

Alternatively, you can run the configuration script directly from USS with the absolute path of your edited response file. For example:

```
zWPSConfig.sh -augment -response /working_directory_path/ManagedDB2.rsp
```

Where *working_directory_path* is the location of the ManagedDB2.rsp file that you have edited.

For more information about the command, see zWPSConfig.sh and zWESBConfig.sh scripts.

2. Wait for the configuration to run.
3. **Optional:** Back up the file system that contains the empty node configuration.

Results

The empty node has now been configured with WebSphere Process Server for z/OS functions.

What to do next

You can now create other empty nodes, in the same way, as required. Then, federate each of the empty nodes to the deployment manager cell; see “Federating the empty node into the deployment manager cell.”

Federating the empty node into the deployment manager cell:

When you federate the empty node into the deployment manager cell you associate the empty node with the deployment manager so that you can use the deployment manager to administer the node. You cannot use the empty node for processing work until you federate the empty node.

Before you begin

Before you federate an empty node to the deployment manager cell, you must perform the following task:

- Create one or more empty nodes.

Procedure

1. Start the deployment manager. See Starting a server from the MVS console.
2. Locate the JCL member BBOWMNAN on the system where your node is.
3. Submit the BBOWMNAN job and check that it completes with a return code of 0.
4. If you have chosen not to start the node agent automatically when you used the WebSphere Customization Tools (WCT) application to configure the empty managed node, you must start the node agent manually now. See Starting a server from the MVS console.
5. Back up the configuration file system data sets for the deployment manager and for any managed nodes.

Results

The empty node has now been federated into the deployment manager cell and you can administer the node using the deployment manager in the administrative console.

What to do next

You can now use the administrative console to create a cluster. See “Creating a cluster” on page 117 for more information.

Creating and configuring the databases

Before the WebSphere Process Server instance can function, the supporting database must be configured and the database objects defined.

About this task

The database to support a WebSphere Process Server instance may be implemented on either Derby or a DB2. A Derby database may only be used with a stand-alone server. A DB2 database may be used for either a stand-alone server or a network deployment environment. Refer to “Augmenting with WebSphere Process Server” on page 63 for information on deciding which database to use.

The Derby database will be created automatically when the stand-alone server is augmented. The DB2 database objects will need to be created as a separate exercise.

The DDL to create a DB2 database may be located in one of two places: - The default directories into which the DbDesignGenerator.sh tool generates DDL is <WAS home>/util/dbUtils Refer to “Creating the database design file using the database design tool” on page 57 for information about generating the DDL using the database design tool. DDL is also generated when BPM components are configured, either at augmentation or by using the various wizards which are available in the administration console. This DDL is located in the WebSphere Process Server profiles directory under <WAS home>/profiles/default/dbscripts

Choosing how to create your DB2 database

There are various tools which may be used to run the DDL to create the DB2 database. WebSphere Process Server provides the sample createDB.sh script, which may be run from the USS command environment to create the database objects.

Alternatively, DB2 tools such as SPUFI or DSNTEP2 which are run from a z/OS environment may be used to execute the DDL.

Choosing which tool to use

You may choose one tool over another based on experience and familiarity or personal preference. Your organisation may also implement standards or conventions for the tools used to create DB2 objects, particularly in a production environment.

Considerations for choosing the createDB.sh sample

- createDB.sh can create all your database objects in one simple execution of the tool, so it is a good choice if this is your first server implementation.
- createDB.sh runs the DDL generated by the DbDesigngenerator.sh tool.
- createDB.sh creates database objects according to a defined naming convention. However, it is provided as a sample script so you may customise it to suit your own naming organisation.
- createDB.sh automatically completes customisation of the CEL and SIB DDL.
- createDB.sh is run from a USS environment.
- createDB.sh does not produce an audit trail of the objects which it has created.

Considerations for choosing other tools

- The DDL files may need copying from the WebSphere Process Server file system into a partitioned dataset (PDS). The Ddl2Pds.sh tool may be used to copy the files across. Refer to Ddl2Pds.sh script for more information.
- There is no restriction on the naming or organisation conventions which apply to the database objects.
- The CEI DDL and the SIB DDL files will need customising before they can be run. (Note - you can use createDB.sh with the -RunSQL option to generate the CEI and SIB DDL without executing it. You can then use the DDL generated by createDB.sh.)
- Some tools may be run from a z/OS environment.
- The tools may produce an audit trail of the DB2 database commands which have been issued.

Creating DB2 database objects using the createDB.sh script

You can run the createDB.sh script to create the DB2 database and populate it with objects after you have configured for WebSphere Process Server. You can use the createDB.sh script for either a stand-alone configuration that uses DB2, or for a network deployment configuration. If you use createDB.sh, all the database objects for the configuration are created in a single database. If you want to use multiple databases, use a different method of creating the databases and database objects; for example use the supplied SPUFI or the DBUtility.sh script.

Before you begin

Before you run the createDB.sh script, you must complete the following steps:

- Decide on a naming convention for the databases.
- Decide on schema names and SQL IDs.
- For a stand-alone configuration: Create the stand-alone server for use with DB2; for a network deployment configuration: Create the deployment manager. Ensure that when you configured the nodes the dbDelayConfig parameter in the response file was set to true.

- Create the required buffer pools.
- Run the DbDesignGenerator.sh tool to generate the DDL to create the database tables for all the components you intend to configure. You must run the DbDesignGenerator.sh from the <WAS_HOME>/util/dbUtils directory, and generate the DDL into the default directory names. Note that you must create a network deployment design if you want to generate DB2 for z/OS DDL for the Process Choreographer and Process Choreographer Reporter, even if you are setting up the database for a stand alone server.

About this task

The createDB.sh script creates a database and populates it with all the DB2 database objects that are required by WebSphere Process Server, including the database objects used by SCA, Business Process Choreographer, and Common Event Infrastructure. In a sysplex environment where the database is not shared across the sysplex and the installation of WebSphere Process Server and the installation of DB2 are on different LPARs, you must run createDB.sh on the system that has the installation of DB2 rather than the one that has the installation of WebSphere Process Server.

Procedure

1. On the stand-alone server node or the deployment manager node, access the USS command shell, then switch to the administrator user ID. For example:


```
su wsadmin
```
2. Copy the sample createDB.sh file to your working directory. For example:


```
cp /usr/lpp/zWPS/V7R0/zos.config/samples/createDB.sh /u/work
```
3. Assign the appropriate permissions to the copy of the createDB.sh file:


```
chmod 755 createDB.sh
```
4. Customize the parameters in the copy of the createDB.sh file as required by your system. See the createDB.sh script topic for more information
5. Save the edited file.
6. Copy sibDropxx.sql & sibSchemaxx.sql to your working script directory. For example:


```
cp /usr/lpp/zWPS/V7R0/zos.config/samples/sibDropxx.sql /u/work
cp /usr/lpp/zWPS/V7R0/zos.config/samples/sibSchemaxx.sql /u/work
```
7. Set your WAS_HOME variable appropriately, For example:


```
export WAS_HOME=/WebSphere/V6S01Z1/AppServer
```
8. If necessary set your LIBPATH and STEPLIB variables in order to access DB2 code, For example,


```
export LIBPATH=/ZOS180/usr/lpp/db2910/lib:$PATH
export STEPLIB=SYS2.DB2.V910.SDSNEXIT:SYS2.DB2.V910.SDSNLOAD:SYS2.DB2.V910.SDSNLOAD:$STEPLIB
```
9. Run the customized createDB.sh script. Check the error.out file for any errors in the cdbtmp directory that is created by the createDB.sh script in your /u/work directory.
10. If you are creating a stand-alone configuration, verify the WebSphere Process Server installation:
 - a. Start the server.
 - b. Open the administrative console by opening a browser window and typing the URL of the server that you want to view. For example:


```
http://server_name.domain_name:port_number/admin
```
 - c. Log in to the administrative console.

- d. Verify you can see WebSphere Process Server on the Welcome page. You can click on this for more information.
- e. Navigate around the console to check the following:
 - The enterprise applications are started.
 - The messaging engines are started.
 - Test the data source connections.

If anything has failed to start you can look in the server job logs for "SEVERE" or "WARNING" messages that provide details about the failure.

Results

The database is created and populated with the database objects that are required by the your configuration.

What to do next

If you are creating a stand-alone configuration, you can now deploy applications to the server.

If you are creating a network deployment configuration, next, you must create one or more empty nodes to add to the deployment manager cell. See "Configuring an empty node with WebSphere Process Server" on page 70.

Creating the DB2 databases and storage groups using SPUFI, DSNTEP2, or DBUtility.sh

The zWPSConfig.sh configuration script generates Data Definition Language (DDL) scripts that you can use to create the DB2 database objects for the configuration. There are several tools that you can use to run the DDL scripts to create the database objects for your configuration.

Before you begin

Before you create the DB2 databases and storage groups, you must complete the following tasks:

- Create the server configuration: for a stand-alone server, see "Creating a stand-alone configuration" on page 66; for a deployment manager node, see "Configuring the deployment manager with WebSphere Process Server" on page 69.
- Make sure that the DDL has been generated for all the components you want configure the database with. You can generate the DDL by completing the following tasks:
 - Running the zWPSConfig.sh script script.
 - Creating and generating a deployment environment with the appropriate components configured.
 - Running the individual component configuration wizards.
 - Using the DbDesignGenerator.sh tool.

About this task

You can run the DDL scripts using DBUtility.sh, SPUFI, or DSNTEP2.

If you want to work in the USS environment, you can run the DDL scripts using the DBUtility.sh script, which is also supplied with WebSphere Process Server. The

createDB.sh script creates all the DB2 database objects in a single database. If you want to create the database objects across multiple databases but still want to work in the USS environment, you can run the DDL scripts using the DBUtility.sh script, or run createDB.sh several times specifying different components for each database name.

Important: After converting from ASCII to EBCDIC check that no SQL statements exceed 71 characters in length because this will lead to line truncation and invalid statements when copying to fixed width MVS data sets.

Procedure

1. Create the databases and storage groups.
2. Populate the databases using the generated DDL scripts. The location of the generated DDL scripts depends on how they were generated.

The default location for DDL generated by DbDesignGenerator.sh is in the directories under the following locations:

- <WAS_ROOT>/AppServer/util/dbUtils for a stand-alone configuration.
- <WAS_ROOT>/DeploymentManager/util/dbUtils for a network deployment configuration.

For DDL generated by other means, the DDL is in the directories under the following locations:

- <WAS_ROOT>/AppServer/profiles/default/dbScripts for a stand-alone configuration.
- <WAS_ROOT>/DeploymentManager/profiles/default/dbScripts for a network deployment configuration.

Where <WAS_ROOT> is the high level directory of your WebSphere Application Server configuration.

3. If you are running the DDL from a USS environment, assign the appropriate permissions to the copies of the files; for example:

```
chmod 755 createTable_AppScheduler.sql
```
4. Edit the values in the file to suit your needs. The database names, storage groups and schema names are customised by the product configuration process. Check the values in each file to make sure they match the values that you entered in the response file that provided input to the configuration script and are suitable for your data base.

Note: The files can be provided in ASCII format. If the tools that you use to view, edit, and run the scripts require the scripts to be in EBCDIC format, use the iconv command to convert the file to EBCDIC. For example:

```
iconv -t IBM-1047 -f IS08859-1 createTable_AppScheduler.sql >  
createTable_AppScheduler_EBCDIC.sql
```

If you have converted the file from ASCII format to EBCDIC but need to run the file in ASCII format, use iconv to convert the file back to ASCII. For example:

```
iconv -t IS08859-1 -f IBM-1047 createTable_AppScheduler_EBCDIC.sql >  
createTable_AppScheduler.sql
```

5. **Optional:** If you want to create database objects outside of the USS environment, for example via SPUFI or DSNTEP2, you can use the supplied Ddl2Pds.sh script to copy the customised DDL from USS to a partitioned dataset. For example, to copy the DDL for the WebSphere Process Server Common component enter a command similar to the following from the /usr/lpp/zWPS/V7R0/zos.config/samples directory:

```
./Ddl2Pds.sh -Source
/WebSphere/V7S05Z1/AppServer/profiles/default/dbscripts/CommonDB/DB2z0SV8/S5CELLDB -PDS HEALDR.DDL2PDS.TEST -Component
WPS
```

6. Run the customized scripts using the tool of your choice. For example:

SPUFI A utility that runs SQL scripts from z/OS. SPUFI uses EBCDIC input.

DSNTEP2

A sample dynamic SQL program provided with the DB2 for z/OS product.

DBUtility.sh

DBUtility.sh is a utility that is supplied with WebSphere Process Server for z/OS and installed in the installation file system. For example: /usr/lpp/zWESB/V7R0/bin/DBUtility.sh. You can use this utility to create the database and storage groups, as well as to run the SQL to create the database tables later, from USS. DBUtility.sh uses ASCII input. Here is an example of the syntax used with the DBUtility.sh script:

```
    /WebSphere/V7S03Z1/AppServer/profiles/default/bin/DBUtility.sh
createTable
-DdbStorageGroup=S3DBST0
-DdbSchemaName=S3CELL
-DsqlScriptName.default=createTable_AppScheduler.sql
-DsqlScriptPath.default=/WebSphere/V7S03Z1/AppServer/profiles/default/dbscripts/CommonDB/DB2z0SV8/S3CELLDB
/createTable_AppScheduler.sql
-DdbType=DB2UDBOS390_V8_1
-DdbName=S3CELLDB
-DprofileName=default
-DprofilePath=/WebSphere/V7S03Z1/AppServer/profiles/default
-DdbJDBCProperties=/wps/dbscripts/db2v8
-DdbConnectionLocation=DSN810PP
-DdbJDBCClasspath=/usr/lpp/db2810/db2810/jcc/classes
-DdbUserId=wsadmin
-DdbPassword=password
-DdbDelayConfig=false
-DdbCreateNew=false
-DdbHostName=wimvsp1.hursley.ibm.com
-DdbServerPort=448
>/tmp/output.out 2>>/tmp/error.out
```

7. Verify that the database, storage group and tables have been created successfully with no errors by inspecting the output.
8. If you are creating a stand-alone configuration, verify the WebSphere Process Server installation:
 - a. Start the server.
 - b. Open the administrative console by opening a browser window and typing the URL of the server that you want to view. For example:

```
http://server_name.domain_name:port_number/admin
```
 - c. Log in to the administrative console.
 - d. Verify you can see WebSphere Process Server on the Welcome page. You can click on this for more information.
 - e. Navigate around the console to check that the server has a status of started. Also check all the applications are started, and that the messaging engines are started. If anything has failed to start you can look in the server job logs for "SEVERE" or "WARNING" messages that provide detail about the failure.

Results

The DB2 databases and storage groups are created and populated with the necessary database objects, such as tables and indexes.

What to do next

If you are creating a stand-alone configuration, you can now deploy applications to the server.

If you are creating a network deployment configuration, next, you must create one or more empty nodes to add to the deployment manager cell. See “Configuring an empty node with WebSphere Process Server” on page 70.

Creating the messaging engine data stores

If the messaging engine data store has not already been created using `createDB.sh` script, use the `sibDDLGenerator` command to generate the DDL statements that the database administrator will use to create the tables for the messaging engine data store.

Before you begin

Before you create and run the SQL scripts:

- Create a cluster.
- Decide which SIBus components you will be creating in your network deployment configuration.

About this task

You must create and populate a messaging engine data source for each of the following components that you configure:

- SCA (System and Application Buses)
- Business Process Choreographer
- Common Event Infrastructure

You can create all the database objects in a single database, in an existing database, or in one or more new databases. Note that:

- Each bus has identical table names. Therefore, the tables for each bus should have a unique schema qualifier.
- Each bus has identical tablespace names. Therefore, a separate database should be used for each bus, or if a single database is used, the tablespace names should be edited to make them unique within the database.
- Note that if `createDB.sh` is used to create the bus databases, it amends the tablespace names to incorporate a 5-character schema qualifier. For example:
 - `OWNERTS => <schema>OWN`, e.g. `A6S1AOWN`
 - `OWNEROTS => <schema>OTS`, e.g. `A6S1AOTS`
 - `MAPTS => <schema>MAP`, e.g. `A6S1AMAP`
 - `LISTTS => <schema>LST`, e.g. `A6S1ALST`
 - `SIB000TS => <schema>00T`, e.g. `A6S1A00T`
 - `SIB000LS => <schema>00L`, e.g. `A6S1A00L`
 - `SIB001TS => <schema> 01T`, e.g. `A6S1A01T`
 - `SIB001LS => <schema> 01L`, e.g. `A6S1A01L`
 - `SIB002TS => <schema> 02T`, e.g. `A6S1A02T`
 - `SIB002LS => <schema> 02L`, e.g. `A6S1A02L`
 - `XACTSTS => <schema>XAT`, e.g. `A6S1AXAT`

- KEYTS => <schema>KEY, e.g. A6S1AKEY

Procedure

For each data source required, follow the instructions in the WebSphere Application Server information center to create and run the SQL scripts: Enabling your database administrator to create the data store tables. For example, to create an SIB data source for SCA, using a new database called S2SIBSCA use the following command::

```
sibDDLGenerator.sh -system db2 -version 8.1 -platform zos -schema
S2CELLS -user wsadmin -create -database S2SIBSCA -storagegroup
S2SIBST0 -buffer pool BP3 -statementend ";" > /u/hssd/SIBSCA.ddl
```

You can use the -buffer pool option (-buffer pool BP3 in the example) to set a non-default buffer pool when you are creating the DDL.

Results

The messaging engine data stores are created.

What to do next

Next, configure SCA, Business Process Choreographer, and Common Event Infrastructure:

- “Configuring SCA support for a server or cluster” on page 119
- Configuring Business Process Choreographer
- Configuring Common Event Infrastructure

Granting table privileges to the JCA authentication alias user ID

If the schema name you are using is not the same as the JCA authentication alias user ID you must grant a sub-set of DB2 privileges to the JCA authentication alias user ID.

About this task

The DDL for the Service Integration Bus already contains commented GRANT commands that you can use as a basis for granting access to the SIB tables. However, the other WebSphere Process Server for z/OS components do not supply any GRANT statements.

Use a schema name that is not the same as the JCA authentication alias to prevent the alias user ID having the power to drop tables. (The power to drop tables is implicitly granted to the creator, that is, the schema.) Note that it does not make sense to grant a privilege like DBADM to the JCA authentication alias user ID because DBADM also has the ability to DROP tables.

If you want the WebSphere Process Server to function while not allowing the alias user ID to have DROP capability, create some GRANT statements by copying the DDL and editing it to construct GRANT commands from the CREATE commands. Create GRANT commands like:

```
GRANT ALL PRIVILEGES ON TABLE
cell.tablename TO userid/sqlid
```

Where userid/sqlid is the JCA authentication alias user ID.

Setting the correct schema name for the SIBs

To ensure the SIB messaging engines can access the appropriate DB2 tables, set the correct schema name for the SIBs.

Before you begin

Before you start:

- Start the server..

About this task

Use the administrative console to change the schema names.

Procedure

1. Log in to the administrative console.
2. Navigate to **Service Integration** → **Buses**.
3. For each bus:
 - a. Select **Messaging engines**, then click the name that is displayed.
 - b. Click **Message store**.
 - c. Change the value of **Schema name** to the name used when creating the DB2 tables for this SIB.
 - d. Click **Apply**.
 - e. Save your configuration changes.
4. Log out of the administrative console.
5. Stop, then restart the server.
6. Look in the output of the Adjunct job log for successful SIB messaging engine startup messages. For example:

```
BB000222I: "BusName"  
CWSID0016I: Messaging engine MessagingEngineName is in state Started.
```

Results

The schema name used by the SIB messaging tables to access the DB2 tables is changed.

Adding the DB2 libraries to the Servant and Adjunct JCL

If your DB2 system does not run with SDSNEXIT, SDSNLOAD, and SDSNLOD2 in LNKLST, update your WebSphere Servant and Adjunct JCL so that the STEPLIB includes the customized DB2 SDSNEXIT, the SDSNLOAD, and the SDSNLOD2.

About this task

If you do not have the STEPLIB set correctly, you will get the problem “Failure in loading T2 native library db2jct2zos”, in the server systout, since the DB2 programs in SDSNLOD2 cannot be loaded. This error causes the SIBs to fail during initialization.

Backup the WebSphere Application Server configuration file system when you have completed the configuration.

Cleaning up the Derby JDBC resources

If you find any JDBC datasources defined for Derby JDBC providers you can delete them. In some circumstances SIB datasources are defined under a Derby JDBC Provider.

About this task

Perform the following steps to cleaning up the Derby JDBC resources

Procedure

1. Log in to the administrative console and navigate to **Resources** → **JDBC Providers**.
2. Set the scope to the cell level by removing any node or server name and clicking **Apply**.
3. Click any Derby JDBC Provider or Derby JDBC Provider (XA) that you find and then click **Data sources** on the right.
4. Delete any data sources with names that contain character strings related to WebSphere Process Server, for example WPS, BPE, SCA, SIB.
5. Save your configuration changes.
6. Navigate to **Resources** → **JDBC Providers** again, and change the scope to your Application Server node.
7. Check and delete any Derby JDBC providers for data sources that relate to WebSphere Process Server.
8. Save your configuration changes.
9. Navigate to **Resources** → **JDBC Providers**, change the scope to the node level and select your application server.
10. Check and delete any Derby JDBC providers for any data sources that relate to WebSphere Process Server.
11. Save your configuration changes.

What to do next

The next step is to verify the installation. See www.ibm.com/support/techdocs/atmsastr.nsf/WebIndex/WP101218 for more information.

Verifying the installation with DB2

When verifying an installation with a DB2 database, it is important to check the Servant and Adjunct job logs to see whether there are any error messages that might indicate problems accessing the data store.

Procedure

1. Ask your DB2 system administrator to check the authorities that have been granted to ensure that you have not granted more authority than necessary to any user ID. It can be tempting to grant DB2 SYSADM authority to the JCA authentication aliases in order to avoid possible problems with DB2 security during the configuration.
2. Ask your DB2 system administrator to check the storage group assignments and buffer pool usage. Incorrect storage group assignment and buffer pool usage might not show up as an error message in a log but might cause problems later. It is better to resolve such problems now rather than when the system has been handed over to people to use. For example, correcting storage groups and VCATs is not easy after the tables and indexes have been used.

3. Log in to the administrative console. See Starting and stopping the administrative console.
4. In the administrative console, check that all the applications are started, the messaging engines are started, and all the data sources can be accessed using the **Test Connection** option. If any application has failed to start, look in the Servant and Adjunct job logs for SEVERE or WARNING messages that provide detail about the failure.
 - If you see DB2 errors such as SQLCODE -204, in the administrative console, set the correct schema name or currentSQLID value in the custom properties section of the data sources. If the schema name is not the same as the user ID in the JCA authentication aliases, the SQL requests try to find tables qualified by the user ID in the JCA authentication alias.
 - If you see DB2 deadlock errors such as SQLCODE -913 Reason Code 00C90088, check the RRULOCK DB2 parameter is set to YES as described in the Tuning checklist section.

What to do next

If all the SIBs have initialized correctly, and you do not see any other errors related to opening JDBC connections, you can continue to customize your configuration of WebSphere Process Server.

ConsolidateJAASAuthAliases.py script

ConsolidateJAASAuthAliases.py is a wsadmin script that is used to consolidate the several JAAS authentication aliases defined for database access by the augment process.

Introduction

When a WebSphere Process Server for z/OS server accesses a secure database subsystem, one of the security mechanisms available to it involves the use of JAAS authentication aliases. A JAAS authentication alias specifies a user identifier and password that is provided when the database subsystem requests authentication credentials. The augment process defines a set of JAAS authentication aliases that are associated with the various data sources and service integration buses for use when they access the database. The aliases are also assigned to a number of WebSphere Relational Resource Adapter CMP Connection Factories.

A fully configured WebSphere Process Server system consists of the following resources and JAAS authentication aliases defined by the augment process:

Table 7. JAAS authentication aliases

Data Sources	JAAS Authentication Alias
ESBLoggerMediationDataSource	WPSDB_Auth_Alias
WBI_DataSource	WPSDB_Auth_Alias
event	<qualifier>/EventAuthDataAliasDB2
CEI ME data source	CEIME_<qualifier>_Auth_Alias
SCA System Bus ME data source	SCASYSME00_Auth_Alias
SCA Application Bus ME data source	SCAAPPME00_Auth_Alias
Business Process Choreographer data source	BPCDB_<qualifier>_Auth_Alias
Business Process Choreographer ME data source	BPCME_00_Auth_Alias

Table 7. JAAS authentication aliases (continued)

Data Sources	JAAS Authentication Alias
Business Process Choreographer reporting function source	BPCEDB_<qualifier>_Auth_Alias
Business Space data source	BSPACE_Auth_Alias
CMP Connection Factories	JAAS Authentication Alias
WBI_DataSource_CF	WPSDB_Auth_Alias (component-managed)
event_CF	<qualifier>/EventAuthDataAliasDB2 (component-managed)
CEI ME data source_CF	CEIME_<qualifier>_Auth_Alias (component-managed)
SCA System Bus ME data source_CF	SCASYSME00_Auth_Alias (component-managed)
SCA Application Bus ME data source_CF	SCAAPPME00_Auth_Alias (component-managed)
Business Process Choreographer data source_CF	BPCDB_<qualifier>_Auth_Alias (component-managed)
Business Process Choreographer ME data source_CF	BPCME_00_Auth_Alias (component-managed)
Business Process Choreographer reporting function source_CF	BPCEDB_<qualifier>_Auth_Alias (component-managed)
Business Space data source_CF	BSPACE_Auth_Alias (component-managed)
SIBuses	JAAS Authentication Alias
<qualifier>-CEI<cell>BUS	CEIME_<qualifier>_Auth_Alias
<qualifier>-SCA.SYSTEM.<cell>.Bus	SCASYSME00_Auth_Alias
<qualifier>-SCA.APPLICATION.<cell>.Bus	SCAAPPME00_Auth_Alias
<qualifier>-BPC.<cell>.Bus	BPCME_00_Auth_Alias

On z/OS all the various data repositories are defined to access the same z/OS database subsystem, for example DB2 for z/OS. In addition, authentication to this common database subsystem is carried out using the same user identifier and password. It would not be uncommon for many, if not all, of the JAAS authentication aliases defined by the augment process to be defined with the same user identifier and password.

Having a number of JAAS authentication aliases defined with the same user identifier and password parameters presents a number of concerns:

- The password for database access will not normally expire, but if it needs to be changed for some reason, it needs to be changed in all the JAAS authentication aliases.
- The administrative console panel for working with JAAS authentication aliases is more cluttered, which reduces usability.
- The names of the JAAS authentication aliases may not conform to local naming conventions.

Purpose

Optionally, you can run the ConsolidateJAASAuthAliases.py script to address these issues. The script is invoked by the WebSphere wsadmin utility to perform the following actions:

- Consolidate the various JAAS authentication aliases listed in Table 1 into a single entry
- Reassign all the resources that referenced the original aliases so that they use the new alias
- Delete the original aliases.

The result is a single JAAS authentication alias that is used to authenticate database access for all the resources created by the WebSphere Process Server for z/OS configuration process.

ConsolidateJAASAuthAliases script

The wsadmin Jython script can be used to consolidate the various JAAS authentication aliases created by WebSphere Process Server or WESB configuration into a single entry.

By default the location of the script is /usr/lpp/zWPS/V7R0/zos.config/samples.

Scope of the script

The script was originally developed for WebSphere Process Server or WESB for z/OS V6.1.0.1 running on WebSphere Application Server for z/OS V6.1.0.15. The script was tested against a stand-alone server and a network deployment cell consisting of the deployment manager node and a single application server node.

Invocation of the script

The script is provided as an argument to the WebSphere wsadmin tool. You can provide seven mandatory parameters and one optional parameter to the script. The following code shows the syntax for the wsadmin Jython script (split over several lines to improve clarity):

```
/AppServerRoot/bin/wsadmin.sh
-host host name
-port host port
-lang jython
-f ConsolidateJAASAuthAliases.py
JAAS authentication alias name
user ID
password
[scan mode]
```

Parameters

-host *host name*

The host address of the target server, or of the deployment manager for a network deployment cell.

-port *host port*

The SOAP port number of the target server.

-lang *jython*

The language the script is written in, Jython.

-f ConsolidateJAASAuthAliases.py

If the script is not located in the current directory, you must include the path in which the script is stored.

JAAS authentication alias name

The name of the new JAAS authentication alias to be created. This can be any name, but it is good practice to choose a descriptive name, for example WPSDBAccess.

user ID

The user identifier to be provided for authentication to the database subsystem.

password

The password to be provided for authentication to the database subsystem.

[scan mode]

An optional parameter. If this parameter is missing (that is, only three parameters are provided to the script) any changes made by the script are committed when the script completes processing. If any string is provided as an eighth parameter, the script reports all the changes that it would make, but they are rolled back when the script completes processing. Scan mode can be useful for assessing the scope of the changes that the script will effect.

The script provides a report of all the actions it has taken.

Script processing

Processing of the script consists of the following steps:

1. A new JAAS authentication alias is created, based on the parameters provided to the script.
2. All the JAAS authentication aliases of interest are identified. The script searches through the list of all aliases looking for alias names that match the following patterns:
 - Starts with "BPCDB_"
 - Starts with "BPCME_"
 - Starts with "CEIME_"
 - Starts with "SCAAPPME"
 - Starts with "SCASYSME"
 - Ends with "EventAuthDataAliasDB2ZOS"
 - Is equal to "WPSDB_Auth_Alias".
3. For each identified alias, all occurrences of it are replaced with the new alias in JDBC data source definitions. The exception to this is data sources with a container managed alias such as event_catalog. Container managed aliases are deprecated, so they are replaced with the new alias as a component managed alias.
4. For each identified alias, all occurrences of it are replaced with the new alias in SIBus messaging engine data store definitions.
5. For each identified alias, all occurrences of it are removed from WebSphere Relational Resource Adapter CMP connection factory definitions.
6. All of the identified aliases are removed from the WebSphere configuration.

If no fourth parameter is supplied to the script, the changes are committed. If any string is provided as a fourth parameter, the changes are backed out, although the script still reports the changes that it would have made.

Sample output

The following output shows a sample execution of the script:

```
/WebSphere/V7T5DM/DeploymentManager/bin: > ./wsadmin.sh -lang jython -host winmvsp1 -port 20502
-f /u/healdr/Jython/ConsolidateJAASAuthAliases.py DB2Alias wsadmin gadzooks
WASX7209I: Connected to process "dmgr" on node T5NodeDmgrMVP1 using SOAP connector; The type of
process is: DeploymentManager
WASX7303I: The following options are passed to the scripting environment and are available as
arguments that are stored in the argv variable: "[DB2Alias, wsadmin, gadzooks]"

ConsolidateJAASAuthAliases: Starting

Created JAAS alias: DB2Alias

Replacing alias reference in data source: ESLoggerMediationDataSource
WPSDB_Auth_Alias => DB2Alias

Replacing alias reference in data source: WBI_DataSource
WPSDB_Auth_Alias => DB2Alias

Removing alias reference from CMP connection factory: WBI_DataSource_CF
Component-managed WPSDB_Auth_Alias

Removing alias: WPSDB_Auth_Alias

Replacing alias reference in data source: event
T5Cell/T5DepEnv.AppTarget/EventAuthDataAliasDB2 => DB2Alias

Removing alias reference from CMP connection factory: event_CF
Component-managed T5Cell/T5DepEnv.AppTarget/EventAuthDataAliasDB2

Removing alias: T5Cell/T5DepEnv.AppTarget/EventAuthDataAliasDB2

Replacing alias reference in data source: CEI ME data source
CEIME_T5DepEnv.AppTarget_Auth_Alias => DB2Alias

Replacing alias reference in SIBus data store of ME: T5DepEnv.AppTarget.000-C
EI.T5Cell.BUS CEIME_T5DepEnv.AppTarget_Auth_Alias => DB2Alias

Removing alias reference from CMP connection factory: CEI ME data source_CF
Component-managed CEIME_T5DepEnv.AppTarget_Auth_Alias

Removing alias: CEIME_T5DepEnv.AppTarget_Auth_Alias

Replacing alias reference in data source: SCA System Bus ME data source
SCASYSME00_Auth_Alias => DB2Alias

Replacing alias reference in SIBus data store of ME:
T5DepEnv.AppTarget.000-SCA.SYSTEM.T5Cell.Bus SCASYSME00_Auth_Alias => DB2Alias

Removing alias reference from CMP connection factory: SCA System Bus ME data
source_CF Component-managed SCASYSME00_Auth_Alias

Removing alias: SCASYSME00_Auth_Alias

Replacing alias reference in data source: SCA Application Bus ME data source
SCAAPPME00_Auth_Alias => DB2Alias

Replacing alias reference in SIBus data store of ME:
T5DepEnv.AppTarget.000-SCA.APPLICATION.T5Cell.Bus SCAAPPME00_Auth_Alias => DB2Alias

Removing alias reference from CMP connection factory: SCA Application Bus ME
data source_CF Component-managed SCAAPPME00_Auth_Alias

Removing alias: SCAAPPME00_Auth_Alias

Replacing alias reference in data source: Business Process Choreographer data source
BPCDB_T5DepEnv.AppTarget_Auth_Alias => DB2Alias

Removing alias reference from CMP connection factory: Business Process Choreographer
data source_CF Component-managed BPCDB_T5DepEnv.AppTarget_Auth_Alias

Removing alias: BPCDB_T5DepEnv.AppTarget_Auth_Alias

Replacing alias reference in data source: Business Process Choreographer ME data source
BPCME_00_Auth_Alias => DB2Alias
```

```

Replacing alias reference in SIBus data store of ME:
  T5DepEnv.AppTarget.000-BPC.T5Cell.Bus BPCME_00_Auth_Alias => DB2Alias

Removing alias reference from CMP connection factory: Business Process Choreographer ME
  data source_CF Component-managed BPCME_00_Auth_Alias

Removing alias: BPCME_00_Auth_Alias

Replacing alias reference in data source: Business Process Choreographer reporting function
  source BPCEDB_T5DepEnv.AppTarget_Auth_Alias => DB2Alias

Removing alias reference from CMP connection factory: Business Process Choreographer
  reporting function source_CF Component-managed BPCEDB_T5DepEnv.AppTarget_Auth_Alias

Removing alias: BPCEDB_T5DepEnv.AppTarget_Auth_Alias

Replacing alias reference in data source: Business Space data source
  BSPACE_Auth_Alias => DB2Alias

Removing alias reference from CMP connection factory: Business Space data source_CF
  Component-managed BSPACE_Auth_Alias

Removing alias: BSPACE_Auth_Alias

Saving configuration

ConsolidateJAASAuthAliases: Completed

```

If the fourth scan mode parameter is provided, the Saving configuration message is replaced by the Running in scan mode, no updates committed message. For example:

```

./wsadmin.sh -lang jython -host winmvsp1 -port 20502
  -f /u/healdr/Jython/ConsolidateJAASAuthAliases.py DB2Alias wsadmin admn4was y

/WebSphere/V7T5DM/DeploymentManager/bin:>./wsadmin.sh -lang jython -host winmvsp 1 -port 20502
  -f /u/healdr/Jython/ConsolidateJAASAuthAliases.py DB2Alias wsadmin gadook y
WASX7209I: Connected to process "dmgr" on node T5NodeDmgrMVP1 using SOAP connector;
  The type of process is: DeploymentManager
WASX7303I: The following options are passed to the scripting environment and are available as
  arguments that are stored in the argv variable: "[DB2Alias, wsadmin, gadook, y]"

ConsolidateJAASAuthAliases: Starting

Created JAAS alias: DB2Alias

...

Removing alias: BSPACE_Auth_Alias

Running in scan mode, no updates committed

ConsolidateJAASAuthAliases: Completed

```

References

The following references provide more information about wsadmin and Jython scripting:

- WebSphere for z/OS V6.1 Information Centre
- WebSphere Process Server and ESB for z/OS Information Centre
- WebSphere z/OS V6.1 – WSADMIN Scripting Primer (with Jython)
- Using Jython Scripting Language With WSADMIN
- Introduction to Jython Part 1: Java programming made easier
- Introduction to Jython Part 2: Programming essentials

Configuring the Business Process Management components

You should now be in a position to start the WebSphere Process Server instance and have it connect to its databases. The next step is to make sure that all the required BPM components are configured.

The Business Process Management components may be configured in one of three ways:

1. Some components may be configured at augmentation time. This is particularly the case for a stand-alone server, for which most of the components may be configured at augmentation time.
2. For a network deployment cell, components may be configured by building and generating a Deployment Environment. Setting up a Deployment Environment will build server clusters and configure BPM components on them according to standard or custom topologies. Note that Deployment Environments are not available for a stand-alone server.
3. The components may be configured individually using the administration application. Various wizards are available in the administration console to assist with this task.

Determine which components you require in your runtime environment. Then configure either a Deployment Environment (network deployment only) or the administration application to configure those which were not set up by augmentation.

The deployment environment wizard versus configuring the components manually

There are two approaches to configuring Business Process Management components after augmentation is complete. You can either use the Deployment Environment wizard to create clusters and servers and deploy components across them according to a standard or a custom topology. Alternatively, you can use the various wizards available in the administration application to configure each of the components individually.

If you are configuring a stand-alone server, then you must configure the components individually. Note that many of the components may have been configured at augmentation.

If you are configuring a network deployment environment, then you may configure the components individually, or you may create and generate a Deployment Environment. The Deployment environment wizard can generate clusters and servers according to a number of pre-defined topologies, and configure multiple components across them all at the same time. If the pre-defined topologies do not meet your needs, you may also create a custom Deployment Environment topology.

For WebSphere Process Server for z/OS, you are recommended to start with the single cluster topology.

Considerations for building a Deployment Environment

- You are configuring a network deployment environment.
- You want to configure multiple components by stepping through a single wizard in the administration application

- You want to import the database design file to provide the values for database related resource definitions. Refer to “Creating the database design file using the database design tool” on page 57 for information on creating the database design file.
- You have a pre-defined Deployment Environment which you can import into the current environment and customize if necessary.

Considerations for configuring components individually

- You are configuring a stand-alone server or a network deployment environment.
- You do not want to configure all of the components which a Deployment Environment contains.
- You want to configure any clusters or servers upon which the components will be deployed before configuring any components.

For a network deployment environment, configuring a Deployment Environment consisting of a single cluster topology will meet most WebSphere Process Server for z/OS requirements.

Setting up deployment environments

Setting up deployment environments involves creating the deployment environment definition and then generating the environment.

About this task

You can create the deployment environment by using the Deployment Environment configuration wizard or through scripting using wsadmin. When you have finished creating the deployment environment, you can perform additional tasks to complete the deployment environment setup.

For information on choosing how to create your deployment environment, see Planning your deployment environment.

Related tasks

 [Planning your deployment environment](#)

Setting up your deployment environment involves many decisions that affect everything from the number of physical servers to the type of pattern you choose. Each decision will affect how you set up your deployment environment.

Creating deployment environments

Creating deployment environments involves creating the deployment environment definition and then generating the environment. You can create deployment environments using the Deployment Environment Configuration wizard or by using wsadmin.

The Deployment Environment Configuration wizard presents you with a series of panels from which you can configure the components and clusters that make up your deployment environment. When you finish entering information on the Deployment Environment Configuration wizard panels, and you click **Finish** (but not **Generate**), the result is a *deployment environment definition*. Only after you click **Generate** in the Deployment Environment Configuration wizard is the environment "configured". When you generate a deployment environment definition from the Deployment Environment Configuration wizard, the system configures all the clusters and components based on the data in the generated definition.

As well as being able to create deployment environments from the Deployment Environment Configuration wizard, you can also create deployment environments using wsadmin scripting. As with the Deployment Environment Configuration wizard, the wsadmin function for creating a deployment environment has two stages - first you create the deployment environment definition and then you generate the deployment environment from that definition.

Creating deployment environments using the Deployment Environment configuration wizard:

You can create the deployment environment by using the Deployment Environment configuration wizard.

Creating a deployment environment using a pattern:

After you select a deployment pattern, use the Deployment Environment Configuration wizard to create the deployment environment that is based on the pattern.

Before you begin

On the administrative console of the deployment manager navigate to **Servers > Deployment Environments**.

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

The procedure for creating deployment environments using the deployment environment wizard includes steps for selecting patterns and features, and therefore it is assumed that you have read and understood the information on patterns and features documented in the planning section.

It is assumed that you have installed the product and that you have created the deployment manager profile and the associated nodes.

Additionally, one of the steps in the Deployment Environment Configuration wizard includes importing a database design document. The database design document defines the database configuration for the selected deployment environment features. WebSphere Process Server includes a response-driven database design tool (DDT) that creates a database design document based on user inputs. The document then can be used by the DDT to create the database scripts and by the WebSphere Process Server deployment environment wizard to configure the databases used in the deployment environment. For more information on the DDT and for more information on database configuration in general, see *Configuring databases*.

About this task

This task describes the procedure for creating a deployment environment that is based on a specific pattern and uses the Deployment Environment Configuration wizard.

Note: If you make an error while you are working in the wizard, you can go back by clicking **Back**.

Procedure

1. From the administrative console, go to the Deployment Environments page by clicking **Servers** → **Deployment Environments** .
2. Launch the Deployment Environment Configuration wizard by clicking **New** on the Deployment Environments page.

- a. The **Create a deployment environment based on a pattern** option is selected. **Create a deployment environment based on a pattern** is the system default and it is the option described in this topic.

Deployment environment patterns capture commonly used business integration topologies. A pattern provides you with a template for the deployment environment that you are creating.

Note: Patterns have a direct relationship to the products supported by the configured deployment manager. WebSphere Process Server supports a specific set of patterns, with the *Remote messaging and remote support* pattern being the system default. If your deployment manager supports other products in addition to WebSphere Process Server, additional patterns may apply. Consult product-specific documentation for information on patterns as they apply to the products.

For information on the types patterns included with and supported by WebSphere Process Server, see *Topology types and deployment environment patterns* in the Planning section.

See *Custom deployment environment layout configuration* for information on using the Custom Deployment Topology Detail page to configure your custom deployment environment.

- b. Enter a unique name for the deployment environment in the **Deployment environment name** field.
- c. **Optional:** To view all of the configuration steps in the wizard, select **Detailed: Show all steps**.

If you choose **Fast path: Show only needed steps** the wizard displays only those pages that **do not** have assigned default values. Choose **Fast path: Show only needed steps** only if you are agreeable to accepting the system-provided default values for the deployment environment configuration.

This topic assumes that you have chosen **Detailed: Show all steps**

- d. Click **Next** to display the Deployment Environment Features page.
3. On the Deployment Environment Features page, select the feature for the deployment environment and click **Next** to either view a list of compatible features, or to view a list of deployment environment patterns. Features represent the runtime processing capabilities of your deployment environment.

The list of available features on the Deployment Environment Features page is based on the deployment manager profile. If your deployment manager profile has been augmented to include other products alongside WebSphere Process Server (for example, WebSphere Business Monitor or WebSphere Business Services Fabric), then the Deployment Environment Features page also lists these features.

If you have installed and configured a profile for WebSphere Process Server, then the Deployment Environment Features page includes the following:

- **WESB**, for WebSphere Enterprise Service Bus, which provides a deployment environment that supports mediations.

- **WPS**, for WebSphere Process Server, which provides a deployment environment that supports mediations, business processes, human tasks, and business rules.

The default value for the deployment environment feature matches the runtime capabilities of your deployment manager.

4. On the Select compatible deployment environment features page, select additional features as necessary and click **Next** to view the list of patterns associated with your primary and ancillary feature selections.

Note: The Select compatible deployment environment features page is displayed only if the deployment manager has been augmented with other business process management (BPM) features, such as WebSphere Business Monitor.

For an understanding of the relationship of features and compatible features, see the information on deployment environments in the Planning section.

5. On the Select the deployment environment pattern page, select the pattern for the selected deployment environment, then click **Next** to display the Select Nodes page.

The list of patterns that display on the Deployment Environment Patterns page is dynamic. This list is activated by, and dependent on, the following environment conditions and configuration decisions:

- The platform on which you have installed the software
- The selections that you have made on the Select the deployment environment feature page and the Select compatible deployment environment features page.

For a detailed description of the relationship of patterns to features, see Topology patterns and supported BPM product features

6. Optional: On the Select Nodes page, select the nodes that you want to include in this deployment environment, then click **Next** to display the Clusters page.

Select at least one node for the deployment environment. For high-availability and failover environments, select at least two nodes. For scalability, select all nodes.

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

7. Optional: On the Clusters page, assign the required number of cluster members on each node for each cluster *type* (Application Deployment Target, Messaging Infrastructure and Supporting Infrastructure) of the deployment environment.

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

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

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

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

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

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

a. **Optional:**

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

The information on each substep page is as follows:

Cluster

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

The value varies depending on the cluster type, as follows:

- Application Deployment Target
- Supporting Infrastructure
- Messaging Infrastructure

For information on the functional role provided by each cluster type, see *Topologies and deployment environment patterns*

Cluster Name

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

Cluster Short Name

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

Cluster Member Name

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

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

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

Cluster Member Short Name

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

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

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

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

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

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

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

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

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

- a. Configure a full URL path for all REST services by selecting either **https://** or **http://** from the **Protocol** list.
 - b. Enter a name in the **Host Name or Virtual Host in a Load-Balanced Environment** field.
 - c. In the **Port** field, enter the port that a client needs to communicate with the server or cluster.
 - d. In the table of REST services, if you want to modify the description of the REST service endpoint, overwrite the entry in the Description field. The other fields are read-only.
 - e. Click **Next** to go to the Import the database configuration page.
9. Optional: On the Import the database configuration page, click **Browse** to go to the database design document or enter the path to the database design document and then click **Next** to go to the Data sources page. The design document can be based on a database design that you created using the database design tool (DDT), or it can be the supplied design document based on the pattern and feature that you have selected.

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

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

On this page, define the database information for the components that are included in this deployment environment. Where possible, the wizard supplies default information for the parameters, but change those values to match the values that you defined when you planned the environment.

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

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

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

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

Oracle database considerations:

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

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

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

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

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

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

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

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

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

12. On the Business Process Choreographer page, set parameters for the Business Process Choreographer configuration and then click **Next** to display the System web applications page. On this page you specify the values for:

- Security roles
- Authentication aliases

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

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

The table contains the following control information.

Web Application

The name of the Web application.

Some of the components that are part of the deployment environment you are creating contain web applications. The **Web application** column can include the following components:

- Business Space

- Business Process Choreographer Explorer
- Business Rules Manager

Context Root

The current value of the context root for the component.

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

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

Description

The description of the Web application context root.

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

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

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

- a. Check for deferred configuration steps

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

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

15. Change the server short names to meet z/OS system naming conventions. You should limit your server short names to seven characters to allow the runtime to add an S or an A to a short name to designate servant regions or adjuncts.

To change the server short name for z/OS, perform the following steps:

- a. From the administrative console, navigate to **Servers** → **Server Types** → **WebSphere application servers**.
- b. Click the name of the server that you want to change.
- c. Type the new server short name in the **Short Name** area.

For example change the existing short name it from BBOS001 to WZxxZ1 where *xx* is the prefix of the cell, for example WZT5Z1.

For more information on server naming conventions for z/OS, see the WebSphere Application Server information center.

Results

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

What to do next

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

Related concepts

Deployment environments

A deployment environment is a collection of configured clusters, servers, and middleware that collaborates to provide an environment to host Service Component Architecture (SCA) interactions. For example, a deployment environment might include a host for message destinations, a processor of business events, and administrative programs.

Topologies and deployment environment patterns

There are different topology layouts. Before you install and configure WebSphere Process Server, review the information in this section. Understanding topology concepts will help you to make educated decisions on how to install and configure the product.

Related tasks

General steps for implementing a deployment environment

After designing a deployment environment, you will perform specific tasks to make that design a reality. Regardless which method you use to implement the deployment environment, you will perform the same general steps.

“Configuring deferred configurations for a deployment environment” on page 110
If you must defer the creation of your databases and tables, use the Deferred Configuration page. This page provides instructions on how to locate and run scripts for database and table creation.

“Creating the database design file using the database design tool” on page 57
Use the database design tool (DDT) to generate a design file that is used to create the database tables required by WebSphere Process Server. The DDT generates the design file from a user specified properties file or user interactive input. The resulting design file is then used by the DDT to create the database scripts that are used to create the database tables. Additionally, the design file can be used as input during profile creation and during deployment environment configuration to specify the database configuration properties.

“Configuring custom deployment environments” on page 102
Use the Custom Deployment Topology Detail page to configure your custom deployment environment.

Related information

z/OS application server naming conventions

Configuring Business Process Choreographer

Custom deployment environment layout configuration:

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

“Selecting Clusters and Single Servers to use with a deployment environment” on page 98 defines the clusters and servers that make up your deployment environment. Unlike the patterned deployment environments, where clusters are created for each function, in a custom deployment environment you add the clusters and servers that you need to perform functions.

“Defining the Deployment Environment Configuration” on page 99 describes the functions you configure for the clusters and servers. These functions are messaging, Common Event Infrastructure, or application support.

Before you complete the deployment environment configuration in the system by generating it, you can return to your configuration and make changes. After you generate the deployment environment configuration in the system, you can look at the current configuration. You can also add more servers and clusters, configure more functions, or you can remove servers and clusters from management by this deployment environment. You cannot undo a function configuration that you have already generated, and you cannot remove a server or cluster from the deployment environment definition that is still required by another server or cluster in your deployment environment.

Requirements for all custom deployment environments

A custom deployment environment layout includes these restrictions:

- After you complete a configuration by generating the deployment environment, the associated controls become checked and disabled. This means you cannot undo the configuration.
- After you generate the deployment environment, if a control is not checked and disabled for a component, you must configure the associated functions in the following order: configure the associated messaging engine, then configure the Common Event Infrastructure (CEI), then the application support (described later in this topic).
- The configurations that exist on a system override the topology layout configuration. Thus, exporting a custom topology reflects the actual configuration of the servers involved in the topology.

The Topology Layout page in the administrative console has four sections that you must configure for a custom topology:

- Select Cluster and Single Servers
- Messaging
- Common Event Infrastructure
- Components

The following sections include other requirements for completing a custom topology layout configuration.

Selecting Clusters and Single Servers to use with a deployment environment

Use the Select Cluster and Single Servers section of the Topology Layout page to manage the clusters and servers within the deployment environment and define which functions they provide.

The Select Cluster and Single Servers section of the Topology Layout page includes a list of available clusters and servers, which you configure as part of the deployment environment. You assign clusters and servers to collaborative units in the function configuration. Each collaborative unit represents a group of clusters and servers that provides, as a whole, a function in the deployment environment. You can remove clusters or servers from the deployment environment. However, you can remove only clusters or servers that are no longer needed by other clusters or servers in the configuration.

Defining the Deployment Environment Configuration

Use the Specify the Deployment Environment Configuration section of the Topology Layout page to define which clusters or servers participate in specific functions for the deployment environment.

Messaging

Note: Partitioned messaging engines are not supported.

You use the fields in the Messaging tab to configure the messaging destination for selected targets. Each table represents one collaborative unit, and the Messaging section can include multiple tables. You must select only one target (Cluster/Server) for the option of local configuration for each unit, and all other targets in this unit assume the remote destination. When applications send messages to targets with a remote destination configuration, the system routes the messages to the local target for their unit.

The messaging configuration applies to the Service Component Architecture (SCA), the CEI, and the Business Process Choreographer system buses.

To prevent conflicts with the local destinations in your topology configuration, the following rules apply:

- The SCA system bus messaging engine configuration determines the local and remote destination locations. The SCA application, the CEI and the Business Process Choreographer bus configurations follow the SCA system bus configuration.
- If you locate the messaging engines for other buses on different targets in a unit, then the other targets in that unit assume the remote destination role. If the CEI or Business Process Choreographer buses have different configurations, an information message indicates that the messaging engine for a given bus is not located on the same target as the SCA messaging engine.
- If you try to add a target that already has a remote or local destination configured that conflicts with the current bus settings of a given unit, the system generates an error message.

Common Event Infrastructure

You configure CEI on the CEI tab, like Messaging. The CEI can have multiple tables, each representing a unit. In each table you select one CEI cluster or server (Cluster/Server column) that acts as the server by selecting the **Server** radio button. All targets that are not configured as a server assume the destination role. On the corresponding targets the event Infrastructure emitter factory Java Naming and Directory Interface (JNDI) name is configured so that Common Base Events that are emitted on this target are sent to the server in their respective collaborative unit.

Application support

The Application Support tab lists all of the components that you can configure for a given deployment target. You configure component functions in a related collaborative unit. For example, you configure a Business Process Choreographer event collector in a unit to collect the Common Base Events that are emitted by the Business Process Choreographer container that is configured in the same unit. Each component configuration has requirements and dependencies on other component configurations. Dependencies are represented by cleared and disabled controls. To enable them, you must configure dependent controls first.

Note: Dependent controls are configured on either on the Messaging or the CEI tab.

Table 8 describes the relationships between the components.

Table 8. Deployment environment component relationships

Component	Purpose	Related Component	Considerations
Service Component Architecture (SCA)	<p>Configures the deployment target for SCA application support.</p> <p>The SCA system and application bus members are configured locally if the corresponding messaging configuration is local; otherwise, they are configured remotely with the remote destination as specified in the corresponding messaging unit.</p>	Messaging	SCA configuration is not available if you have not configured the deployment target for messaging.
Business Process Choreographer container	<p>Configures the deployment target for both business flow and human task support.</p> <p>The configuration follows the SCA configuration for setting up the Business Process Choreographer system bus.</p>	<p>Messaging</p> <p>Service Component Architecture</p> <p>Business Process Choreographer Explorer</p>	<p>Business Process Choreographer configuration is not available if the deployment target has not been configured for messaging or if it has not been configured for Service Component Architecture support.</p> <p>One collaborative unit supports one Business Process Choreographer configuration. Add as many units as you need on the Application Support tab.</p> <p>To manage a container, consider configuring Business Process Choreographer Explorer.</p>

Table 8. Deployment environment component relationships (continued)

Component	Purpose	Related Component	Considerations
<p>Business Process Choreographer Explorer</p>	<p>Configures Business Process Choreographer Explorer on the selected deployment target.</p> <p>Business Process Choreographer Explorer is a Web application that manages the Business Process Choreographer container that is configured in the same collaborative unit.</p> <p>It includes an optional reporting function (Business Process Choreographer Explorer reporting) which was previously known as the <i>Business Process Choreographer Observer</i>.</p>	<p>Business Process Choreographer container</p>	<p>The Business Process Choreographer Explorer configuration is available after you have selected a Business Process Choreographer container configuration in the same collaborative unit.</p> <p>You must configure the deployment target for Web application support</p> <p>You can configure as many instances of Business Process Choreographer Explorer on a deployment target as you want. Add the deployment target to the collaborative units with a configured container and check the Business Process Choreographer Explorer configuration control.</p>

Table 8. Deployment environment component relationships (continued)

Component	Purpose	Related Component	Considerations
Business Process Choreographer event collector	<p>Configures the Business Process Choreographer event collector on the selected deployment target.</p> <p>The Business Process Choreographer event collector gathers Common Base Events that are emitted from the Business Process Choreographer container that is configured in the same collaborative unit. Statistical information about the observed container is recorded in a database.</p>	<p>Business Process Choreographer container</p> <p>Common Event Infrastructure</p>	<p>Configure first the Common Event Infrastructure server on the same deployment target that you plan to use for the Business Process Choreographer event collector. The Business Process Choreographer event collector is available only after you configure the Business Process Choreographer container in the same collaborative unit.</p> <p>If you are not sure whether you need to observe a given Business Process Choreographer container, you can configure this function later.</p>
Business Rules Manager	<p>Configures the Business Rules Manager on the selected deployment target.</p> <p>The Business Rules Manager allows you to configure business rules that determine business process behavior.</p>	Service Component Architecture	<p>The Business Rules Manager configuration control is available after you configure SCA support on the same deployment target.</p> <p>You can configure only one Business Rules Manager for a deployment environment.</p> <p>You might need to configure only one Business Rules Manager in your system because one instance can manage the business rules configuration of the entire cell.</p>

Configuring custom deployment environments:

Use the Custom Deployment Topology Detail page to configure your custom deployment environment.

Before you begin

- Verify that deployment environments exist on this deployment manager.

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

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

Restrictions:

- The configurations that exist on a system take precedence over the deployment environment configuration. Thus, exporting a custom deployment environment reflects the actual configuration of the servers involved in the deployment environment.
- You need to configure the messaging units before configuring the component units. If the check box is unavailable, then you have not yet configured messaging support.

About this task

For a custom deployment environment, you can decide how to configure each function according to your needs. Configure each function to either clusters or single servers. There are three major areas in configuring a custom deployment environment topology:

- Messaging, which supports component internal communication.
- Common Event Infrastructure, which unifies event and monitoring functionality.
- Application Support, which supports business integration service components such as business processes and human tasks.

For more information see “Overview of custom deployment environment layout configuration.”

Procedure

1. In **Select Clusters and Servers for use with this Deployment Environment**, select a cluster or server from the list.
2. Click **Add**. The cluster or single server will be added to the table below.
3. Repeat steps 1 and 2 until you have selected all the clusters and servers you need for this deployment environment.
4. Select the **Messaging** tab.
 - a. Decide how many independent messaging units you need for your deployment environment and add that number by clicking **Add New Unit**. The system names each unit Messaging Unit *x*, where *x* is the number of the unit.
 - b. Assign clusters and servers from the table created in step 2 to each unit. Select the cluster or server to add to the unit and then choose the unit from **Add selected to unit**.
 - c. Decide which deployment target in each unit is to host local messaging support and configure the local messaging host by clicking **Local Bus Member** on the row that defines that deployment target in the unit.

All other clusters or servers are automatically configured for remote messaging destinations.

5. Click on the **Common Events Infrastructure** tab.
 - a. Decide how many independent Common Events Infrastructure units you need for your deployment environment and add that number by clicking **Add New Unit**.

The system names each unit Common Event Infrastructure Unit x , where x is the number of the unit.
 - b. Assign clusters and servers from the table created in step 2 on page 103 to each unit.

Select the cluster or server to add to the unit and then choose the unit from **Add selected to unit**.
 - c. Decide which deployment target in each unit is to host the Common Event Infrastructure server and configure the Common Event Infrastructure server host by clicking **Server** on the row that defines that deployment target in the unit.

All other clusters or servers are automatically configured for remote Common Event Infrastructure destinations.
6. Click on the **Application Support** tab. This tab shows all the components that can be configured for a given deployment target.

Restriction: You must complete the messaging units for each component before you can configure the component in this section. For example, if the check box is unavailable for Service Component Architecture, then the associated messaging units have not been configured. See "Overview of custom deployment environment layout configuration" for additional restrictions.

- a. Decide how many independent Application Support units you need for your deployment environment and add that number by clicking **Add New Unit**.

The number of units you need depends on how many Business Process Choreographer containers you need. If you do not need Business Process Choreographer containers a single unit will be sufficient for Service Component Architecture applications.

The system names each unit Application Support Unit x , where x is the number of the unit.
- b. Assign clusters and servers from the table created in step 2 on page 103 to each unit.

Select the cluster or server to add to the unit and then choose the unit from **Add selected to unit**.
- c. In a unit, select what cluster or server belongs to each component for your deployment environment.
- d. Repeat steps 6b and 6c until you configure all the components in each unit you need for your deployment environment.

What to do next

After completing or making edits to an existing deployment environment, the Custom Deployment Environment Configuration wizard opens. You can review the information and make any necessary changes.

Related concepts

Deployment environments

A deployment environment is a collection of configured clusters, servers, and middleware that collaborates to provide an environment to host Service Component Architecture (SCA) interactions. For example, a deployment environment might include a host for message destinations, a processor of business events, and administrative programs.

Topologies and deployment environment patterns

There are different topology layouts. Before you install and configure WebSphere Process Server, review the information in this section. Understanding topology concepts will help you to make educated decisions on how to install and configure the product.

Creating deployment environments using the command line:

You can use `wsadmin` to create a deployment environment. The `createDeploymentEnvDef` and `generateDeploymentEnv` provide a command-line equivalent to creating the deployment environment using the deployment environment wizard.

Add nodes to a deployment environment definition using the command line:

You can add nodes to a deployment environment definition using the `wsadmin.sh` command.

Before you begin

The task assumes that the node has been federated to the deployment manager.

This command to add a node to the deployment environment definition will fail if the topology is already configured.

You must be at the deployment manager to which you are adding nodes.

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

About this task

This task adds a federated node to a deployment environment definition and uses the `wsadmin.sh` command.

Procedure

1. Open a command window.

The `wsadmin.sh` command can be found at either the `<WPS>/profiles/<dmgr profile>/bin` directory, or the `<WPS>/bin` directory.

2. At the command prompt, enter the `wsadmin.sh` command to enter the `wsadmin.sh` environment.
3. Enter the `addNodeToDeploymentEnvDef` command to add the node to the deployment environment definition.

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

Example

This example adds a node (**MyNode**) to deployment environment definition (**myDepEnv**) with administrative security enabled:

Attention: If you are adding a node to a single cluster topology pattern, the value for `-toplogyRole` must be set to **ADT**. Deployment environment topology patterns are specified when you create the deployment environment using either the `createDeploymentEnvDef` command or the Deployment Environment Configuration wizard.

```
wsadmin.sh -connType SOAP -host myDmgr -port 8879 -user dmgrAdmin -password dmgrPass  
> $AdminTask addNodeToDeploymentEnvDef {-toplogyName myDepEnv -nodeRuntime WPS  
-toplogyRole Messaging -nodeName MyNode -serverCount 3}
```

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

Related reference



`addNodeToDeploymentEnvDef` command

Use the `addNodeToDeploymentEnvDef` command to add a node to an existing deployment environment definition.

Generating deployment environments using the command line:

You can generate deployment environments using the `wsadmin.sh` interface. This capability allows you to configure multiple deployment environments unattended on a deployment manager using a script.

Before you begin

You must enter the commands on the deployment manager on which you are configuring deployment environments.

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

About this task

After you have imported or created deployment environments on a deployment manager, you can configure the deployment environments using the `generateDeploymentEnv` command.

Procedure

1. Enter the `wsadmin.sh` environment.
2. Enter the `generateDeploymentEnv` command for each topology you are configuring.
3. To define the cluster short names, perform the following steps:
 - a. Log in to the administrative console and navigate to **Servers > Clusters > WebSphere application server clusters**.
 - b. Click the name of the cluster that you want to change.

- c. Type the new cluster short name in the Short Name area. For example change it from BBOC001 (which is usually the default value) to WCLxx where xx is the prefix of the cell, for example WCLT5
4. To define the server short names, perform the following steps:
 - a. Log in to the administrative console and navigate to **Servers > Server Types > WebSphere application servers**.
 - b. Click the name of the server that you want to change.
 - c. Type the new server short name in the Short Name area. For example change it from BBOS001 to WZxxZ1 where xx is the prefix of the cell, for example WZT5Z1.

See z/OS application server naming conventions for more information about defining the server short names.

Example

The following command configures the eastEnvironment and westEnvironment topologies on host myDmgr.

```
wsadmin.sh -connType SOAP -host myDmgr -port 8879
> $AdminTask generateDeploymentEnv {-topologyName eastTopology}
> $AdminTask generateDeploymentEnv {-topologyName westTopology}
> $AdminConfig save
```

Note: If administrative security is enabled, you are prompted for a user ID and password after the system processes the wsadmin.sh command.


What to do next

Save the configured deployment environments. From the command line, enter \$AdminConfig save.

Related information

generateDeploymentEnvFromDef command

Importing deployment environment definitions using the command line

 Managing node agents

Validate the deployment environment definition from the command line:

You can validate the deployment environment definition using the wsadmin.sh command.

Before you begin

The task assumes that the node has been federated to the deployment manager.

You must be at the deployment manager to which you are validating the deployment environment definition.

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

About this task

This task validates the deployment environment definition and uses the `wsadmin.sh` command.

Procedure

1. Open a command window.
The `wsadmin.sh` command can be found at either the `<WPS>/profiles/<dmgr profile>/bin` directory, or the `<WPS>/bin` directory.
2. At the command prompt, enter the `wsadmin.sh` command to enter the `wsadmin.sh` environment.
3. Enter the `validateDeploymentEnvDef` command to validate the deployment environment definition.

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

Example

This example validates the deployment environment definition (**myDepEnv**) with administrative security enabled:

```
wsadmin.sh -connType SOAP -host myDmgr -port 8879 -user dmgrAdmin -password -dmgrPass  
> $AdminTask validateDeploymentEnvDef { -topologyName topOne}
```

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

Editing deployment environment settings

You can edit and modify the deployment environment settings

Configuring host aliases:

Configure the IBM HTTP server or a server of your choice to allow communication between managed nodes and the deployment manager.

Before you begin

Create and configure a deployment manager and associated nodes.

About this task

The managed nodes and the deployment manager must be able to communicate with each other, so the host name alias for each node in the deployment target cluster must be visible to the deployment manager. The host name alias consists of the DNS host name and port number. You use this alias as part of a URL to access applications when they are running on the deployment target.

Note: This procedure uses two application cluster members that are referred to as `AppCluster_member1` and `AppCluster_member2`. Substitute your server names in the instructions.

Procedure

1. From the administrative console, navigate to **Servers** → **Server Types** → **WebSphere application servers** → *AppCluster_member1*.
2. Click the name.

3. Under the Communications heading, expand **Ports** and note the port value listed for *WC_defaulthost*. You will need to use it later.
4. Repeat steps 1 on page 108 through 3, for every cluster member. Repeat this for each additional application cluster member.
When you are finished, you will have a list of the cluster members and the port numbers for their default host.
5. From the administrative console, navigate to **Environment** → **Virtual Hosts** → **default_host**.
6. Under **Additional Properties**, click **Host Aliases**.
7. If an entry for the correct combination of host name and port value for cluster members is not displayed, add the missing entries to the list.
8. If you added new entries to the list, click **Save** and then **Synchronize**.

What to do next

Verify your installation by installing a test application.

Configuring a data source for your deployment environment:

Configure your business integration data source for the first time using the Database Provider Configuration page.

Before you begin

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

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

About this task

Use the Data Sources page to configure the collection of all the data sources that are needed in your deployment environment.

The component that needs the data source determines all required fields based on the **Database Provider** selected, and these fields must be completed. The component completes the rest of the fields with default values. You may either keep the default values or change them to meet your needs. In most cases, the component determines the **Scope** value.

You can configure a business integration data source only once. After you configure the data source and save it, some text boxes will be unavailable and you cannot change the values. All other text boxes on the page can be edited.

Procedure

1. In the Data Sources page, select check box next to the data source to configure.
2. Click **Edit Provider** to edit additional data source fields that are not shown on this page.

Note: Alternatively, you can just click the name of the data source in the **Data Source** column.

3. Enter the information. For a list of supported database types, see “Database specifications.”
4. Click **Apply** or **OK** to save your changes.

Related information

Configuring databases
 Common database specifications

Configuring authentication aliases for a deployment environment:

From one administrative console page, you can review or edit all your authentication aliases.

Before you begin

- Verify that deployment environments exist on this deployment manager.

Navigate to the administrative console of a deployment manager **Servers** → **Deployment Environments** → *deployment_environment_name* → **Related Items** → **Authentication Aliases**.

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

About this task

From this consolidated list of authentication aliases, you can:

- Review all the aliases for a given deployment environment
- Access the authentication configuration page through the *Alias_name* link

The **Reset** button resets the selected rows to the currently configured values. Click *Alias_name* to access the authentication configuration page where you make your changes.

Procedure

1. Select the row you want to change.
2. Do one of the following:

Option	Description
To edit the row	Click <i>Alias_name</i> .
To reset the row	Click Reset .

Editing a row takes you to the authentication configuration page where you make your changes.

3. Click **OK** or **Apply** to save any changes.

Related information

Authentication

Configuring deferred configurations for a deployment environment:

If you must defer the creation of your databases and tables, use the Deferred Configuration page. This page provides instructions on how to locate and run scripts for database and table creation.

Before you begin

- Verify that deployment environments exist on this deployment manager.

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

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

About this task

Use this procedure when you must create database tables or schemas separately from configuring a deployment environment.

The Deferred Configuration page shows the configuration steps needed to correctly configure your topology's databases. In most cases this page shows:

- Script location
- Instructions on how to run the scripts

Procedure

1. Perform the instructions provided in the Deferred Configuration page.
2. When you finish, click **Configuration Done**.

What to do next

A text box shows by whom and when the deferred configuration was performed last. The displayed instructions remain on this page for future reference.

Related tasks

“Creating a deployment environment using a pattern” on page 90

After you select a deployment pattern, use the Deployment Environment Configuration wizard to create the deployment environment that is based on the pattern.

Verifying your deployment environment

Before moving your production applications to the new environment, you must test to make sure that all of the components operate correctly.

Before you begin

Complete the implementation of your deployment environment as described in “Implementing a deployment environment.”

1. Install the software
2. Configure a node to host a deployment manager
3. Configure nodes
4. Federate nodes to the deployment manager
5. Cluster nodes together to provide function for the deployment environment

About this task

How you verify the deployment environment depends on whether the environment you implemented is an IBM-supplied deployment environment or a

custom deployment environment. You can manage IBM-supplied deployment environments from a single panel in the administrative console. You must create and manage custom deployment environments manually in the administrative console.

Procedure

1. Identify the type of deployment environment you are verifying.
You should already have this information based on your original plans.
2. Start the deployment environment.

Type of deployment environment	How to start
IBM-Supplied pattern	Start from System administration > Deployment environments > Deployment environment configuration as described in "Starting and stopping deployment environments."
Custom	Start from Servers > Clusters as described in "Verifying a custom deployment environment starts." Note: You must start all the servers and clusters defined in the deployment environment.

3. Install the test application.
4. Configure the test application for routing.
5. Start the test application.
6. Run the test application and verify those results.

What to do next

Install your production applications.

Verifying the application deployment target cluster starts:

To verify that the application deployment target cluster can start, you must start all the clusters in your deployment environment. This is an example for a three cluster deployment environment.

Before you begin

You need to create and configure the clusters for the messaging engines, Common Event Infrastructure (CEI) event server application and the application deployment target.

About this task

To verify that the application deployment cluster can start, you will start each cluster in turn.

Notes:

- This description assumes that you configured three clusters in the topology named MECluster, SupportCluster, and AppCluster. Substitute the actual cluster names and repeat the appropriate steps for any additional clusters in your deployment environment.

Procedure

1. From the administrative console on the deployment manager, expand **Servers**, then select **Clusters**.
2. Start the clusters.
 - a. Select the check box beside **MECluster**.
 - b. Select **Start**, and wait for the MECluster to start as shown by a green arrow.
 - c. Select the check box beside **SupportCluster**.
 - d. Select **Start**, and wait for the SupportCluster to start as shown by another green arrow.
 - e. Select the check box beside **AppCluster**.
 - f. Select **Start**, and wait for the AppCluster to start as shown by another green arrow.
3. Click the messaging buses.
 - a. Wait until all the clusters start.
 - b. Click **Service Integration** → **Buses**
 - c. Verify the messaging engine is running for each bus.
 - 1) Select the bus name.
 - 2) Click **Local Topology** to display the bus topology.
 - 3) Expand the bus until you see the status of the messaging engines.
4. Check the cluster members' job logs for controllers, servants, and adjuncts. Make sure that they have no errors, and look for the line `Server AppCluster_member1` is open for e-business or `Server AppCluster_member2` is open for e-business indicating that the cluster started successfully. Correct any errors you find before continuing.

What to do next

After correcting any errors, you configure the host aliases.

Note: After correcting configuration errors, you must stop the cluster and restart it for the configuration changes to take effect.

Troubleshooting tip: When examining the log you may see a message that states that a messaging engine failed to start because it could not find a certain bus. Restarting the clusters eliminates this message.

Installing the test application:

Install the test application to begin the process of verifying your deployment environment.

Before you begin

- You need to create and install your completed deployment environment.
- Log in to the deployment manager administrative console.

About this task

Use the application provided for you with WebSphere Process Server called BPCIVTApp (Business Process Choreographer Installation Verification Test) to verify that you installed and configured your WebSphere Process Server environment correctly. First you must install the application.

For more information about installing this application, see “Verifying that Business Process Choreographer works”. For more information about installing applications from the administrative console, see “Installing application files with the console.”

Note: If you have not enabled business processes and human tasks, you cannot use BPCIVTApp to test your deployment environment. In this case, you must install and run a Service Component Architecture application that uses business rules and selectors to exercise your deployment environment. Change the process to test the deployment environment to fit your application.

Procedure

1. From the administrative console, select **Applications** → **New Application** → **New Enterprise Application**.
2. Make sure that **Local file system** is selected, and then browse for the file `bpcivt.ear`. It will be in the `install_root/installableApps` directory.
3. Select the file `bpcivt.ear`, then select **Open**.
4. These steps assume you will use the default configurations. Select **Next** on the subsequent panels until you reach the Summary page. During these steps you will be selecting various options and mapping the module to the servers as described in other topics. For testing purposes, map this module to the application deployment target cluster.

Note: You will not have to map the module to the application target cluster on a stand-alone server.

5. Select **Finish**.
6. Select **Save**, then **Synchronize**.

What to do next

Configuring the test application for routing:

Use this procedure to configure your test application for routing.

Before you begin

You need to install your test application.

About this task

You first configure the application and then generate the plug-in configuration files.

Note: The description assumes a cluster named `AppCluster` and a Web server named `Webserver1`. If your test application uses human tasks or business processes, make sure you have already configured Business Process Choreographer on your application cluster.

For information on managing modules, module settings and mapping modules, see the WebSphere Application Server information center.

Procedure

1. Configure the application (or applications) that you will run to identify the Web server and the deployment target to the application, as follows.

- a. From the administrative console, select **Applications** → **Application Types** → **WebSphere enterprise applications**.
 - b. Select the name of the application.
 - c. Select **Manage modules**.
On this panel, each Module must map to one or more targets, identified under Server.
 - d. From the choices listed under Clusters and servers, select *Webserver1* (the Web server you configured previously) and *AppCluster* (the application deployment target).
 - e. Select **Apply**, then select **OK**.
 - f. Repeat steps 1d through 1e until you have configured all Web servers and deployment targets for your deployment environment.
 - g. Select **Save**, then **Synchronize**.
2. Generate the plug-in configuration file.
 - a. From the administrative console, select **Servers** → **Server Types** → **Web servers**.
 - b. Select the check box next to the name *Webserver1*.
 - c. Select **Generate plug-in**. A plug-in configuration file is created, as indicated by the message in the top of the window.
 - d. Repeat steps 2b and 2c as many times as needed for your deployment environment.

What to do next

Stop and restart the deployment manager and node agent. Next start the test application.

Starting the test application:

Use this procedure to start your test application to test your implementation.

Before you begin

You need to install and configure the test application for routing.

About this task

You start your test application from the administrative console.

Procedure

1. From the administrative console, select **Applications** → **Application Types** → **WebSphere enterprise applications**.
2. Select the check box next to the application name and select **Start**. Wait until a green arrow appears, indicating that the application has started successfully.

What to do next

After you start the test application, run this application.

Note: If the application does not start correctly, refer to the log files to find error messages indicating the problem.

Running the test application:

Use this procedure to run your test application to determine if your deployment environment is operating correctly.

Before you begin

You need to start your test application.

About this task

Successful execution of this application shows that your deployment environment is operating correctly. Follow the same procedure on the other member of the application deployment target cluster to make sure that it also functions correctly.

Procedure

1. In a browser window, enter a URL in the following form: `http://hostname:portnumber/testapp` where *hostname* is the fully qualified DNS name or IP address of the system hosting the cluster member on which you installed the application, and *portnumber* is the port number associated with default host for that cluster member and *testapp* is the name of your test application.

2. Examine the logging messages on the screen.

If your test application contains human tasks, you should see logging messages being written to the screen starting with Looking up the HumanTaskManager API EJB. . . . The application will proceed to create a task, claim it, check input and output data, complete the task, and delete it. The word Passed appears near the end of the log messages to indicate that the application ran successfully.

Make sure that you see all messages you have embedded in your application to indicate success.

What to do next

Install and start other test applications.

Installing and accessing other applications:

Install and access applications from the administrative console or Business Process Choreographer Explorer to further test your deployment environment.

Before you begin

You must have successfully installed and configured a deployment environment.

About this task

You can install and start other applications similarly to the way you installed your test application. To access these applications you will use the administrative console or Business Process Choreographer Explorer.

Procedure

1. Locate your application.
In the administrative console click **Applications** → **New Application** and locate the application to install.
2. Install the application.
3. Start the application.
4. Access the application.

Enter a URL for the application in a browser window. For example, `http://hostname:portnumber/myapp` where *hostname* is the fully qualified DNS name (or IP address) of the system corresponding to the cluster member on which you've installed the application, *portnumber* is the port number associated with `default_host` for that cluster member, and *myapp* is the name of the application that you want to access.

From Business Process Choreographer Explorer:

- a. Enter a URL in the following form in a browser window:
`http://hostname:portnumber/bpc` where *hostname* is the fully qualified DNS name (or IP address) of the system corresponding to the cluster member, on which you've installed the application, and *portnumber* is the port number associated with `default_host` for that cluster member.
A page will appear labeled **My Tasks**, but with no tasks listed.
 - b. Select **My Process Templates**. You should see templates listed corresponding to any applications that you installed.
 - c. Use the interface controls on the page to start a task, work on it, complete it, and so on. For more information on running Business Process Choreographer tasks, see "Administering business processes and human tasks."
5. If desired, you can check the servant logs for the cluster member to view a record of the application and check for errors.

Configuring the Business Process Management components manually

To configure the Business Process Management components manually you will need to use the administrative console to perform the tasks detailed in the following subtopics:

Creating a cluster

The following instructions explain how to create a cluster with one cluster member in an empty managed node. The benefit of using the administrative console to create a cluster is that you can undo your changes as you go and you can use a graphical user interface. It is good practice to create a cluster because it makes it easier later to develop the cluster, for example if you want to add more servers to the cluster for reasons of scalability.

Before you begin

Before you create a cluster, start the deployment manager and the node agent for the managed node. See *Starting a server from the MVS console* for more information.

Procedure

1. Log in to the administration console and navigate to **Servers** → **Clusters** → **WebSphere application server clusters**.
2. Click **New**.
3. Type a name for the cluster and short name.
4. Click **Next**.
5. Type the long and short names for the server.
6. From the **Select node** list, click the node in which you want to define the server.
7. Select the button to create the member using an application server template.

8. Select **defaultProcessServerZOS** from the drop down as the application server template with which to create the cluster member.
9. Click **Next**. A blank form is displayed which you can use to define additional cluster members. The server you have just created is listed at the bottom of the screen.
10. Click **Next**.
11. Check the details on the summary screen and click **Next**.
12. Save your configuration changes. The cluster you have just created is displayed in the list.
13. Customize the port numbers to suit your configuration. See techdoc TD104066 for directions, and for a jython script to assist you. The technote can be found in *Creating new Application Servers in WAS V6.1 for z/OS*.

Results

The cluster is created with your chosen server in the selected managed node as the first cluster member.

What to do next

Next, Create the messaging engine data sources.

Configuring components

You can configure the following Business Process Management components:

- Configuring SCA support for a server or cluster
- Configuring all REST services on the administrative console
- Configuring Business Process Choreographer
- Configuring Business Space
- Configuring Business Space on WebSphere Portal
- Configuring business rules and selectors
- “Configuring the relationship service” on page 202
- Configuring extended messaging resources
- Setting up the messaging server environment
- Configuring the JNDILookup Web Service
- “Configuring Common Event Infrastructure” on page 213
- Configuring WebSphere Business Integration Adapters
- Configuring WebSphere Process Server for Service Federation Management

Re-configuring a WebSphere Enterprise Service Bus with WebSphere Process Server functions

If you have previously configured a WebSphere Process Server node with only WebSphere Enterprise Service Bus features but your business requirements have since changed, you can reconfigure the server so that it accesses the additional components of WebSphere Process Server, for example, Business Process Choreographer features.

About this task

When you unloaded the WebSphere Process Server media on to the z/OS system and ran the `zWPSInstall.sh` script, you set up the necessary product definitions for

WebSphere Process Server even though you subsequently ran the `zWESBConfig.sh` script to configure the node as a WebSphere ESB node. You can, therefore, run the `zWPSCConfig.sh` script to reconfigure the node to use the additional features of WebSphere Process Server.

Procedure

On the WebSphere ESB node, run the `zWPSCConfig.sh` script. See `zWPSCConfig.sh` and `zWESBConfig.sh` scripts for more details.

The `zWPSCConfig.sh` script verifies that the server is configured for WebSphere ESB, then configures the server with the additional components that are required for WebSphere Process Server; for example Business Process Choreographer.

Results

Your node is now installed and configured with WebSphere Process Server.

Configuring SCA support for a server or cluster

Use the Service Component Architecture (SCA) console page to enable a server or cluster in a network deployment environment to host service applications, their required messaging engines and destinations, or both.

Before you begin

Before configuring SCA support, determine the following:

- Where to host the messaging engines and destinations (use either a local or remote bus member).
- Whether you need to configure the SCA system bus only, or whether you also need to configure the SCA application bus. The application bus is configured by default and is required if you plan to deploy SCA applications that use WebSphere Business Integration Adapters.

Security role required for this task: You must be logged in as administrator or configurator to perform the following task.

About this task

Service applications require the use of one or more of the automatically created service integration buses, which must have configured messaging engines for destinations. By default, new servers and clusters in a network deployment configuration are not configured to host SCA applications and their destinations.

To configure SCA support on your server or cluster, perform the following steps.

Procedure

1. From within the administrative console, click one of the following, depending on your scope:
 - **Servers** → **Server Types** → **WebSphere application servers** → *server_name* → **Service Component Architecture**
 - **Servers** → **Clusters** → **WebSphere application server clusters** → *cluster_name* → **Service Component Architecture**
2. Click **Support the Service Component Architecture components**.

3. In the Bus Member Location panel, specify where you want to host the destinations and messaging engines required by the SCA applications. There are two options:
 - **Local.** Specifies that you plan to host SCA applications, destinations, and messaging engines on the current server or cluster.
 - **Remote.** Specifies that you plan to host SCA applications on the current server or cluster while hosting destinations and messaging engines on a remote server or cluster (also referred to as a *deployment target*).
4. **(Remote bus member only)** If you selected **Remote** in the previous step, specify the remote server or cluster you want to use to host application destinations and messaging engines. Use the drop-down menu to select an existing deployment target (one that is already configured as a member of the SCA system bus), or click **New** to select a new server or cluster from the Browse Deployment Target page.

If you select a new server or cluster from the Browse Deployment Target page, the necessary messaging is automatically configured on that target when you complete the SCA configuration documented in this topic.

5. Use the table in the System Bus Member panel to verify or modify the system bus data source configuration.
 - a. Verify any default values in the **Database name**, **Schema**, **Create Tables**, **User name Password**, **Server**, and **Provider** fields. See the online help for detailed information about these fields and the values they accept.
 - b. If no default values exist in these fields, or if the default values are incorrect, enter the appropriate values for the system bus data source. You can enter values directly in the field or by clicking **Edit** and making edits on the Data Source details page.
 - c. Optional: Ensure that the data source can contact and authenticate with the database by clicking **Test Connection**.
6. Use the table in the Application Bus Member panel to verify or modify the application bus data source configuration.
 - a. Ensure the **Enable the WebSphere Business Integration Adapter components** option is selected.

Note: If you do not want to use the application bus, clear the **Enable the WebSphere Business Integration Adapter components** option and proceed to Step 7.

- b. Verify any default values in the **Database name**, **Schema**, **Create Tables**, **User name Password**, **Server**, and **Provider** fields. See the online help for detailed information about these fields and the values they accept.
 - c. If no default values exist in these fields, or if the default values are incorrect, enter the appropriate values for the application bus data source. You can enter values directly in the field or by clicking **Edit** and making edits on the Data Source details page.
7. Click **OK** to complete the SCA configuration.
8. Save your changes. You can also optionally review the changes you have made.

What to do next

Next, you can continue configuration as required:

- Configuring Business Process Choreographer
- Configuring Common Event Infrastructure

Related information

Considerations for Service Component Architecture support in servers and clusters
Servers and clusters can support Service Component Architecture (SCA) applications, application destinations, or both.

Considerations for Service Component Architecture support in servers and clusters

Servers and clusters can support Service Component Architecture (SCA) applications, application destinations, or both.

SCA applications (also called service applications) require the use of one or more of the automatically created service integration buses. Each application uses a set of messaging resources, which are called *destinations*. These destinations require configured messaging engines, and they can be hosted on the same server or cluster as the application or on a remote server or cluster. Messaging engines use database data stores.

By default, new servers and clusters in a network deployment configuration are not configured to host SCA applications and their destinations.

Note: A stand-alone server has SCA support automatically configured. You cannot disable this configuration.

To enable this support, use the Service Component Architecture page in the administrative console. For servers, ensure that the application class-loader policy is set to `Multiple`.

Before enabling SCA support for a server or cluster in a network deployment or managed node environment, determine which of the following possible configurations you want to implement:




- **Remote bus member configuration:** The server or cluster hosts SCA applications, but the destinations are hosted on a remote server or cluster. This scenario requires the remote service integration bus members to be configured with the messaging engines needed to host the destination.

While the use of remote messaging requires initial investment in planning for and configuring the service integration bus and its members, that configuration can be reused by multiple members within the application cluster. Messages are distributed to every member. In addition, the initial configuration can be structured to provide failover support.

- **Local bus member configuration:** The server or cluster hosts both SCA applications and application destinations. The required messaging engines are configured using the local bus members on the server or cluster.

Refer to the planning topics to help you decide which configuration is appropriate for your environment.

Related information

-  [Configuring class loaders of a server](#)
-  [Learning about service integration buses](#)
-  [Messaging engines](#)

Configuring all REST services on the administrative console

Configure all Representational State Transfer (REST) services for your environment by using the REST service administrative console page.

Before you begin

Before you complete this task, you must have installed your WebSphere business process management product.

About this task

The deployment of the REST services is performed automatically in a stand-alone server profile. For other types of configurations, the administrative console page allows you to configure REST services for all of your product's widgets in Business Space. On the REST Services page, you can view all services for your environment and enable or disable each service individually.

Procedure

1. Click **Services** → **REST services** → **REST services**.
The REST Services page opens, displaying all REST services in your environment.
2. For the **Scope section**, designate all to view all REST services in your environment, or select a server or cluster where you have REST services enabled.
3. In the table that lists the REST services for the provider, in each row, select the **Enabled** check box if you want to enable the individual REST service, or clear the **Enabled** check box if you want to disable the individual REST service.
4. For each individual service that you want to enable, type a meaningful description in the **Description** column.
5. Click **OK** to commit the changes to the services.

Configuring REST services in a service provider

Configure Representational State Transfer (REST) services in a service provider by using the REST service providers configuration administrative console page.

Before you begin

Before you complete this task, you must have installed your WebSphere business process management product.

About this task

The deployment of the REST services is performed automatically in a stand-alone server profile. For other types of configurations, the administrative console page allows you to configure REST services. On the REST service providers

configuration page, you can view all services for a selected service provider and enable or disable each service individually. If you prefer to manage REST services by your server or cluster (or by business processes or human task components), use the REST Services administrative console page.

Procedure

1. Click **Services** → **REST services** → **REST service providers** → .
The REST service providers page opens, displaying all REST service providers.
2. Click a provider link to configure the services for the group of REST services managed by that provider.
The REST service providers configuration page opens, displaying all REST services in the provider.
3. Select a **Protocol** from the list for all REST services that you want to configure so they are available in your runtime environment. Configure a full URL path by selecting either **https://** or **http://** and then type the **Host Name or Virtual Host in a Load-Balanced Environment** and **Port**. Use a fully qualified host name.
If you want REST requests to go directly to the application server, type the application server host name and port. If you want REST requests to go to a proxy server or HTTP server that sits in front of one or more application servers, type the host name and port of the proxy server or HTTP server that you have already set up. In an environment with a load balancer or proxy server between the browser and the Business Space and REST services, make sure that what you designate for the protocol, host, and port matches the browser URL for accessing Business Space.
4. In the table that lists the REST services for the provider, in each row, select the **Enabled** check box if you want to enable the individual REST service, or clear the **Enabled** check box if you want to disable the individual REST service.
5. For each individual service that you want to enable, type a meaningful description in the **Description** column.
6. Click **OK** to commit the changes to the services.

Configuring REST services for a server, cluster, or component

Configure Representational State Transfer (REST) services for a server, cluster or a component by using the REST Services administrative console page.

Before you begin

Before you complete this task, you must have installed your WebSphere business process management product.

About this task

The deployment of the REST services is performed automatically in a stand-alone server profile. For other types of configurations, the REST Services administrative console page allows you to configure services for a server, a cluster, or a component.

Procedure

1. Click one of the following.
 - For system REST services on a server, click: **Servers** → **Server Types** → **WebSphere application servers** → *name_of_server* → **Business Integration** → **REST Services**

- For system REST services on a cluster, click: **Servers** → **Clusters** → **WebSphere application server clusters** → *name_of_cluster* → **Business Integration** → **REST Services**
- For business process REST services on a server, click: **Servers** → **Server Types** → **WebSphere application servers** → *name_of_server* → **Business Integration** → **Business Flow Manager** → **REST Services**
- For business process REST services on a cluster, click: **Servers** → **Clusters** → **WebSphere application server clusters** → *name_of_cluster* → **Business Integration** → **Business Flow Manager** → **REST Services**
- For human task REST services on a server, click: **Servers** → **Server Types** → **WebSphere application servers** → *name_of_server* → **Business Integration** → **Human Task Manager** → **REST Services**
- For human task REST services on a cluster, click: **Servers** → **Clusters** → **WebSphere application server clusters** → *name_of_cluster* → **Business Integration** → **Human Task Manager** → **REST Services**

The REST Services page appears, displaying all default REST services that you can configure for use with your server or cluster (or Business Flow Manager or Human Task Manager component). If a REST service has already been configured, you see a message displayed.

2. Select a **Protocol** from the list for all REST services that you want to configure so they are available in your runtime environment. Configure a full URL path by selecting either **https://** or **http://** and then type the **Host Name or Virtual Host in a Load-Balanced Environment** and **Port**. Use a fully qualified host name.

If you want REST requests to go directly to the application server, type the application server host name and port. If you want REST requests to go to a proxy server or HTTP server that sits in front of one or more application servers, type the host name and port of the proxy server or HTTP server that you have already set up. In an environment with a load balancer or proxy server between the browser and the Business Space and REST services, make sure that what you designate for the protocol, host, and port matches the browser URL for accessing Business Space.

3. In the table of REST services, in each row, select the **Enabled** check box if you want to enable the individual REST service, or clear the **Enabled** check box if you want to disable the individual REST service.
4. In the table of REST services, type a meaningful description for each of the REST services in the **Description** field.
5. Click **OK** to commit the changes to the services.

To modify the REST service configuration at later time, you can come back to the REST Services page or you can use other administrative console pages to manage the configuration of REST service endpoints. The REST service providers page allows you to select service provider that you want to configure. The REST services page accessed from **Services** → **REST services** allows you to configure all REST services in your environment.

Configuring REST services using the command line

Representational State Transfer (REST) services must be configured before you can use them in your runtime environment. If you do not use the REST Services administrative console page, use the `updateRESTGatewayService` command.

Before you begin

Before you complete this task, you must have installed your WebSphere business process management product.

For WebSphere Process Server, if you have configured Business Process Choreographer, the Human Task Management REST services are already configured. However the REST Services Gateway application, which is a service provider for other REST services, must be configured with the `updateRESTGatewayService` command.

Procedure

1. Open a command window.
The `wsadmin` command can be found in the `profile_root/bin` directory for a stand-alone server environment, or in the `deployment_manager_profile_root/bin` directory for a network deployment environment.
2. At the command prompt, type the `wsadmin` command to start the `wsadmin` environment.
3. Use the `updateRESTGatewayService` command to configure REST services specifying the cluster or the server and node. The `-enable` parameter is optional, and if not specified, defaults to `true`.
4. Run `save` command.

Example

The following example uses Jython to run the `updateRESTGatewayService` command and then save the changes. It configures the REST services on a cluster.

```
AdminTask.updateRESTGatewayService(['-clusterName  
cluster_name'])  
AdminConfig.save()
```

The following example uses Jacl:

```
$AdminTask updateRESTGatewayService {-clusterName  
cluster_name}  
$AdminConfig save
```

Configuring Business Process Choreographer

For information on how to configure Business Process Choreographer, go to the WebSphere Process Server for z/OS, version 7.0, information center and review the topics under **Configuring WebSphere Process Server > Configuring Business Process Choreographer**. You can also find this information in the *Business Process Choreographer* PDF.

Configuring Business Space

Install and configure Business Space powered by WebSphere to set up a common interface for application users to create, manage, and integrate Web interfaces across the IBM WebSphere business process management portfolio.

Before you begin

You must install the product software. When you install your product, Business Space files are included with the installation for the profiles that you configured.

For WebSphere Process Server runtime environments that need the Human Task Management widgets, you must configure Business Process Choreographer. For more information, see *Configuring Business Process Choreographer* in the WebSphere Process Server documentation.

About this task

Business Space is supported with the following database products to match support for the WebSphere product you are using:

- Derby Embedded (for WebSphere Business Monitor, WebSphere Business Services Fabric, WebSphere Enterprise Service Bus, and WebSphere Process Server).
- Derby Network Server (for WebSphere Business Monitor, WebSphere Enterprise Service Bus, and WebSphere Process Server).
- DB2 Universal (for WebSphere Business Compass, WebSphere Business Monitor, WebSphere Business Services Fabric, WebSphere Enterprise Service Bus, and WebSphere Process Server).
- DB2 for IBM i (for WebSphere Enterprise Service Bus and WebSphere Process Server).
- DB2 for z/OS (for WebSphere Business Monitor, WebSphere Business Services Fabric, WebSphere Enterprise Service Bus, and WebSphere Process Server).
- Microsoft® SQL Server Enterprise 2005 SP 2 and 2008 (for WebSphere Business Services Fabric, WebSphere Enterprise Service Bus, and WebSphere Process Server).
- Oracle 11g (for WebSphere Business Compass, WebSphere Business Monitor, WebSphere Business Services Fabric, WebSphere Enterprise Service Bus, and WebSphere Process Server).

Monitor **Process Server / ESB** If you install WebSphere Process Server, WebSphere Enterprise Service Bus, or WebSphere Business Monitor and create a stand-alone server profile with the typical option, Business Space is installed and configured automatically with a Derby Embedded database. If you are using a stand-alone server profile, you can use the Profile Management Tool with the advanced option to configure Business Space to work with your runtime environment. For more information, see "Configuring Business Space using the Profile Management Tool."

For all products, if you are setting up deployment manager and custom profiles, the simplest way to configure Business Space is with the Deployment Environment Configuration wizard. For more information, see "Configuring Business Space using the Deployment Environment Configuration wizard."

If you have a stand-alone server environment or you are using the Deployment Environment wizard to configure your runtime environment, Representational State Transfer (REST) service endpoints are configured and enabled automatically. For other environments, use the REST services administrative console page to configure the REST services. If you want widgets to be available in Business Space, you must configure the REST service endpoints for those widgets. You must register the REST endpoints so that Business Space associates widgets with the endpoints and the widgets appear in the palette for use.

If you are using deployment manager and custom profiles, you can use the administrative console to configure Business Space.

After your original setup work on the Profile Management Tool or the administrative console, you must also configure the database tables for Business Space. For more information, see "Configuring Business Space database tables."

No matter what tool you used to configure Business Space, you must make sure Business Space works with the security for your environment. For more information, see "Setting up security for Business Space."

Business Space is built on Lotus® Mashups technology. For frequently asked questions and general troubleshooting information about Lotus Mashups, see <http://www.lotus.com/ldd/mashupswiki.nsf/xpViewCategories.xsp?lookupName=Troubleshooting&SessionID=CDFG4HK6EQ>.

What to do next

After you have installed and configured Business Space, users of your runtime environment can open it from the following URL: `http://host:port/BusinessSpace`, where *host* is the name of the host where your server is running and *port* is the port number for your server.

Configuring Business Space using the Profile Management Tool

You can configure Business Space powered by WebSphere using the Profile Management Tool.

About this task

You can start the Profile Management Tool after product installation. In addition, you can use the Profile Management Tool capabilities from the command line by using the `manageprofiles` command-line utility parameter `-configureBSPACE` after product installation. In both situations, Business Space is installed with the same database product as the database product you designate for the Common database. If you selected a database that is not supported with Business Space, the Profile Management Tool configures Business Space with the Derby Embedded database.

Process Server / ESB The Profile Management Tool is not available with WebSphere Process Server for z/OS and WebSphere Enterprise Service Bus for z/OS. For those products, configure Business Space using the administrative console.

For all products, for deployment manager and custom profiles, you can use the administrative console or the Deployment Environment Configuration wizard. See "Configuring Business Space using the administrative console" or "Configuring Business Space using the Deployment Environment Configuration wizard". If you use the Profile Management Tool to create a deployment manager and custom profiles (managed nodes) with the **Deployment environment** profile creation option, Business Space is configured automatically with your deployment environment, but you must manually run scripts to configure the database tables.

For more advanced configuration options on a stand-alone server profile, you must use pages on the administrative console to configure Business Space. For example, if you want to designate a data source that is different than the database you selected for your profile (the WebSphere Business Monitor database, the WebSphere Business Compass database, or the WebSphere Process Server common database), you must use the administrative console to configure Business Space.

If you have decided to use these more advanced configuration options, which require using the administrative console, make sure to complete the following steps:

- When you create the stand-alone server profile using the Profile Management Tool, use the **Advanced** profile creation option and clear the **Configure Business Space** check box, so you can configure Business Space later using the administrative console.
- See "Configuring Business Space using the administrative console."

If you are configuring a stand-alone server, complete step 1. If you are configuring a deployment environment, complete step 2.

Procedure

1. For a stand-alone server, start the Profile Management Tool, select the **Stand-alone server profile** option and complete the following steps.
 - a. Complete one of the following steps on the Profile Creation Options page:
 - Select the **Typical** profile creation option if you want to accept a default installation and configuration of Business Space using the Derby Embedded database.
 - Select the **Advanced** option if you want to configure advanced options for the profile you are creating. Then on the Business Space Configuration page, make sure that the **Configure Business Space** check box is selected. If you want to configure Lotus Webform Server to work with Human Task Management widgets in Business Space, select the **Configure Lotus Webform Server** check box and enter the Webform Server translator and installation root.

Business Space is configured with your product data source. If you are using the Profile Management Tool with IBM WebSphere Dynamic Process Edition, Business Space is configured with the WebSphere Process Server data source.
 - b. When you designate the host name for your profile, use a fully qualified host name.
 - c. On the Database Design page, you have the option of using a database design file that you have created using the database design tool that contains all database configuration for your product, including the database configuration information for Business Space.
 - d. Complete the profile creation using the Profile Management Tool. Business Space is installed. It is configured for the same database product as the that you designated for the Common database (or with Derby Embedded if the database product is not supported).
 - e. If the database is remote, you must configure the database tables after running the Profile Management Tool. See "Configuring Business Space database tables."
2. For a deployment environment, start the Profile Management Tool, select the **Deployment manager profile** or **Custom profile** option and complete the following steps.
 - a. On the Profile Creation Options page, select the **Deployment environment** option to configure each profile with customized configuration values and use it in a deployment environment based on a supplied pattern.
 - b. Follow the Profile Management Tool steps to create a deployment manager profile and custom profiles (managed nodes).

- c. After all the custom nodes are federated, run scripts to configure the database tables manually. See "Configuring Business Space database tables."

What to do next

Note: If your product database is an Oracle database, Business Space is configured with the Profile Management Tool or the `manageprofiles` command-line utility to use the same database, with the default schema `IBMBUSSP`, and the default password that you input during profile creation. If you want to use a different password for the `IBMBUSSP` user name, you must use the administrative console to update JDBC Resources: Find the data source `jdbc/mashupsDS`. Modify the value of the authentication alias to make it match the password of the Business Space schema name. Save your changes and restart the server.

Before using Business Space, set up security that you need to use with Business Space and the widgets your team is using. For more information, see "Setting up security for Business Space."

Note: Business Space uses a proxy component to connect to your REST services. In some cases, if REST services are not responsive, you must update the connection timeout settings from Business Space to your REST services, depending on the performance of the REST service servers. If the REST service connections are timing out, update the following settings. By default, the `socket-timeout` value is set to 30 seconds. Change it to an appropriate value for your situation.

1. Open the file `profile_root/BusinessSpace/node_name/server_name/mm.runtime.prof/config/proxy-config.xml`
2. Change the `proxy:value` for `socket-timeout`. The time is specified in milliseconds.

```
<proxy:meta-data>
  <proxy:name>socket-timeout</proxy:name>
  <proxy:value>30000</proxy:value>
</proxy:meta-data>
```
3. Run the `updateBlobConfig` command using the `wsadmin` scripting client, designating the following parameters: **-serverName** and **-nodeName** for a stand-alone server or **-clusterName** for a cluster, **-propertyFileName** with the value of the path for the `proxy-config.xml` file, and **-prefix** with the value `Mashups_`.
4. Restart the `mm_was_node_server` application from administrative console or the entire server application.

Configuring Business Space as part of the Deployment Environment Configuration wizard

Business Space configuration and Representational State Transfer (REST) service configuration for widgets in Business Space are automatically included in the Deployment Environment Configuration wizard. You can decide which REST services to configure.

Before you begin

Before you begin this task, you must complete the following tasks:

- Install your product.
- Create a profile, making sure to designate a fully qualified host name for the profile.
- Enable security, if you want to set up a secured environment for Business Space.

About this task

If you are setting up deployment manager and custom profiles, this method is the simplest way to configure Business Space.

Procedure

1. On the administrative console, click **Servers** → **Deployment Environments** → **New**. A series of pages in the wizard guides you through the process of creating your deployment environment.
2. Either define the new deployment environment or import a file that contains deployment environment definitions. You can create a deployment environment based on one of the IBM-supplied patterns or you can create a custom deployment environment.
3. On the Deployment Environment Patterns page, select one of the deployment environment patterns.
4. On the Select Nodes page, designate the nodes to participate in your deployment environment.
5. On the Clusters page, specify the number of cluster members from each node to assign to specific deployment environment functions.
6. On the Database page, configure the data source for Business Space, one of the components listed in the table. You can edit the description, test the connection, and set the database product you want to use for the Provider. You cannot select the **Create tables** check box on this page for Business Space. Database tables must be configured manually for Business Space. The database product list contains all databases supported by each component.
7. On the Security page, configure the authentication aliases WebSphere uses when accessing secure components. The authentication alias user name and password can be changed on this page. These aliases are used to access secure components but do not provide access to data sources.
8. For WebSphere Process Server configuration, supply information required to configure the application deployment target to support the deployment of the Business Process Choreographer components. Specify the context roots, security, and human task manager mail session values the wizard uses to configure Business Process Choreographer for this deployment environment.
9. For WebSphere Process Server configuration, configure the business rules manager to run on the cluster or server.
10. On the REST Services page, configure the services for the widgets you want available on Business Space for your runtime environment.
 - Type the port number and the host or virtual host that a client needs to communicate with the server or cluster. In a clustered environment, this is typically the load-balancing server host name and port.
 - If you leave the host and port fields empty, the values default to values of an individual cluster member host and its HTTP port. For a load-balanced environment, you must later change the default values to the virtual host name and port of the load-balancing server. Make sure to designate a fully qualified host name.
 - Set the description for the widgets if needed.
11. On the next page, click **Finish** or **Finish and Generate Environment**.
12. Run the scripts to configure the database tables for Business Space before starting the deployment environment or the clusters. For more information, see "Configuring Business Space database tables."

What to do next

Business Space uses a proxy component to connect to your REST services. In some cases, if REST services are not responsive, you must update the connection timeout settings from Business Space to your REST services, depending on the performance of the REST service servers. If the REST service connections are timing out, update the following settings. By default, the socket-timeout value is set to 30 seconds. Change it to an appropriate value for your situation.

1. Open the file `profile_root/BusinessSpace/node_name/server_name/mm.runtime.prof/config/proxy-config.xml`
2. Change the `proxy:value` for `socket-timeout`. The time is specified in milliseconds.

```
<proxy:meta-data>
  <proxy:name>socket-timeout</proxy:name>
  <proxy:value>30000</proxy:value>
</proxy:meta-data>
```
3. Run the `updateBlobConfig` command using the `wsadmin` scripting client, designating the following parameters: `-serverName` and `-nodeName` for a stand-alone server or `-clusterName` for a cluster, `-propertyFileName` with the value of the path for the `proxy-config.xml` file, and `-prefix` with the value `Mashups_`.
4. Restart the `mm_was_node_server` application from administrative console or the entire server application.

Configuring Business Space for network deployment environments

If you have a distributed or network deployment environment, configure Business Space using the administrative console or commands.

About this task

If you are using deployment manager and custom profiles, you must configure Representational State Transfer (REST) endpoints, configure Business Space, register the REST endpoints, and configure database tables.

Configuring REST services

If you have a stand-alone server environment or you are using the Deployment Environment wizard to configure your runtime environment, Representational State Transfer (REST) services are configured and enabled automatically. For other environments, use the administrative console to configure the REST services.

About this task

If you want widgets to be available in Business Space, you must configure the REST services for those widgets. Later you must register the REST endpoints so that Business Space associates widgets with the endpoints and the widgets appear in the palette for use.

You can configure all REST services for a specific server or cluster. Or, you can select individual services to configure. You can manage individual service configuration by viewing all services for a service provider or by viewing all services for your environment.

Configuring all REST services on the administrative console:

Configure all Representational State Transfer (REST) services for your environment by using the REST service administrative console page.

Before you begin

Before you complete this task, you must have installed your WebSphere business process management product.

About this task

The deployment of the REST services is performed automatically in a stand-alone server profile. For other types of configurations, the administrative console page allows you to configure REST services for all of your product's widgets in Business Space. On the REST Services page, you can view all services for your environment and enable or disable each service individually.

You also must register the REST endpoints with Business Space. Then Business Space associates widgets with these endpoints, and the widgets appear in the palette for use.

If you want to configure multiple instances of the same REST service endpoint, you must manually edit the endpoints file and the widgets metadata file. For more information, see "Enabling Business Space widgets for multiple endpoints."

Procedure

1. Click **Services** → **REST services** → **REST services**.
The REST Services page opens, displaying all REST services in your environment.
2. For the **Scope section**, designate all to view all REST services in your environment, or select a server or cluster where you have REST services enabled.
3. In the table that lists the REST services for the provider, in each row, select the **Enabled** check box if you want to enable the individual REST service, or clear the **Enabled** check box if you want to disable the individual REST service.
4. For each individual service that you want to enable, type a meaningful description in the **Description** column.
5. Click **OK** to commit the changes to the services.

What to do next

- Configure Business Space.
- Configure the database tables (if you are using a remote database or a network deployment environment).
- Register REST service endpoints.
- For multiple instances of service endpoints, for example if you have partitioning of work on two clusters, and you want to have widgets showing data from each cluster, you must enable the additional widgets manually for each additional cluster.
- Set up security for Business Space.

Configuring REST services in a service provider:

Configure Representational State Transfer (REST) services in a service provider by using the REST service providers configuration administrative console page.

Before you begin

Before you complete this task, you must have installed your WebSphere business process management product.

About this task

The deployment of the REST services is performed automatically in a stand-alone server profile. For other types of configurations, the administrative console allows you to configure REST services for all of your product's widgets in Business Space. On the REST service providers configuration administrative console page, you can view all services for a selected service provider and enable or disable each service individually. This page allows you to manage individual service configuration by working with all services for a service provider. If you prefer to manage REST services by your server or cluster (or by business processes or human task components), use the REST Services administrative console page.

You also must register the REST endpoints with Business Space. Then Business Space associates widgets with these endpoints, and the widgets appear in the palette for use.

If you want to configure multiple instances of the same REST service endpoint, you must manually edit the endpoints file and the widgets metadata file. For more information, see "Enabling Business Space widgets to work with multiple endpoints."

Procedure

1. Click **Services** → **REST services** → **REST service providers** → .
The REST service providers page opens, displaying all REST service providers.
2. Click a provider link to configure the services for the group of REST services managed by that provider.
The REST service providers configuration page opens, displaying all REST services in the provider.
3. Select a **Protocol** from the list for all REST services that you want to configure so they are available in Business Space. Configure a full URL path by selecting either **https://** or **http://** and then completing the **Host Name or Virtual Host in a Load-Balanced Environment** and **Port** fields. Use a fully qualified host name.
If you want REST requests to go directly to the application server, type the application server host name and port. If you want REST requests to go to a proxy server or HTTP server that sits in front of one or more application servers, type the host name and port of the proxy server or HTTP server that you have already set up. In an environment with a load balancer or proxy server between the browser and the Business Space and REST services, make sure that what you designate for the protocol, host, and port matches the browser URL for accessing Business Space.
4. In the table that lists the REST services for the provider, in each row, select the **Enabled** check box if you want to enable the individual REST service, or clear the **Enabled** check box if you want to disable the individual REST service.
5. For each individual service that you want to enable, type a meaningful description in the **Description** column.
6. Click **OK** to commit the changes to the services.

What to do next

- Configure Business Space.
- Configure the database tables (if you are using a remote database or a network deployment environment).
- Register REST service endpoints.
- For multiple instances of service endpoints, for example if you have partitioning of work on two clusters, and you want to have widgets showing data from each cluster, you must enable the additional widgets manually for each additional cluster.
- Set up security for Business Space.

Configuring REST services for a server, cluster, or component:

Configure Representational State Transfer (REST) services for a server, cluster or a component by using the REST Services administrative console page.

Before you begin

Before you complete this task, you must have installed your WebSphere business process management product.

About this task

The deployment of the REST services is performed automatically in a stand-alone server profile. For other types of configurations, the REST Services administrative console page allows you to configure services for a server, a cluster, or a component.

You also must register the REST endpoints with Business Space. Then Business Space associates widgets with these endpoints, and the widgets appear in the palette for use.

If you want to configure multiple instances of the same REST service endpoint, you must manually edit the endpoints file and the widgets metadata file. For more information, see "Enabling Business Space widgets to work with multiple endpoints."

Procedure

1. Click one of the following.
 - For REST services on a server, click: **Servers** → **Server Types** → **WebSphere application servers** → *name_of_server* → **Business Integration** → **REST Services**
 - For REST services on a cluster, click: **Servers** → **Clusters** → **WebSphere application server clusters** → *name_of_cluster* → **Business Integration** → **REST Services**
 - For business process REST services on a server, click: **Servers** → **Server Types** → **WebSphere application servers** → *name_of_server* → **Business Integration** → **Business Flow Manager** → **REST Services**
 - For business process REST services on a cluster, click: **Servers** → **Clusters** → **WebSphere application server clusters** → *name_of_cluster* → **Business Integration** → **Business Flow Manager** → **REST Services**

- For human task REST services on a server, click: **Servers** → **Server Types** → **WebSphere application servers** → *name_of_server* → **Business Integration** → **Human Task Manager** → **REST Services**
- For human task REST services on a cluster, click: **Servers** → **Clusters** → **WebSphere application server clusters** → *name_of_cluster* → **Business Integration** → **Human Task Manager** → **REST Services**

The REST Services page appears, displaying all default REST services that you can configure for Business Space widgets for use with your product or component (Business Flow Manager or Human Task Manager). If a REST service has already been configured, you see a message displayed.

2. Select a **Protocol** from the list for all REST services that you want to configure so they are available in Business Space. Configure a full URL path by selecting either **https://** or **http://** and then completing the **Host Name or Virtual Host in a Load-Balanced Environment** and **Port** fields. Use a fully qualified host name. If you want REST requests to go directly to the application server, type the application server host name and port. If you want REST requests to go to a proxy server or HTTP server that sits in front of one or more application servers, type the host name and port of the proxy server or HTTP server that you have already set up. In an environment with a load balancer or proxy server between the browser and the Business Space and REST services, make sure that what you designate for the protocol, host, and port matches the browser URL for accessing Business Space.
3. In the table of REST services, in each row, select the **Enabled** check box if you want to enable the individual REST service, or clear the **Enabled** check box if you want to disable the individual REST service.
4. In the table of REST services, type a meaningful description for each of the REST services in the **Description** field.
5. Click **OK** to commit the changes to the services.

To modify the REST service configuration at later time, you can come back to the REST Services page or you can use other administrative console pages to manage the configuration of REST service endpoints. The REST service providers page allows you to select service provider that you want to configure. The REST services page accessed from **Services** → **REST services** allows you to configure all REST services in your environment.

What to do next

- Configure Business Space.
- Configure the database tables (if you are using a remote database or a network deployment environment).
- Register REST service endpoints.
- For multiple instances of service endpoints, for example if you have partitioning of work on two clusters, and you want to have widgets showing data from each cluster, you must enable the additional widgets manually for each additional cluster.
- Set up security for Business Space.

Configuring REST services using the command line:

All widgets required for your product are installed with Business Space powered by WebSphere. The Representational State Transfer (REST) services for widgets must be configured, enabled, and registered with Business Space before your team can use the widgets in Business Space. If you do not use the REST Services administrative console page, use the `updateRESTGatewayService` command.

Before you begin

Before you complete this task, you must have installed your WebSphere business process management product.

For WebSphere Process Server, if you have configured Business Process Choreographer, the Human Task Management REST services are already configured. However the REST Services Gateway application, which is a service provider for other REST services, must be configured with the `updateRESTGatewayService` command.

About this task

The deployment of the REST services is performed automatically in a stand-alone server profile. For other types of configurations, the REST Services administrative console page or the `updateRESTGatewayService` command allows you to configure services for REST application programming interfaces (APIs) for all of your product's widgets in Business Space.

You also must register the REST endpoints with Business Space. Then Business Space associates widgets with these endpoints, and the widgets appear in the palette for use.

If you want to configure multiple instances of the same REST service endpoint, you must manually edit the endpoints file and the widgets metadata file. For more information, see "Enabling Business Space widgets for multiple endpoints."

Procedure

1. Open a command window.
The `wsadmin` command can be found in the `profile_root/bin` directory for a stand-alone server environment, or in the `deployment_manager_profile_root/bin` directory for a network deployment environment.
2. At the command prompt, type the `wsadmin` command to start the `wsadmin` environment.
3. Use the `updateRESTGatewayService` command to configure REST services specifying the cluster or the server and node. The `-enable` parameter is optional, and if not specified, defaults to `true`.
4. Run the `save` command.

Example

The following example uses Jython to run the `updateRESTGatewayService` command and then save the changes. It configures the REST services on a cluster.

```
AdminTask.updateRESTGatewayService(['-clusterName  
  cluster_name'])  
AdminConfig.save()
```

The following example uses Jacl:

```
$AdminTask updateRESTGatewayService {-clusterName  
  cluster_name}  
$AdminConfig save
```

What to do next

- Configure Business Space.

- Configure the database tables (if you are using a remote database or a network deployment environment).
- Register REST service endpoints.
- For multiple instances of service endpoints, for example if you have partitioning of work on two clusters, and you want to have widgets showing data from each cluster, you must enable the additional widgets manually for each additional cluster.
- Set up security for Business Space.

Configuring Business Space and registering REST endpoints on the administrative console

You can install and configure Business Space powered by WebSphere using the administrative console.

Before you begin

Before you begin this task, you must complete the following tasks:

- Install the product software and created a profile. When you install your product, Business Space files are included with the installation for the profiles that you set up. Your profile is not configured for Business Space until you explicitly configure Business Space on the profile.
- Configure Business Process Choreographer for WebSphere Process Server runtime environments that need the Human Task Management widgets. For more information, see "Configuring Business Process Choreographer" in the WebSphere Process Server documentation.
- Enable security, if you want to set up a secured environment for Business Space.
- Configure Representational State Transfer (REST) services. If you have a stand-alone server environment or you are using the Deployment Environment wizard to configure your runtime environment, the REST service endpoints are configured and enabled automatically. For other environments, use the REST services administrative console page to configure the REST services. If you want widgets to be available in Business Space, you must configure the REST services for those widgets. On the Business Space Configuration administrative console page, you register the REST endpoints so that Business Space associates widgets with the endpoints and the widgets appear in the palette for use.
- If you want to configure Business Space on a server or cluster using a different data source than the product data source: Create the data source in the server or cluster scope with the correct JNDI name of jdbc/mashupDS before configuring Business Space using the administrative console.
- For Oracle, to use a different schema for the Business Space tables than the one used by the product database, complete the following steps to create a data source manually before you open the Business Space Configuration page:
 - Create the schema using the database product software.
 - Use the administrative console to configure the JDBC provider.
 - Use the administrative console to create a data source with the JNDI name of jdbc/mashupDS at the server or cluster scope, depending on your environment.
 - Use the administrative console to create an authentication alias. Set the user name to the schema you created and set the authentication according to your Oracle setup.
 - Set the authentication alias on the data source.

About this task

If you are using deployment environments or other advanced profile configuration, you must use the administrative console to configure Business Space to work with your runtime environment. Business Space is a browser-based graphical user interface for the business users of the application that is running with the profile you set up. In Business Space, you and your application users can customize content from products in the WebSphere business process management portfolio.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane click **Servers** → **Server Types** → **WebSphere application servers** or **Servers** → **Clusters** → **WebSphere application server clusters**.
3. Select the name of your server or cluster target.
4. On the Configuration page, under **Business Integration**, click **Business Space Configuration**. The Business Space Configuration page appears. If Business Space has already been configured, you can view this page but cannot edit the fields.
5. Select the **Install Business Space service** check box.
6. In the **Database schema name** box, type the name of the database schema you want to use for the Business Space database.

Note: In Oracle, the schema is the same as the user name set on the authentication alias on the data source.

7. If no data source is designated in the **Existing Business Space data source** field, go to **Create Business Space data source using:** and select a data source that connects to the database you want to use with Business Space.

Designating a data source under **Create Business Space data source using:** creates a data source for Business Space with a JNDI name of `jdbc/mashupDS` that is modeled on the data source you selected.

The Business Space data source is created on the server or cluster on which you are configuring Business Space, even if the product data source is on a different server or cluster.

Note: If you do not see an existing data source that you want to use, you must cancel the Business Space Configuration page, set up the database and the data source that you want to use, and then restart the Business Space Configuration page to complete the configuration. For more information, see the *Before you begin* section.

8. Click **OK**.
9. To register the proper deployment target (cluster or server) for the system Representational State Transfer (REST) endpoints for each of the widgets you are using in Business Space, click **REST service endpoint registration**. The target that you select for a REST service endpoint type can set the scope of the data displayed in some widgets. Or, you might want to select a particular cluster or server for better performance or availability. If you do not specify the target, the REST endpoint of this type is not registered with Business Space, and any widgets that need the REST service endpoint of this type will not be visible in Business Space.
10. Save the configuration.
11. Run the scripts to configure the database tables for Business Space before starting the deployment environment or the clusters. The scripts were

generated when you completed the configuration. For more information, see [Configuring Business Space database tables](#).

What to do next

Note: If you are using Oracle, the password of the authentication alias of the Business Space data source is set to same as the schema name of Business Space. The default value of the schema is IBMBUSSP. When you configure Business Space, you can specify a different schema on the administrative console or in the command line. In that case, the default password is the same as the schema you specify. If you want to use a different password for the Business Space user name, you must use the administrative console to updated JDBC Resources: Find the data source jdbc/mashupsDS. Modify the value of the authentication alias to make it match the password of the Business Space schema name. Save your changes and restart the server.

Note: Business Space uses a proxy component to connect to your REST services. In some cases, if REST services are not responsive, you must update the connection timeout settings from Business Space to your REST services, depending on the performance of the REST service servers. If the REST service connections are timing out, update the following settings. By default, the socket-timeout value is set to 30 seconds. Change it to an appropriate value for your situation.

1. Open the file *profile_root/BusinessSpace/node_name/server_name/mm.runtime.prof/config/proxy-config.xml*
2. Change the proxy: value for socket-timeout. The time is specified in milliseconds.

```
<proxy:meta-data>
  <proxy:name>socket-timeout</proxy:name>
  <proxy:value>30000</proxy:value>
</proxy:meta-data>
```
3. Run the updateBlobConfig command using the wsadmin scripting client, designating the following parameters: **-serverName** and **-nodeName** for a stand-alone server or **-clusterName** for a cluster, **-propertyFileName** with the value of the path for the proxy-config.xml file, and **-prefix** with the value Mashups_.
4. Restart the mm_was_node_server application from administrative console or the entire server application.

Configuring Business Space using the command line

You can set up and configure Business Space powered by WebSphere using the wsadmin command. You can use the wsadmin command to perform the same configuration of Business Space that you can perform in the administrative console.

Before you begin

Before you begin this task, you must complete the following tasks:

- Install the product software and create a profile. When you install your product, Business Space files are included with the installation for the profiles that you set up. Your profile is not configured for Business Space until you explicitly configure Business Space on the profile.
- Configure Business Process Choreographer for WebSphere Process Server runtime environments that need the Managing Tasks and Workflows widgets. For more information, see "Configuring Business Process Choreographer" in the WebSphere Process Server documentation.
- Enable security, if you want to set up a secured environment for Business Space.

- Configure Representational State Transfer (REST) services. If you have a stand-alone server environment or you are using the Deployment Environment wizard to configure your runtime environment, the REST service endpoints are configured and enabled automatically. For other environments, use the REST services administrative console page to configure the REST services. If you want widgets to be available in Business Space, you must configure the REST service endpoints for those widgets. You must register the REST endpoints so that Business Space associates widgets with the endpoints and the widgets appear in the palette for use.
- If you want to configure Business Space on a server or cluster using a different data source than the product data source: Create the data source in the server or cluster scope with the correct JNDI name of jdbc/mashupDS before configuring Business Space (before running the configureBusinessSpace command).
- For Oracle, to use a different schema for the Business Space tables than the one used by the product database, complete the following steps to create a data source manually before you run the commands to install and configure Business Space in the procedure below:
 - Use the administrative console to configure the JDBC provider.
 - Use the administrative console to create a data source with the JNDI name of jdbc/mashupDS at the server or cluster scope, depending on your environment.

About this task

You can use the command line to configure Business Space if you want to write scripts instead of using the administrative console to configure Business Space.

If you are not sure whether Business Space is already configured, you can run the getBusinessSpaceDeployStatus command to check whether Business Space is configured on a server, cluster, or cell. For more information about that command, see "getBusinessSpaceDeployStatus command."

Procedure

1. Open a command window.
The wsadmin command can be found in the *profile_root/bin* directory for a stand-alone server environment, or in the *deployment_manager_profile_root/bin* directory for a network deployment environment.
2. At the command prompt, type the wsadmin command to start the wsadmin environment.
3. Use the installBusinessSpace command to install the Business Space enterprise archive (EAR) files in your runtime environment.
4. Use the configureBusinessSpace command to configure the data source for Business Space and copy the scripts that configure the database tables to *profile_root/dbscripts/BusinessSpace/node_name_server_name/database_type/database_name* for a stand-alone server or *profile_root/dbscripts/BusinessSpace/cluster_name/database_type/database_name* for a cluster. You must run the scripts that configure the database tables. For more information about the scripts, see "Configuring Business Space database tables."
If you are using a database design file for database configuration, you can use the **-bspacedbDesign** parameter to designate that file when you run the configureBusinessSpace command.
5. After each command, run AdminConfig.save((Jython) or \$AdminConfig save (Jacl).

6. Run the scripts to configure the database tables for Business Space before starting the deployment environment or the clusters. For more information, see [Configuring Business Space database tables](#).

Results

Configuring Business Space sets up a browser-based graphical user interface for the business users of your application that is running with the profile you set up. In Business Space, you and your application users can customize content from products in the WebSphere business process management portfolio.

Example

The following example uses Jython to run the `installBusinessSpace` and `configureBusinessSpace` commands to install the EAR files and configure the data source for Business Space on a cluster. The example designates the schema and the product database to use with Business Space when multiple products are installed. In a situation where both WebSphere Process Server and WebSphere Business Monitor are installed, this example creates a Business Space data source using the properties of the WebSphere Process Server data source.

```
AdminTask.installBusinessSpace(['-clusterName myCluster -save true'])
AdminTask.configureBusinessSpace(['-clusterName myCluster -schemaName mySchema -productTypeForDatasource WPS -save true'])
```

The following example uses Jacl:

```
$AdminTask installBusinessSpace {-clusterName myCluster -save true}
$AdminTask configureBusinessSpace {-clusterName myCluster -schemaName mySchema -productTypeForDatasource WPS -save true}
```

What to do next

Note: If you are using Oracle, the password of the authentication alias of the Business Space data source is set to same as the schema name of Business Space. The default value of the schema is `IBMBUSSP`. When you configure Business Space, you can specify a different schema on the administrative console or in the command line. In that case, the default password is the same as the schema you specify. If you want to use a different password for the Business Space user name, you must use the administrative console to update JDBC Resources: Find the data source `jdbc/mashupsDS`. Modify the value of the authentication alias to make it match the password of the Business Space schema name. Save your changes and restart the server.

To enable Business Space for your runtime environment, you must perform the following steps after configuring Business Space from the command line.

- Register the endpoints with the `registerRESTserviceEndpoint` command.
- Set up security that you need to use with Business Space and the widgets your team is using. For more information, see ["Setting up security for Business Space."](#)

Note: Business Space uses a proxy component to connect to your REST services. In some cases, if REST services are not responsive, you must update the connection timeout settings from Business Space to your REST services, depending on the

performance of the REST service servers. If the REST service connections are timing out, update the following settings. By default, the socket-timeout value is set to 30 seconds. Change it to an appropriate value for your situation.

1. Open the file `profile_root/BusinessSpace/node_name/server_name/mm.runtime.prof/config/proxy-config.xml`
2. Change the `proxy:value` for `socket-timeout`. The time is specified in milliseconds.

```
<proxy:meta-data>
  <proxy:name>socket-timeout</proxy:name>
  <proxy:value>30000</proxy:value>
</proxy:meta-data>
```
3. Run the `updateBlobConfig` command using the `wsadmin` scripting client, designating the following parameters: **-serverName** and **-nodeName** for a stand-alone server or **-clusterName** for a cluster, **-propertyFileName** with the value of the path for the `proxy-config.xml` file, and **-prefix** with the value `Mashups_`.
4. Restart the `mm_was_node_server` application from administrative console or the entire server application.

Configuring Business Space database tables

You can manually install database tables for Business Space powered by WebSphere on a remote database server with scripts that are generated by the installation program. If you are using a deployment environment, or your database is remote, you must install these tables after configuring Business Space.

Before you begin

Before you complete this task, you must have completed the following tasks:

- Installed the product.
- Created profiles and configured servers or clusters for Business Space.
- For Oracle: created the database.
- For Microsoft SQL Server: set SQL Server instance authentication. The SQL Server JDBC driver supports mixed authentication mode only. Therefore, when the SQL Server instance is created, the authentication must be set to **SQL Server and Windows**.
- For all databases, make sure that the database is installed using a UTF-8 Universal character set if you want to use Business Space in your environment.
- Made sure that your application server with Business Space is stopped.

Monitor **Process Server / ESB** If you are using DB2 for z/OS and the required resources have not already been set up as part of the core product installation, complete the following additional items before you begin this task:

- Create a TEMP database and a TEMP table space to contain the declared temporary tables for processing scrollable cursors.
- Create a dedicated STOGROUP to contain the Business Space data.

Monitor **Process Server / ESB** For DB2 for z/OS, if you want to use a different storage group (for example, if you don't want Business Space database tables to be added to the same database and storage group as the common database), you must edit and run the `createStorageGroup.sql` script after you configure Business Space and before you configure the Business Space database tables.

- Edit the createStorageGroup.sql file, available in the following location: *profile_root/dbscripts/BusinessSpace/node_name_server_name/database_type/database_name* for a stand-alone server, or *profile_root/dbscripts/BusinessSpace/cluster_name/database_type/database_name* for a cluster, where *database_type* is either DB2z0SV8 or DB2z0SV9.
- Change the VCAT value from @VCAT@ to the name or alias of the catalog of the integrated catalog facility for the storage group to use.

If you are using DB2 V9.x, and you would like performance improvements, edit the createTableSpace.sql file. The createTableSpace.sql file is available in *profile_root/dbscripts/BusinessSpace/node_name_server_name/database_type/database_name* for a stand-alone server, or *profile_root/dbscripts/BusinessSpace/cluster_name/database_type/database_name* for a cluster.

- Change IMMEDIATE SIZE 8000 PAGESIZE 32K to IMMEDIATE SIZE 8000 AUTOMATIC PAGESIZE 32K.
- Add the line PREFETCHSIZE AUTOMATIC after EXTENTSIZE 16 under both CREATE SYSTEM TEMPORARY TABLESPACE @TSDIR@TMTP and CREATE REGULAR TABLESPACE @TSDIR@REGTP.

About this task

The configBusinessSpaceDB script sets up tables for Business Space with a specific database. If you want to create tables on an existing database other than the specific one, use the createDBTables script with your product.

Procedure

1. Make sure that you are using a user ID with sufficient authority to create tables.
2. Locate the script in the profile you most recently configured, and save it to a location on the same system with the database.
 - For all databases except DB2 for z/OS, locate the configBusinessSpaceDB.bat or configBusinessSpaceDB.sh script.
 - **Process Server / ESB** For WebSphere Process Server for z/OS and WebSphere Enterprise Service Bus for z/OS, locate the createDB.sh script if you want to configure the Business Space database tables with all other database objects.
 - **Monitor** **Process Server / ESB** For DB2 for z/OS, if you don't run the createDB.sh script, you must run the Business Space files individually. Locate createStorageGroup.sql, createDatabase.sql, createTablespace.sql, createTables_BusinessSpace.sql, and createTable.sql.

By default, the scripts are located in the following directory:

profile_root/dbscripts/BusinessSpace/node_name_server_name/database_type/database_name for a stand-alone server, or *profile_root/dbscripts/BusinessSpace/cluster_name/database_type/database_name* for a cluster. The updated scripts (with the information that you entered during profile creation) are located in the profile for the server or cluster that you most recently configured. If you used the Deployment Environment Configuration wizard, the scripts are located in the deployment manager profile. When configuring a remote database, copy the scripts from the system where your product is installed to a place on the remote system.

3. **Process Server / ESB** For WebSphere Process Server for z/OS and WebSphere Enterprise Service Bus for z/OS: If you are configuring DB2 for z/OS, you can use the createDB.sh script to configure the Business Space database tables with all other database objects in one database. For more information, see "Creating DB2 database objects using the createDB.sh script" in the WebSphere Process Server for z/OS documentation.

4. Open a command prompt and run one of the following commands, based on your platform.

For Derby, run the command in the default location (*profile_root*/dbscripts/BusinessSpace/*node_name_server_name/database_type/database_name* for a stand-alone server).

For other database types, copy the folder with the batch files and scripts to the same location as your database and run the command there. Your user ID must have access to the command-line interpreter for the database type and have permission to run commands.

- **On Linux, UNIX, and z/OS platforms:** configBusinessSpaceDB.sh

For Derby, DB2 and SQL Server, use the optional **-createDB** parameter if you want to create a different database instead of using the existing database.

Note: When using SQL Server, you see the following warning statements in the systemout.log file after running the database script: ... Warning! The maximum key length is 900 bytes If you are using the federated repositories as a user registry, you can ignore the warnings. If you are using the stand-alone LDAP registry, ensure that all the user distinguished name (DN) entries in your organization are less than the 131 character limit. If any of the user DN entries exceed 131 characters, you must change the user account registry to the federated repositories option.

For z/OS, run the following files in order:

- createStorageGroup.sql
- createDatabase.sql
- createTablespace.sql
- createTables_BusinessSpace.sql
- createTable.sql

What to do next

- Update the endpoints for widgets that you want to be available in Business Space.
- Set up security for Business Space and the widgets that your team is using.

Registering Business Space widget REST service endpoints using the command line

If you configure Business Space using the administrative console, you must register Representational State Transfer (REST) endpoints so that your team can use the widgets in Business Space. If you do not register your endpoints on the administrative console using the Business Space Configuration and the System REST service endpoint registration pages, you can register them using the registerRESTServiceEndpoint command.

Before you begin

Before you complete this task, you must have completed the following tasks:

- Installed the product.

- Configured the REST services for the widgets that you are using in Business Space by using the REST Services administrative console page or the `updateRESTGatewayService` command. If you have a stand-alone server environment or you are using the Deployment Environment wizard to configure your runtime environment, the REST services are configured and enabled automatically. You configure REST services for business processes and human tasks by configuring the Business Process Choreographer and the Human Task Manager container.
- Configured Business Space by using either the Business Space Configuration administrative console page or the `installBusinessSpace` and `configureBusinessSpace` commands.
- Configured the database tables (if you are using a remote database or a network deployment environment).

About this task

REST services are registered automatically if you have a stand-alone server environment and you configured Business Space with the administrative console or the Profile Management Tool, or if you used the Deployment Environment wizard to configure your runtime environment. Otherwise, you must configure the REST services and then register them.

The System REST service endpoint registration administrative console page or the `registerRESTServiceEndpoint` command allows you to register endpoints for REST services for all of your product's widgets in Business Space. Then Business Space automatically associates widgets with these endpoints, and the widgets appear in the Business Space palette for use.

The `registerRESTServiceEndpoint` command allows you to register a set of endpoints for a given provider, a deployment target, or all unique endpoints in a cell. This command registers the endpoints of the REST services that are in the same cell as Business Space.

Procedure

1. Open a command window.
The `wsadmin` command can be found in the `profile_root/bin` directory for a stand-alone server environment, or in the `deployment_manager_profile_root/bin` directory for a network deployment environment.
2. At the command prompt, type the `wsadmin` command to start the `wsadmin` environment.
3. Use the `registerRESTServiceEndpoint` command to register the Business Space endpoints for REST services for all your product's widgets.
4. After each command, run the `save` command.

Example

The following example uses Jython to run the `registerRESTServiceEndpoint` command and then save the changes. It registers all configured and enabled REST services on the cluster with Business Space.

```
AdminTask.registerRESTServiceEndpoint('[-clusterName
  name_of_rest_services_cluster -businessSpaceClusterName
  name_of_business_space_cluster]')
AdminConfig.save()
```

where *name_of_rest_services_cluster* is the cluster name where REST services are configured and *name_of_business_space_cluster* is the cluster name where Business Space is deployed.

The following example uses Jacl:

```
$AdminTask registerRESTServiceEndpoint
{-clusterName name_of_rest_services_cluster
-businessSpaceClusterName name_of_business_space_cluster}
$AdminConfig save
```

where *name_of_rest_services_cluster* is the cluster name where REST services are configured and *name_of_business_space_cluster* is the cluster name where Business Space is deployed.

The **appName**, **webModuleName**, **type**, **version**, **nodeName**, **serverName**, or **clusterName** parameters are optional.

If you do not specify **type**, **appName**, and **webModuleName** parameters, all unique REST service endpoints configured on the deployment target are registered.

If you do not specify any of those parameters, all unique REST service endpoints configured on any deployment target are registered.

What to do next

Business Space uses a proxy component to connect to your REST services. In some cases, if REST services are not responsive, you must update the connection timeout settings from Business Space to your REST services, depending on the performance of the REST service servers. If the REST service connections are timing out, update the following settings. By default, the socket-timeout value is set to 30 seconds. Change it to an appropriate value for your situation.

1. Open the file *profile_root/BusinessSpace/node_name/server_name/mm.runtime.prof/config/proxy-config.xml*
2. Change the `proxy:value` for `socket-timeout`. The time is specified in milliseconds.

```
<proxy:meta-data>
  <proxy:name>socket-timeout</proxy:name>
  <proxy:value>30000</proxy:value>
</proxy:meta-data>
```
3. Run the `updateBlobConfig` command using the `wsadmin` scripting client, designating the following parameters: **-serverName** and **-nodeName** for a stand-alone server or **-clusterName** for a cluster, **-propertyFileName** with the value of the path for the `proxy-config.xml` file, and **-prefix** with the value `Mashups_`.
4. Restart the `mm_was_node_server` application from administrative console or the entire server application.

Configuring a proxy server or load balancer to use with Business Space

If you are using Business Space in an environment with a proxy server or a load-balancing server, you must set up your environment so that Business Space and widgets work properly.

About this task

In a Network Deployment, or clustered, environment, you might set up a proxy server or an HTTP server for security reasons and for workload balancing. Instead of incoming HTTP requests going directly to an application server, they go to a proxy server that can spread the requests across multiple application servers that perform the work.

You can use other routing servers in place of or in front of the proxy server, for example IBM HTTP Server.

Important: The proxy server (or an alternate routing server) is required for workload balancing HTTP requests across two or more cluster members. The proxy server allows clients to access the applications within this topology.

In an environment with a load balancer or proxy server between the browser and the Business Space and REST services, make sure that what you designate for the REST services protocol, host, and port matches the browser URL for accessing Business Space. On the REST service providers page on the administrative console, verify that for all providers, such as the Business Flow Manager and the Human Task Manager, have the correct protocol, host, and port. For more information about modifying the REST services, see *Configuring REST services in a service provider*.

If you use IBM HTTP Server, you must complete additional mapping steps to verify that modules are mapped to the web server and that the host aliases are configured.

If you use a reverse proxy setup for an HTTP server, you must map the URLs for Business Space and widgets.

Configuring IBM HTTP Server for Business Space:

If you use IBM HTTP Server, you must complete additional mapping steps so that Business Space works in your environment.

Before you begin

Before you configure IBM HTTP Server to work with Business Space, complete the following steps:

- Install IBM HTTP Server
- Make sure that Secure Sockets Layer (SSL) is enabled for IBM HTTP Server.
- Make sure that the web server definition for IBM HTTP Server has been added to the application server.

During the installation of the IBM HTTP Server plug-in, a `configureWeb_server` script is produced by the install process on the web server machine. The `configureWeb_server` script is intended to map web application modules to the web server. Therefore, run this script after the generation of the deployment environment.

Procedure

1. Make sure that modules are mapped to the web server. For each of the applications required by Business Space, verify that the web server is one of the selected targets.
 - a. Log in to the administrative console as an administrative user.

- b. Click **Applications** → **Application Types** → **WebSphere enterprise applications**.
 - c. From the Enterprise Applications panel, click the name of the application. Check the following applications. You might have some or all applications in this list, based on which products you are using with Business Space.
 - **BPMAdministrationWidgets_nodename_servername** (for WebSphere Enterprise Service Bus and WebSphere Process Server)
 - **BusinessSpaceHelpEAR_nodename_servername** (for all products)
 - **BSpaceEAR_nodename_servername**(for all products)
 - **BSpaceWebformsEnabler_nodename_servername** (for all products)
 - **HumanTaskManagementWidgets_nodename_servername** (for WebSphere Process Server and WebSphere Business Monitor)
 - **REST Services Gateway** (for all products)
 - **REST Services Gateway Dmgr** (for WebSphere Enterprise Service Bus and WebSphere Process Server)
 - **mm.was_nodename_servername** (for all products)
 - **WBMDashboardWeb_nodename_servername** (for WebSphere Business Monitor)
 - **webWidgets_nodename_servername** (for WebSphere Enterprise Service Bus)
 - **widgets_busleader_nodename_servername** (for WebSphere Business Compass)
 - **widgets_pubserver_nodename_servername** (for WebSphere Business Compass)
 - **widgets_fabric_nodename_servername** (for WebSphere Business Services Fabric)
 - d. For each application, on the Configuration tab, under Modules, click **Manage Modules**.
 - e. On the Manage Modules page for your application, make sure that the web server is one of the selected targets for each of your modules.
 - In the table, check the Server column for each module to make sure that the web server is one of the selected targets for each of your modules. For example, for the `mm.was_nodename_servername` application, look for the web server to be displayed in the Server column:
WebSphere:cell=qaxs41Cell02,node=qaxs41Node03,server=httpserver
WebSphere:cell=qaxs41Cell02,cluster=Golden.WebApp.
 - If you need to add the web server, select the check box next to the name of the module. Then, in the Clusters and servers list, use the Ctrl key to select multiple targets. For example, to have a web server serve your application, press the Ctrl key and then select the application server cluster and the web server together. Click **Apply**, **OK** and **Save** to save any changes.
2. Verify that the host name alias `default_host` contains the correct information for every cluster member, web server, or proxy server.
 - a. Log in to the administrative console as an administrative user.
 - b. Click **Servers** → **Server Types** → **WebSphere application servers**.
 - c. For every cluster member, click the name of the application server to view the port number for the **WC_defaulthost** port name.
 - Under Communications, expand **Ports**.
 - For the port name **WC_defaulthost**, remember its port number.

- d. From the left navigation area of the administrative console, click **Environment** → **Virtual hosts**.
- e. Click the **default_host** name.
- f. Under Additional Properties, click **Host Aliases**.
- g. If the host name and port number for the cluster members is not displayed on the list, click **New** to add the missing entry to the list. The wildcard character * (asterisk) is supported for the host name.
- h. If you add a new entry, click **Save** and **Synchronize**.

Mapping Business Space URLs for a reverse proxy server:

If you have a reverse proxy setup for your HTTP server, when you are configuring the HTTP server to work with Business Space, you must map the URLs for Business Space and the widgets that your team uses.

Procedure

1. Edit your HTTP server configuration file.
2. Map all of the URLs for Business Space and the widgets that your business users work with in the runtime solution.

URLs for general Business Space framework (all products):

- /BusinessSpace/*
- /mum/*
- /help/*
- /BspaceWebformsProxy/*
- /themes/*

Additional URLs for WebSphere Business Compass widgets:

- /WBPublishingDRAFT/*
- /BusinessLeader/*
- /BusinessLeaderWidgets/*

Additional URLs for WebSphere Business Services Fabric widgets:

- /bpm/bslm/v1/*
- /bpm/glossary/v1/*
- /bpm/governance/v1/*
- /bpm/bvars/v1/var/*
- /rest/*

Additional URLs for WebSphere Business Monitor widgets:

- /BusinessDashboard/*
- /DashboardABX/*
- /monitorServerComponent/*
- /mobile/*
- /rest/*
- /AlphabloxServer/*
- /AlphabloxAdmin/*
- /AlphabloxTooling/*
- /BloxBuilder/*

Additional URLs for WebSphere Enterprise Service Bus widgets:

- /BspaceWidgetsHM/*

- /rest/*
- /PolymorphicWidget/*
- /scaWidget/*
- /ServiceMonitorGraphWidget/*
- /StoreAndForward/*

Additional URLs for WebSphere Process Server widgets:

- /BSpaceWidgetsHM/*
- /SecurityManagerWidgets/*
- /BSpaceWidgetsBCM/*
- /rest/*
- /PolymorphicWidget/*
- /scaWidget/*
- /ServiceMonitorGraphWidget/*
- /StoreAndForward/*

Enabling Business Space widgets for cross-cell environments

You must manually edit endpoints files if Business Space is running on a different cell than where the Representational State Transfer (REST) services are running, or if widgets are on different cells than Business Space.

Before you begin

Before you complete this task, you must have completed the following tasks:

- Installed the product.
- Created profiles, and configured Business Space on a deployment target (server or cluster).
- Configured the database tables (if you are using a remote database or deployment environment).
- For WebSphere Business Compass, you must first update the Endpoints table in the WebSphere Business Compass database. Set the Server_Name column to the Business Space Internet Protocol and the Port column to the Business Space port.

About this task

All widgets required for your product are installed with Business Space, but you must configure and register the endpoints needed by the widgets before your team can use them in Business Space. You can configure and register the endpoints by using administrative console pages. However, if your product and REST services are installed on a different cell than Business Space, you must edit REST service endpoints files so that they access the REST services and your widgets work properly in Business Space.

Edit one or more of the following endpoint files, based on the products you have installed, and the widgets you are using with Business Space:

- WebSphere Business Compass: pubserverEndpoints.xml and busLeaderEndpoints.xml.
- WebSphere Business Monitor: monitorEndpoints.xml
- WebSphere Business Monitor with Alphablox: monitorABXEndpoints.xml
- WebSphere Business Services Fabric: fabricEndpoints.xml

- WebSphere Enterprise Service Bus: `wesbWidgetEndpoints.xml` (for Mediation Policy Administration, Service Browser, and Proxy Gateway widgets), `bpmAdministrationEndpoints.xml` (for Administration widgets)
- WebSphere Process Server: `wpsEndpoints.xml`, `bpmAdministrationEndpoints.xml` (for Administration widgets), `wesbWidgetEndpoints.xml` (for Mediation Policy Administration, Service Browser, and Proxy Gateway widgets), `HumanTaskManagementEndpoints.xml` (for business processes and human tasks), `bspaceWFSEndpoints.xml` (for using Lotus Webform Server with Human Task Management widgets)
- All products: `wsumEndpoint.xml` (for user membership)

If you are an administrator, you can register endpoints and enable widgets by performing the following steps.

Procedure

1. Copy widgets from the cell where they were installed to the cell where Business Space is configured during product installation. Widgets can be found in the `install_root\BusinessSpace\widgets` directory and can be copied to a temporary folder.
2. Run the `installBusinessSpaceWidgets` command to install, deploy, and register designated widgets located in the `install_root\BusinessSpace\widgets` directory.
 - a. Make sure the target server (for a stand-alone server environment) or the deployment manager (for a network deployment environment) is up and running, and on that profile, open a command window.
The `wsadmin` command can be found at the `profiles\profile_name\bin` directory.
 - b. At the command prompt, type the `wsadmin` command to start the `wsadmin` environment.
 - c. Run the `installBusinessSpaceWidgets` command. For a clustered environment, specify the `-clusterName` parameter. For a stand-alone server environment, specify the `-serverName` and `-nodeName` parameters. Specify the `-widgets` parameter with the full path for the directory or file that contains the widgets.
3. Locate the endpoint files in the `install_root\BusinessSpace\registryData\endpoints` directory. For a cluster, use the deployment manager profile root. The file names all end with `Endpoints.xml` or `Endpoint.xml`.
4. For each endpoint file that you are configuring, make a backup copy.
5. Create the following directory on the deployment manager profile of the first cell (if it does not exist): `profile_root\BusinessSpace\registryData\` (where `profile_root` is typically `install_root\profiles\profile_name` or `install_root\pf\profile_name`) and copy the endpoint registration file to that directory.
6. Configure the endpoints as needed by editing the endpoint files. Each endpoint in the endpoint file is designated by a `<tns:Endpoint>` block. Identify the block that you want to change.

Tip: If you do not intend to activate some endpoints, you can remove them from the file to prevent confusion.

The location identified by an endpoint is specified in `<tns:url>`. This value is a path in a web module, specified as a full or relative HTTP URL. By default, the URL is relative. Change it to a full URL path, for example,

`https://virtualhost.com:virtualport/rest/bpm/htm` or `http://host1:9445/WBPublishingDRAFT/`, where the protocol, host, and port identify how the product web module can be accessed.

To locate the port number for the server, perform the following steps:

- Log in to the administrative console.
- Click **Servers** → **Server Types** → **WebSphere application servers**.
- Click the server for which you want to find the port number, and then expand the Ports section.

All applications use the same port as shown in either the `wc_defaulthost` (unsecured host) parameter or the `wc_defaulthost_secure` (secure host) parameter.

Note: If you are using an HTTP server to access your web modules for load balancing, use the host name and port settings of the HTTP server.

7. In the cell where the Business Space server is configured, run the `updateBusinessSpaceWidgets` command to update the endpoint URLs after you have modified the endpoints XML files.
 - a. For your profile, open a command window. The `wsadmin` command can be found at the `profiles\profile_name\bin` directory. For a clustered environment, run the command from the `deployment_manager_profile_root\bin` directory. For a stand-alone server environment, run the command from the `profile_root\bin` directory.
 - b. At the command prompt, type the `wsadmin` command to start the `wsadmin` environment.
 - c. Run the `updateBusinessSpaceWidgets` command. For a clustered environment, specify the `-clusterName` parameter. For a stand-alone server environment, specify the `-serverName` and `-nodeName` parameters. Specify the `-endpoints` parameter with the full path for the directory where the widget endpoint files are located or the full path to a specific endpoint file.

Example

The following example endpoint file is for WebSphere Business Monitor widgets.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- START NON-TRANSLATABLE -->
<tns:BusinessSpaceRegistry
  xmlns:tns="http://com.ibm.bspace/BusinessSpaceRegistry"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://com.ibm.bspace/BusinessSpaceRegistry
  BusinessSpaceRegistry.xsd ">

  <tns:Endpoint>
    <tns:id>{com.ibm.wbimonitor}monitorServiceRootId</tns:id>
    <tns:type>{com.ibm.wbimonitor}monitorServiceRootId</tns:type>
    <tns:version>1.0.0.0</tns:version>
    <tns:url>/rest/</tns:url>
    <tns:description>Location of backing services for Monitor widgets
  </tns:description>
  </tns:Endpoint>

</tns:BusinessSpaceRegistry>
<!-- END NON-TRANSLATABLE -->
```

What to do next

- After running the `installBusinessSpaceWidgets` command or the `updateBusinessSpaceWidgets` command, you must perform manual steps to update Business Space templates and spaces. For more information, see [Updating Business Space templates and spaces after installing or updating widgets](#).
- For multiple instances of service endpoints, for example for partitioning of work on two clusters, and you want to have widgets showing data from each cluster, you must enable the additional widgets manually for each additional cluster. You must edit both the widget endpoints files and the widget catalog files. For more information, see [Enabling Business Space widgets to work with multiple endpoints](#).
- If you have enabled security for your environment, you must make sure that it is set up properly to work with Business Space.

Enabling Business Space widgets to work with multiple endpoints

If you have one Business Space instance configured and you have a need to create another instance of the service endpoints in your environment, you must configure Business Space so that the widgets can display data from the multiple service endpoints. You must edit two files: the endpoints file, which registers endpoints with Business Space, and the widget catalog file, which contains definitions of widgets.

Before you begin

Before you complete this task, you must have completed the following tasks:

- Installed the product.
- Created a server or cluster and configured it for Business Space .
- Configured the database tables (if you are using a remote database or deployment environment).
- Configured the additional Representational State Transfer (REST) services for your additional widgets.

About this task

In a deployment environment, you can have partitioning of work. For example, you can have two clusters, one that processes accounting data and one that processes insurance data. However, a service endpoint serves only one cluster. To access both partitions of work from Business Space, you must register two separate widgets, one for each partition of work, so you can access them both from Business Space. For example, you could have an Account Human Task List widget and an Insurance Task List widget in the catalog (both with the same actual human task list code).

You must manually edit the endpoints file and the widget catalog file.

Widget endpoint files are bundled with each product and are added during the installation of the product. You must edit one or more of the following endpoint files, based on the products you have installed, and the widgets you are using with Business Space:

- WebSphere Business Compass: `pubserverEndpoints.xml` and `busLeaderWidgetEndpoints.xml`.
- WebSphere Business Monitor: `monitorEndpoints.xml`

- WebSphere Business Monitor with Alphablox: monitorABXEndpoints.xml
- WebSphere Business Services Fabric: fabricEndpoints.xml
- WebSphere Enterprise Service Bus: wesbWidgetEndpoints.xml (for Mediation Policy Administration, Service Browser, and Proxy Gateway widgets), bpmAdministrationEndpoints.xml (for Administration widgets)
- WebSphere Process Server: wpsEndpoints.xml, bpmAdministrationEndpoints.xml (for Administration widgets), wesbWidgetEndpoints.xml (for Mediation Policy Administration, Service Browser, and Proxy Gateway widgets), HumanTaskManagementEndpoints.xml (for business processes and human tasks), bspaceWFSEndpoints.xml (for using Lotus Webform Server with Human Task Management widgets)
- All products: wsumEndpoint.xml (for user membership)

Widget catalog files contain the definition of widgets for your product. You must edit one or more of the following widget files, based on the products you have installed, and the widgets you are using with Business Space:

- WebSphere Business Compass: catalog_pubserverWidgets.xml and catalog_busLeaderWidgets.xml
- WebSphere Business Monitor: catalog_WBMonitor.xml
- WebSphere Enterprise Service Bus: catalogProxyGateway.xml and catalog_ServiceAdmin.xml
- WebSphere Process Server: catalog_BPMAAdministration.xml, catalog_BusinessRules.xml, catalog_ServiceAdmin.xml, and catalog_HumanTaskManagement.xml
- WebSphere Business Services Fabric: catalog_fabric.xml

Both the endpoint files and the widget catalog files are located at *install_root*\BusinessSpace\registryData\. The endpoints files are located in the endpoints subdirectory, and the catalog files are located in the catalogs subdirectory.

The directory *install_root*\BusinessSpace\registryData\ contains endpoint and widget catalog template files for your product. You can copy the files that you need to use as a template and add your changes.

Procedure

1. In order to have multiple instances of a widget, you must install the applications that provide widgets with a unique application name and context root for each widget instance.
 - a. Deploy the widget application on the Business Space deployment target (the same server or cluster on which the **BSpaceEAR_server_node** application is running) for each widget instance. Depending on the products you are using, deploy one or more of the following Enterprise Archive (EAR) files:
 - BPMAdministrationWidgets_nodename_servername (for WebSphere Enterprise Service Bus and WebSphere Process Server)
 - HumanTaskManagementWidgets_nodename_servername (for WebSphere Process Server and WebSphere Business Monitor)
 - WBMDashboardWeb_nodename_servername (for WebSphere Business Monitor)
 - wesbWidgets_nodename_servername (for WebSphere Enterprise Service Bus)

- `widgets_busleader_nodename_servername` (for WebSphere Business Compass)
 - `widgets_pubserver_nodename_servername` (for WebSphere Business Compass)
 - `widgets_fabric_nodename_servername` (for WebSphere Business Services Fabric)
- b. When deploying, update the application name and the web module context root names to a unique name. Take note of the context root names that you use.
2. Edit the new REST service endpoints for the additional application deployment targets (the server or cluster where the REST services application is deployed). Create an endpoints file to add service endpoints.
 - a. Locate the endpoint files in the `install_root\BusinessSpace\registryData\endpoints` directory. Copy the endpoints template file, and remove all the endpoints that you do not intend to change.
 - b. Edit the endpoints file and add an additional service endpoint starting with `<tns:Endpoint>`, with a unique ID (`<tns:id>`) and the URL for the new endpoint (`<tns:url>`), but with the same version, and optionally all the locales as the original endpoint. The type (`<tns:type>`) must have the same value as the ID (`<tns:id>`). You can change the name and description, for example, My team's insurance task list.
 - c. When adding endpoints, pay attention to the following information:
 - `<tns:id>`: The ID can be any string but must be unique for all registered endpoints. Ensure that this ID is unique when you are adding additional endpoints.
 - `<tns:type>`: The type must have the same value as `<tns:id>`.
 - `<tns:url>`: For the service endpoint, if the URL is relative, then it is assumed that the REST service endpoint is co-located with the Business Space server. If the URL is relative, make sure the URL is same as the context root you deployed, but with beginning and end directory indications, for example, `<tns:url>/BspaceWidgetsWPS2/</tns:url>`. If your endpoint is on a remote system, update this field with an absolute URL, but with an end directory indication.
 - `<tns:description>`: Type a meaningful description that further details the nature of the data set that this endpoint is working on. It could either be based on the cluster that is working on the data set or the nature of the data set, for example, insurance claim human tasks or accounting data human tasks.
 - d. Save your changes.

Example service endpoint, located in `monitorEndpoints.xml`:

```
<tns:Endpoint>
  <tns:id>{com.ibm.wbimonitor}monitorServiceRootId</tns:id>
  <tns:type>{com.ibm.wbimonitor}monitorServiceRootId</tns:type>
  <tns:version>1.0.0.0</tns:version>
  <tns:url>/rest/</tns:url>
  <tns:description>Location of backing services for Monitor widgets
</tns:description>
</tns:Endpoint>
```

3. In the endpoints file, add a widget endpoint for each widget instance.
 - a. Edit the endpoints file that you created in step 2. Add an additional widget endpoint starting with `<tns:Endpoint>` and with a unique ID (`<tns:id>`). The type (`<tns:type>`) must have the same value as the ID (`<tns:id>`). The URL for the new endpoint (`<tns:url>`) should be the same as the context root you deployed in step 1., but with beginning and end directory

indications, for example, `<tns:url>/BSpaceWidgetsWPS2/</tns:url>`. The widget endpoint you add should contain the same version and can optionally contain all the locales as the original endpoint. You can change the name and description.

- b. When adding endpoints, pay attention to the following information:
 - `<tns:id>`: The ID can be any string but must be unique for all registered endpoints. Ensure that this ID is unique when you are adding additional endpoints.
 - `<tns:type>`: The type must have the same value as `<tns:id>`.
 - `<tns:url>`: For the widget endpoint, make sure the URL is same as the context root you deployed, but with beginning and end directory indications, for example, `<tns:url>/BSpaceWidgetsWPS2/</tns:url>`.
 - `<tns:description>`: Type a meaningful description that further details the nature of the data set that this endpoint is working on. It could either be based on the cluster that is working on the data set or the nature of the data set, for example, insurance claim human tasks or accounting data human tasks.
- c. Save your changes.

Example widget endpoint, located in `monitorEndpoints.xml`:

```
<tns:Endpoint>
<tns:id>{com.ibm.wbimonitor}monitorWidgetRootId2</tns:id>
  <tns:type>{com.ibm.wbimonitor}monitorWidgetRootId2</tns:type>
  <tns:version>1.0.0.0</tns:version>
  <tns:url>/newMonitorWidgetContextRoot/</tns:url>
  <tns:description>Location for Monitor widgets</tns:description>
</tns:Endpoint>
```

4. Create a widget catalog file to add new widget definitions.
 - a. Locate the widget catalog file in the `install_root\BusinessSpace\registryData\catalogs` directory. Copy the catalog template file. For the new file name, use the following standard: `catalog_widget.xml` (with no spaces in the file name), where `widget` is the same as the `id` value of the `<catalog>` element in the file. Remove all the `<category>` elements that you do not intend to change. For the category that you are working with, remove all the `<entry>` elements that you do not intend to change.
 - b. Add an `<entry>` with a unique ID, for example, `id="{com.ibm.bspace.widget}widget_id` and a unique name, for example, `unique-name="{com.ibm.bspace.widget}widget_name`. You can keep all the other definitions.
 - c. Change the title and description to make the new widget available as a distinct widget in Business Space that outlines the nature of the new endpoint. For example, you could name your widget `My team's insurance task list` in the `<title>`. The title should help the business users choose the right widget. The description should help the business users understand the nature of the data and the functionality of the widget that they are selecting.
 - d. Edit the new widget catalog XML file to reference the new widget endpoint: Change the definition to match the `<tns:id>` of the widget endpoint you added in step 3.a.
For example, change it to: ...

```
<definition>endpoint://{com.ibm.wbimonitor}monitorWidgetRootId2/com/ibm/wbimonitor/common/iWidgets/instances_iWidget.xml</definition>
...

```
 - e. In the `<metadata>` of the catalog file, make sure the `endpoint://` matches the type and the ID in the endpoint file (`<tns:type>` and `<tns:id>`).

f. In the <metadata> of the catalog file, make sure the "refVersion" : matches the version in the endpoint file (<tns:version>).

g. Save your changes.

Example widget catalog file:

```
<entry id="{com.ibm.wbimonitor}instances"
unique-name="{com.ibm.wbimonitor}instances">
  <title>
    <!-- END NON-TRANSLATABLE -->
    <nls-string xml:lang="en">Instances</nls-string>
    <!-- START NON-TRANSLATABLE -->
  </title>
  <description>
    <!-- END NON-TRANSLATABLE -->
    <nls-string xml:lang="en">Instances</nls-string>
    <!-- START NON-TRANSLATABLE -->
  </description>
  <shortDescription>
    <!-- END NON-TRANSLATABLE -->
    <nls-string xml:lang="en">This widget displays a dashboard with
the available monitoring context in either individual instances or user-
defined groups of context instances.</nls-string>
    <!-- START NON-TRANSLATABLE -->
  </shortDescription>
  <definition>endpoint://{com.ibm.wbimonitor}monitorWidgetRootId
/com.ibm.wbimonitor/common/iWidgets/instances_iWidget.xml</definition>
  <content>endpoint://{com.ibm.wbimonitor}monitorWidgetRootId/img/
thumb_instances.gif</content>
  <preview>endpoint://{com.ibm.wbimonitor}monitorWidgetRootId/img/
prev_instances.gif</preview>
  <previewThumbnail>endpoint://{com.ibm.wbimonitor}monitorWidgetRootId/
img/prev_instances.gif</previewThumbnail>
  <help>endpoint://{com.ibm.bspace}bSpaceWidgetHelpRootId/topic/
com.ibm.bspace.help.wdg.mon.doc/dash/help_instance_whatIs.html</help>
  <icon>endpoint://{com.ibm.wbimonitor}monitorWidgetRootId/img/
icon_instances.gif</icon>
  <metadata name="com.ibm.mashups.builder.autoWiringEnabled">true
</metadata>
  <metadata name="com.ibm.bspace.version">7.0.0.0</metadata>
  <metadata name="com.ibm.bspace.owner">International Business
Machines Corp.</metadata>
  <metadata name="com.ibm.bspace.serviceEndpointRefs">
[{"name":"serviceUrlRoot", "required":"true",
"refId":"endpoint://{com.ibm.wbimonitor}monitorServiceRootId",
"refVersion":"1.0.0.0"}]</metadata>
</entry>
```

5. Place the new endpoint file and the new catalog file in a compressed file, and run the updateBusinessSpaceWidgets command, using the **-widgets** parameter to specify the compressed file location..

What to do next

- After running the updateBusinessSpaceWidgets command, you must perform manual steps to update Business Space templates and spaces. For more information, see Updating Business Space templates and spaces after installing or updating widgets.
- If Business Space is running on a different cell than where the REST services are running, you must manually edit the endpoints files.
- If you have enabled security for your environment, you must make sure that it is set up properly to work with Business Space.

Configuring widgets for multiple products

You can configure or add Business Space widgets for one BPM product on a Business Space that has already been configured with a different BPM product by using the installBusinessSpaceWidgets command.

Before you begin

Before you complete this task, you must have completed the following tasks:

- Completed all steps to install and configure a BPM product, and configured Business Space.
- Completed all steps to install and configure the additional BPM product.

About this task

You can install more than one BPM product that works with Business Space and configure the widgets for both products after you install the second product. However, if you install a second BPM product after you have already configured Business Space with widgets for the first product, you must use the `installBusinessSpaceWidgets` command to add and configure the second product widgets to work with the same Business Space.

In a stand-alone augmentation, widgets are installed automatically. For example, widgets are installed if you create a WebSphere Process Server stand-alone profile, configure the server for Business Space, install WebSphere Business Monitor, and augment the already-configured server to WebSphere Business Monitor. But for a network deployment environment, when you augment a deployment manager to another product, no additional widgets are installed and configured.

Procedure

1. Make sure the deployment manager profile is up and running, and on that profile, open a command window.
The `wsadmin` command can be found at the `profiles/profile_name/bin` directory.
2. At the command prompt, type the `wsadmin` command to start the `wsadmin` environment.
3. Use the `installBusinessSpaceWidgets` command to install, deploy, and register designated widgets located in the `install_root/BusinessSpace/widgets` directory.

Example

The following example uses Jython to run the `installBusinessSpaceWidgets` to install widgets for IBM WebSphere Business Monitor to work with the Business Space environment that has been previously configured for IBM WebSphere Process Server.

```
AdminTask.installBusinessSpaceWidgets(['-nodeName node_name
-serverName server_name -widgets
install_root\BusinessSpace\widgets\WBM\Widgets_WBMonitor.zip'])
```

The following example uses Jacl:

```
$AdminTask installBusinessSpaceWidgets {-nodeName node_name
-serverName server_name -widgets
install_root\BusinessSpace\widgets\WBM\Widgets_WBMonitor.zip}
```

What to do next

To enable Business Space for your runtime environment, you must perform the following steps after configuring the widgets.

- After running the `installBusinessSpaceWidgets` command or the `updateBusinessSpaceWidgets` command, perform manual steps to update Business Space templates and spaces. For more information, see [Updating Business Space templates and spaces after installing or updating widgets](#).
- Configure REST services. For more information, see [Configuring REST services](#).

- Register REST endpoints. For more information, see "Configuring Business Space and registering REST endpoints on the administrative console."
- Verify security is set up properly to work with Business Space and the widgets your team is using. For more information, see Setting up security for Business Space.

Setting up specific widgets to work in Business Space

Some of the widgets that come with your product require additional configuration steps before you can use them in Business Space.

About this task

Your business process management product includes several widgets, and some require additional configuration to communicate with your solution from Business Space.

Configuring the service monitor

If you are creating a new server and you want to use the Service Monitor widget (available in Business Space) to measure the response time and request throughput for services exposed or invoked by an SCA module, configure and enable service monitoring in the administrative console.

Before you begin

Required security role for this task: If administrative security is enabled, you must be logged in with an administrative role to perform this task.

About this task

The service monitor has a client/server architecture.

- Service monitor agent: Measures the throughput and response time for operations and sends the measurement data to the service monitor server
- Service monitor server: Gathers and aggregates response time and throughput measurements from all running service monitor agents, and then calculates and stores the statistics.

In a deployment environment, the server runs on a support cluster, while the agent runs in the application cluster on the server where you deployed your module. In a stand-alone server environment, the server and agent both run on the stand-alone server.

Important: If you are using an external HTTP server to access Business space, make sure to configure the HTTP server to allow encoded slashes. Refer to the HTTP server documentation for details.

Procedure

1. Log into the administrative console with administrator privileges.
2. Configure the service monitor server.
 - a. From within the console, click **Servers** → **Server Types** → **WebSphere application servers** → *servername* → **Service Monitor**.
 - b. On the Service Monitor page, click **Enable service monitor**.
 - c. Examine the default values for the service monitor buffer size and the query size limit and, if necessary, revise them.

- d. Specify the service monitoring targets. These are the service monitor agents you want to gather data from.

Table 9. Monitoring

Targets to monitor	Steps to perform
Monitor all running service monitor agents	Ensure the All enabled service monitor agents option is checked.
Monitor a specific subset of running service monitor agents	<ol style="list-style-type: none"> 1. Clear the All enabled service monitor agents option. A collection table appears; if this is a new configuration, the table is empty. 2. Click Add. The Browse Deployment Targets page opens. 3. From the collection table on the Browse Deployment Targets page, select the deployment target whose agent you want to monitor. 4. Click OK to return to the Service Monitor Server page. 5. Repeat Step 2 through Step 4 until you have added all the agents you want to monitor.

- e. From the Service Monitor Server page, click **OK**. The configuration is saved and takes effect immediately.
3. Configure the service monitor agent.
 - a. From within the console, click **Servers** → **Server Types** → **WebSphere application servers** → *servername* → **Service Monitor Agent**.
 - b. On the Service Monitor Agent page, click **Enable service monitor agent**.
 - c. Examine the default values for the agent configuration and, if necessary, revise them.
 - d. Click **OK**.

Enabling forms for running Human Task Management widgets in Business Space

If you are working with WebSphere Process Server, you must take additional steps to enable forms for working with Human Task Management widgets in Business Space.

About this task

Topic scope: This topic applies to the following products:

- WebSphere Business Compass
- WebSphere Business Monitor
- WebSphere Process Server
- WebSphere Business Services Fabric

If you have installed Business Space on a different server instance than Business Process Choreographer, you must take additional steps to make forms deployed in separate enterprise applications available to the Human Task Management widgets. This includes HTML-Dojo forms that are generated in WebSphere Integration Developer and IBM Lotus Forms.

Depending on whether both Business Space and WebSphere Process Server are configured on deployment targets in the same WebSphere Network Deployment cell or in different cells, complete one of the following steps:

Procedure

1. For a setup in a single cell: When deploying an enterprise application that contains a process or a human task and forms, you must map the Web modules that contain the HTML files or Lotus form definitions for the forms to the same deployment target that Business Space is configured on.
2. For a setup in a cross-cell environment: Deploy the Web module containing the HTML files or Lotus form definitions for the forms on the deployment target that hosts Business Space in the remote cell. When deploying the Web module, you must specify the context root as defined for the forms in the Human Task Editor in WebSphere Integration Developer. Start the new application on the Business Space server or cluster.

What to do next

If you are using Lotus Webform Server to work with the Human Task Management widgets, you must configure Lotus Webform Server for Business Space.

Configuring Lotus Webform Server for Human Task Management widgets in Business Space:

If you are working with WebSphere Process Server Human Task Management widgets, and you want to use Lotus Webform Server to work with forms during runtime, you must configure Business Space to use Lotus Webform Server.

Before you begin

Before you can use Lotus Webform Server with the Human Task Management widgets in Business Space, you must install Lotus Webform Server 3.5.1 with fix pack 1 or later.

Webform Server can only run on a machine with 32-bit architecture.

When you install Webform Server, make sure to select both **Webform Server - Application Server** and **Webform Server - Translator Server** on the Server components page on the installation tool. On the Optional Deployment settings page, make sure to select **Deploy Webform Server - Translator Server to WebSphere Application Server**. Do not select **Deploy API to WebSphere Application Server** or **WebSphere Process Server**.

Note: If you are using a Derby database, you must install Lotus Webform Server in a separate profile. It cannot use the same profile as Business Space and WebSphere Process Server.

About this task

Topic scope: This topic applies to the following products:

- WebSphere Business Compass
- WebSphere Business Monitor
- WebSphere Process Server
- WebSphere Business Services Fabric

Depending on your environment, perform one of the following three steps.

Procedure

1. If you have a single-server environment and Lotus Webform Server is already installed on the same system as WebSphere Process Server, configure Lotus Webform Server for Business Space by using the Profile Management Tool. Otherwise, go to step 2.
 - a. Start the Profile Management Tool, and create a stand-alone server profile.
 - b. On the Profile Creation Options page, select the **Advanced** option.
 - c. On the Business Space Configuration page, select the **Configure Lotus Webform Server** check box and enter the Webform Server translator and installation root. For more information, see *Creating Advanced stand-alone server profiles*.
2. If Lotus Webform Server is installed on the same system where WebSphere Process Server is installed (and you did not configure Lotus Webform Server in the Profile Management Tool), perform the following steps. Otherwise, go to step 3.
 - a. For your profile, open a command window. The wsadmin command can be found at the `profiles\profile_name\bin` directory. For a clustered environment, run the command from the `deployment_manager_profile_root\bin` directory. For a stand-alone server environment, run the command from the `profile_root\bin` directory.
 - b. At the command prompt, type the wsadmin command to start the wsadmin environment. For example, on Windows platforms, type `wsadmin.bat -conntype NONE`.
 - c. On the same machine where Webform Server is located, run the `configureWebformServer` command, designating the local host and location.

For example, run the following command using Jython:

```
AdminTask.configureLotusWebformServer(['-nodeName', node_name,
'-serverName', server_name, '-translatorHTTPLocation',
'http://localhost:8085/translator', '-serverInstallRoot',
'C:/IBM/LotusWebForms/3.5/WebFormServer'])
AdminConfig.save()
```

Or, run the following command using Jacl:

```
$AdminTask configureLotusWebformServer {-nodeName node_name -serverName
server_name -translatorHTTPLocation http://localhost:8085/translator
-serverInstallRoot C:/IBM/LotusWebForms/3.5/WebFormServer}
$AdminConfig save
```

3. If Lotus Webform Server is installed on a different system than where WebSphere Process Server is installed, complete the following steps.
 - a. Copy the `BSpaceWebformsEnabler.ear` from the `profile_root/installableApps/BusinessSpace` directory to the system that has Webform Server installed. Deploy this ear on the remote application server.
 - b. On the local Business Space profile, in the `bspaceWFSEndpoints.xml` file, set the endpoint `{com.ibm.bspace}bspaceWebformsProxyRootId` to reference the fully qualified location of the `BSpaceWebformsEnabler.ear`. For more information about editing endpoints files, see *Enabling Business Space widgets manually for cross-cell environments*.
 - c. On the Webform Server system, open the administrative console on the profile where you configured the Lotus Webform Server.

- d. Set the following variables by clicking **Environment** → **WebSphere Variables**, then selecting the node that contains the server that you are using, and then clicking **New** for setting each new variable.
 - Set the Webform Server Install Directory variable by creating a variable with the name LFS_DIR and value of the Webform Server Install, for example, c:\Program Files\Lotus Webform Server\3.5\WebformServer.
 - Set the LFS_API_DIR variable by creating a variable with the name LFS_API_DIR and value \${LFS_DIR}\Translator\API.
 - Set the LFS_API_LIB_DIR variable by creating a variable with the name LFS_API_LIB_DIR and value \${LFS_API_DIR}\76\java\classes.
 - Set the LFS_DEP_DIR variable by creating a variable with the name LFS_DEP_DIR and value \${LFS_DIR}\redist.
- e. Set the Java Process Definition.
 - Click **Servers** → **Server Types** → **WebSphere application servers** → *server_name* → **Java and Process Management** → **Process Definition** → **Environment Entries**.
 - Add a PUREEDGE_INI property and value: \${LFS_DIR}\Translator\PureEdgeAPI.ini.
- f. Set the LFS_API_LIB and LFS_DEP_LIB shared libraries.
 - Click **Environment** → **Shared Libraries**.
 - Set the scope by selecting the node that contains the server that you are using. The scope must be the same scope as the environment variable settings.
 - Click **New**.
 - Create an entry with name: "LFS_API_LIB" and classpath (one per line):
 - \${LFS_API_LIB_DIR}/pe_api.jar
 - \${LFS_API_LIB_DIR}/pe_api_native.jar
 - \${LFS_API_LIB_DIR}/uwi_api.jar
 - \${LFS_API_LIB_DIR}/uwi_api_native.jar
 - \${LFS_API_LIB_DIR}/commons-codec.jar
 - \${LFS_API_LIB_DIR}/xmlsec-1.4.1.jar
 - Click **OK**.
 - Click **New**.
 - Create an entry with name: "LFS_DEP_LIB" and classpath (one per line):
 - \${LFS_DEP_DIR}/commons-codec-1.3.jar
 - \${LFS_DEP_DIR}/commons-httpclient-3.0.jar
 - \${LFS_DEP_DIR}/ehcache-1.2.2.jar
 - \${LFS_DEP_DIR}/log4j-1.2.8.jar
 - \${LFS_DEP_DIR}/ws_common.jar
 - \${LFS_DEP_DIR}/ws_framework.jar
 - \${LFS_DEP_DIR}/ws_resourcestore.jar
 - \${LFS_DEP_DIR}/ws_resourcebundle.jar
 - Click **OK**.
- g. Set the server class loader.
 - Click **Servers** → **Server Types** → **WebSphere application servers** → *server_name* → **Java and Process Management** → **Classloader**.
 - If a class loader for your application server does not exist, you must create it. Click **New** and select the parent last option.

- Select the class loader for your application server and click **Shared Library References**.
 - Click **Add**.
 - From the Library Name list, select LFS_API_LIB.
 - Repeat for library: LFS_DEP_LIB
 - Click **OK**.
- h. Configure the Webform Translator location.
- Ensure that the BSpaceWebformsEnabler EAR has been deployed
 - Click **Applications** → **Application types** → **WebSphere enterprise applications** → **BSpaceWebformsEnabler** → **Initialize parameters for servlets**.
 - Set the value for the translatorLocation to the http address of the Webform Server Translator. If the Translator has been configured to run on the same machine as the BSpaceWebFormsEnabler, Then leave the default value of: http://localhost:8085/translator
- i. Save all changes to the master configuration, and restart the server.

Enabling images in Human Task Management widgets

If you are setting up Business Space to include Human Task Management widgets, you can create an endpoints file to use images of team members in those widgets. All widgets that are configured to display a user ID and allow grouping by this user ID can be enabled to display images.

Before you begin

Topic scope: This topic applies to the following products:

- WebSphere Business Compass
- WebSphere Business Monitor
- WebSphere Process Server
- WebSphere Business Services Fabric

About this task

By default, Business Space is configured with no image server identified for Human Task Management widgets, but if you want your business users to see images of their team members, you can enable image retrieval in a new widget endpoint file.

Procedure

1. Create a new file in *install_root*\BusinessSpace\registryData\ For example, name it imageEndpoint.xml.
2. Copy in the following template.

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:BusinessSpaceRegistry xmlns:tns="http://com.ibm.bspace/
BusinessSpaceRegistry" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://com.ibm.bspace/BusinessSpaceRegistry
BusinessSpaceRegistry.xsd ">
<tns:Endpoint>
<tns:id>{com.ibm.bspace.htm)bspaceUserImageServiceRootId</tns:id>
<tns:type>{com.ibm.bspace.htm)bspaceUserImageServiceRootId</tns:type>
<tns:version>1.0.0.0</tns:version>
<tns:url>URL</tns:url>
<tns:description>Location of user image services</tns:description>
</tns:Endpoint>
</tns:BusinessSpaceRegistry>
```

3. Update the URL to reference the appropriate image server servlet that you are using for user images.

The image service endpoint is a reference to a URL prefix where the widgets can find images by concatenating the following information:

- The resolved image service endpoint string.
- The unique identifier Virtual Member Manager (VMM) attribute for each user.
- The .jpg file extension.

For example, if the endpoint URL is `http://myserver:9080/UserImageWeb/UserImageServlet/` and the unique identifier for a user is `id123456`, the widgets retrieve that user's image at the following link: `http://myserver:9080/UserImageWeb/UserImageServlet/id123456.jpg`.

4. Run the `updateBusinessSpaceWidgets` command.
 - a. For your profile, open a command window.

The `wsadmin` command can be found at the `profiles/profile_name/bin` directory.
 - b. Use the `updateBusinessSpaceWidgets` command to install, deploy, and register the designated widgets.

Setting up security for Business Space

If you are using Business Space powered by WebSphere with your environment, you must consider security options for how your team will work with artifacts in Business Space. If you want to turn on security for Business Space, set up application security and designate a user repository. To define Business Space administrators, assign a superuser role.

About this task

For best results, enable security before you configure Business Space. On the administrative console Global security administration page, you enable both administrative security and application security. You also designate a user account repository.

Considerations for using a user account registry with Business Space:

- Based on the type of LDAP configuration that you are using, your settings can impact your ability to access Business Space correctly. Make sure that the user filters, the group filters, and mapping settings are configured properly. For more information, see *Configuring Lightweight Directory Access Protocol search filters* in the WebSphere Application Server documentation.
- Based on the type of federated repository configuration that you are using, your settings can affect your ability to access Business Space correctly. Make sure that the realms are configured properly. For more information, see *Managing the realm in a federated repository configuration* in the WebSphere Application Server documentation.
- The LDAP security is set up by default to use the login property `uid` (user ID) for searching in Business Space. If your LDAP security is changed to use another unique LDAP field, such as `mail` (e-mail address) for the login property, then you must modify the `userIdKey` property in the `ConfigServices.properties` file in order for searching to work in Business Space. The `ConfigServices.properties` file is located at `profile_root\BusinessSpace\node_name\server_name\mm.runtime.prof\config\ConfigService.properties` for a stand-alone server or `deployment_manager_profile_root\BusinessSpace\`

cluster_name\mm.runtime.prof\config\ConfigService.properties for a cluster. Change the `userIdKey` attribute from `uid` to match the login property for your LDAP security, for example, `mail`. Then run the `updatePropertyConfig` command using the `wsadmin` scripting client, designating the following parameters: **-serverName** and **-nodeName** for a stand-alone server or **-clusterName** for a cluster, **-propertyFileName** with the value of the path for the `ConfigServices.properties` file, and **-prefix** with the value `Mashups_`.

- If you are using a Microsoft SQL Server database and the **Standalone LDAP** registry, make sure that the user distinguished name (user DN) does not exceed 131 characters. If any of the user DN entries exceed 131 characters, you must designate the **Federated repositories** option for the user account repository. When switching between federated repositories and other registries, all the existing spaces, pages are no longer accessible in Business Space and must be created again.
- If you are using **Federated repositories**, you have additional capabilities in your widgets and framework, such as enhanced search capabilities. When searching for users to share spaces and pages, the search scope includes e-mail, a full user name, and user ID.

If you are using IBM Tivoli Access Manager WebSEAL and want to use it with your Business Space environment, you must complete additional configuration steps. Configure Tivoli Access Manager security with an external Java Authorization Contract for Containers (JACC) provider, configure WebSEAL with Tivoli Access Manager, configure WebSEAL with your product application server, and configure host junctions for your environment.

To set up which users in the Business Space environment will be administrators, you run a script to assign the Business Space superuser role.

Setting application security for Business Space

To turn on security for Business Space you must enable both application security and administrative security.

Before you begin

Before you complete this task, you must have completed the following tasks:

- Checked that your user ID is registered in the user registry for your product.

If you expect to use a secured environment, make sure to enable security before you configure Business Space. If you want to enable or remove security after you have configured Business Space, you must modify both the `MashupAdminForOOBSpace` property and the `noSecurityAdminInternalUserOnly` property in the `ConfigServices.properties` file to set the correct user ID as the valid administrator ID. The `ConfigServices.properties` file is located at *profile_root*\BusinessSpace*node_name**server_name*\mm.runtime.prof\config\ConfigService.properties for a stand-alone server or *deployment_manager_profile_root*\BusinessSpace*cluster_name*\mm.runtime.prof\config\ConfigService.properties for a cluster. Copy the modified file into an empty folder on your system. Then run the `updatePropertyConfig` command using the `wsadmin` scripting client, designating the following parameters:

- **-serverName** and **-nodeName** for a stand-alone server or **-clusterName** for a cluster
- **-propertyFileName** with the value of the path for the `ConfigServices.properties` file
- **-prefix** with the value `Mashups_`

About this task

Business Space is preconfigured to ensure authentication and authorization of access. Users are prompted to authenticate when accessing Business Space URLs. Unauthenticated users are redirected to a login page. Business Space can be accessed by either HTTP or HTTPS, except for the login page, which always redirects to HTTPS. Therefore if using a Web server such as the IBM HTTP Server, you must configure it to support HTTPS.

Authorization to spaces and page content in Business Space is handled internally to Business Space as part of managing spaces.

To enable authenticated access to Business Space, you must have a user registry configured and application security enabled.

Procedure

1. For complete instructions on security, see the security documentation for your product.
2. For the Business Space application, on the Global security administrative console page, select both **Enable administrative security** and **Enable application security**.
3. On the same administrative console page, under **User account repository**, designate either **Federated repositories**, **Local Operating System**, **Standalone LDAP registry**, or **Standalone custom registry**. Review the considerations for selecting a user registry in Setting up security for Business Space.
4. If Business Space is remote from where your product is running, and if the node where Business Space is running and the node where your product is running are not in the same cell, you must complete manual steps to make sure that single-sign-on (SSO) is enabled. For example, if you are using more than one product (WebSphere Business Compass, WebSphere Business Monitor, WebSphere Enterprise Service Bus, or WebSphere Process Server), the servers are on different nodes, and you want them all to be able to work with the Business Space server, you must manually configure SSO. To enable SSO, complete the following steps:
 - a. On the administrative console for each server, open the Global security page by clicking **Security** → **Global security**. Expand **Web and SIP security** and click **single sign-on (SSO)** to make sure that the **Enabled** check box is selected.
 - b. Make sure that all the nodes use the same **User account repository** information (see step 3).
 - c. On the administrative console for the first node, open the Global security page. Under Authentication, click **LTPA**.
 - d. Under **Cross-cell single sign on**, type a password for the key file and a fully qualified key file name, which is a location and file name where you want to export the key file. The fully qualified key file name is the absolute path on the system where your server is running.
 - e. Click **Export keys**. The key file is saved on the system where the server is running.
 - f. If the two nodes are not on the same system, copy the key file physically to the other systems.
 - g. Import the key file on every other node using the same key file: Log on to the administrative console for the other nodes, and go to the Global security > LTPA page. Under **Cross-cell single sign on**, type the password for the

- key file and the fully qualified key file name (use the same password for the exported key file that you copied over), and click **Import keys**.
- h. Restart the server after importing keys on each system.
5. If you are using HTTPS in the endpoints file, the endpoint location is on a different node than Business Space, and the Secure Sockets Layer (SSL) certificate is a self-signed SSL certificate, you must import it.
 - a. Log on to the administrative console for the server that contains Business Space and import the SSL certificate that is used by the remote node where product is running.
 - 1) Under Security, click **SSL certificate and key management**.
 - 2) On the SSL certificate and key management page, under Related items, click **Key stores and certificates**.
 - 3) On the Key stores and certificates page, click **NodeDefaultTrustStore** to modify that truststore type.
 - 4) On the NodeDefaultTrustStore page, under Additional Properties, click **Signer certificates**.
 - 5) On the Signer certificates page for the **NodeDefaultTrustStore**, click the **Retrieve from port** button.
 - 6) On Retrieve from port page, under General Properties, type the host, port, and alias for where your product is running. Click **Retrieve signer information** button and then click **OK**.
 - 7) Restart both servers.
 - b. Log on to the administrative console for the product node and import the SSL certificate that is used by the node where Business Space is running.
 - 1) Repeat steps a. i.-v.
 - 2) On the Retrieve from port page, under General Properties, type the host and port for where Business Space is running. Click the **Retrieve signer information** button and then click **OK**.
 - 3) Restart both servers.

For more information about SSO and SSL, see the WebSphere Application Server information center.

What to do next

- After the administrative security and application security are turned on, you receive a prompt for a user ID and password when you log on to Business Space. You must use a valid user ID and password from the selected user registry in order to log on. After you turn on administrative security, whenever you return to the administrative console, you must log on with the user ID that has administrative authority.
- If you want to restrict logging in to Business Space to a subset of users and groups, you can change the mapping of the Business Space J2EE role. You must update the user/group mapping for two enterprise applications: **BSpaceEAR_node_server** and **mm.was_node_server**. Click **Applications** → **Application Types** → **WebSphere enterprise applications** and select the two applications. In the right panel, under Detail Properties, select **Security role to user/group mapping**. Remap the **businessspaceusers** and **Allauthenticated** roles from the two applications by first removing the special subject. Click **Map Special Subjects** and select **None**. Then click **Map Users** or **Map Groups** and assign each role to your selected users or groups. Note that changing the J2EE role mapping does not affect the user/group search function in Business Space.

- To set authorization to pages and spaces in Business Space, you can manage authorization when you create Business Space pages and spaces.
- **Monitor** **Process Server / ESB** To set up security for the data in the widgets based on users and groups, you must modify the mapping of users to the REST services gateway application. Select the REST services gateway application, and in the right panel, under Detail Properties, select **Security role to user/group mapping**. For the RestServicesUser role, you can add users and groups to it to control access to the data in all the REST services widgets.
- **Process Server / ESB** If you want to restrict access to data in the widgets based on user group roles, consider changing the users assigned to the administrative group roles. You can view the Roles list to see who is assigned to these roles by opening the administrative console, clicking **Security** → **Secure administration, applications, and infrastructure** → **Administrative Group Roles**, and selecting a group.

You might want to consider changing the users assigned to administrative group roles for widgets such as Business Rules and Business Variables.

For example, for the System Health widget, the following administrative roles all have monitoring permissions, all allow access to the administrative console, and therefore allow users assigned to those roles to access data in the System Health widget:

- **Monitor**
- **Configurator**
- **Operator**
- **Administrator**
- **Adminsecuritymanager**
- **Deployer**
- **iscadmins**

Users who are mapped to those administrative group roles have access to the data in the System Health widget. Users who are not mapped to those roles cannot access the data in the System Health widget.

- Finally, some widgets have an additional layer of role-based access for their artifacts created by business users. For WebSphere Process Server administration widgets, the Security Roles widget allows you to assign users and groups to system roles or module roles that determine the level of access that members have for timetables in the Business Calendars widget. For WebSphere Business Compass, the Review Access Control widget manages permissions for users who can review and comment on reviews. For more information, see the online help for your widgets.

Note:

If you find the following errors in the SystemOut.log file, you might have extra attributes in your user registry that cannot be processed:

```
00000046 SystemErr R Caused by: com.ibm.websphere.wim.exception.WIMSystemException: CWWIM1013E
    The value of the property secretary is not valid for entity uid=xxx,c=us,ou=yyy,o=ibm.com.
00000046 SystemErr R at com.ibm.ws.wim.adapter.ldap.LdapAdapter.setPropertyValue(LdapAdapter.java:3338)
```

Set the following attributes in the ConfigServices.properties file to bypass those attributes:

```
com.ibm.mashups.user.userProfile = LIMITED
com.ibm.mashups.user.groupProfile = LIMITED
```

The `ConfigServices.properties` file is located at `profile_root\BusinessSpace\node_name\server_name\mm.runtime.prof\config\ConfigService.properties` for a stand-alone server or `deployment_manager_profile_root\BusinessSpace\cluster_name\mm.runtime.prof\config\ConfigService.properties` for a cluster. After modifying the `ConfigServices.properties` file, run the `updatePropertyConfig` command using the `wsadmin` scripting client, designating the following parameters: **-serverName** and **-nodeName** for a stand-alone server or **-clusterName** for a cluster, **-propertyFileName** with the value of the path for the `ConfigServices.properties` file, and **-prefix** with the value `Mashups_`.

Note:

If you have Java 2 security enabled in a cluster, consider tightening the entry in the server policy applied to the Business Space help location.

The Business Space help location policy is:

```
grant codeBase      "file:${was.install.root}/profiles/profile_name/temp/
node_name/-" {

    permission java.security.AllPermission;

};
```

Tighten the policy by changing it to:

```
grant codeBase      "file:${was.install.root}/profiles/profile_name/temp/
node_name/server_name/BusinessSpaceHelpEAR_node_name_server_name/
BusinessSpaceHelp.war/-" {

    permission java.security.AllPermission;

};
```

Configuring Tivoli Access Manager WebSEAL to work with Business Space

If you have Tivoli Access Manager WebSEAL and you want to use it with Business Space, you must complete several additional configuration steps.

About this task

If you want to use Tivoli Access Manager WebSEAL with Business Space, you must configure Tivoli Access Manager security with an external Java Authorization Contract for Containers (JACC) provider, configure WebSEAL with Tivoli Access Manager, configure WebSEAL with your product application server, and configure host junctions for your environment.

Procedure

1. Configure Tivoli Access Manager with JACC.
 - a. Complete one of the following steps, depending on whether you want to use the administrative console or the `wsadmin` commands.
 - If you want to use the administrative console to configure Tivoli Access Manager with JACC, complete the following steps:
 - 1) Enable Global Security.
 - a) Select **Security** → **Global Security**.

- b) Enable **Administrative security**, **Application security**, and **Java 2 security** with the LDAP server with which Tivoli Access Manager is configured.
- c) Select **Global Security** → **LDAP**, enter the following information, and then click **OK**.

Name	Description
Server user Id	Enter the same user ID that you entered for the administrator DN on Tivoli Access Manager settings. Example: user1
Server user password	user1
Host	LDAP configured with Tivoli Access Manager
Port	Example: 389
Base DN	Example: o=ibm, c=us
Bind DN	Example: cn=SecurityMaster,secAuthority=Default
Bind pwd	password for SecurityMaster user

- d) Save the configuration, and restart the server.
- 2) Enable external authorization with Tivoli Access Manager and JACC.
- a) Select **Security** → **Global Security** → **External authorization providers**.
 - b) In the **Authorization provider** list, select **External JACC provider**, and then click **Configure**. The default properties for Tivoli Access Manager are correct. For default values, do not change.
 - c) Under **Additional Properties**, select **Tivoli Access Manager properties**. Select **Enable embedded Tivoli Access Manager**, enter the following information, and then click **OK**.

Name	Value
Client listening port set	The default setting is 8900 - 8999. Change it only if you want to use different ports.
Policy server (name:port)	Specify your <i>policyserver:port</i> . Example: windomain3.rtp.raleigh.ibm.com:7135
Authorization servers and priority (name:port:priority)	Specify your <i>authorizationserver:port:priority</i> . Example: windomain3.rtp.raleigh.ibm.com:7136:1
Administrator user name	Leave the user name as sec_master (default) , unless you use a different admin name on the Tivoli Access Manager server.
Administrator user password	domino123
User registry distinguished name suffix	Type the name that you want to use for your application server. Example: o=ibm,c=us

Name	Value
Security domain	Leave the Security domain set to Default . Change this setting if you are not using the default domain on the Tivoli Access Manager server. Change this setting if you have multiple domains created on the Tivoli Access Manager server and you want to connect or use a domain other than Default .
Administrator user distinguished name	Type the fully qualified name of the user. Example: cn=user1,o=ibm,c=us Note: This user is the same as the Server user ID configured in the LDAP user registry panel.

The server contacts the Tivoli Access Manager server and creates several properties files under the application server. This process might take a few minutes. If an error occurs, look in system Out and correct the problem.

- If you want to use the wsadmin utility to configure Tivoli Access Manager with JACC, complete the following steps. Perform the following procedure once on the deployment manager server. The configuration parameters are forwarded to managed servers, including node agents, when a synchronization is performed. The managed servers require their own restart for the configuration changes to take effect.
 - 1) Verify that all the managed servers, including node agents, are started.
 - 2) Start the server.
 - 3) Start the command-line utility by running the wsadmin command from the *install_root/bin* directory.
 - 4) At the wsadmin prompt, run the configureTAM command, including the appropriate information from the following table:
Jacl example:
\$AdminTask configureTAM -interactive
Jython example:
AdminTask.configureTAM('-interactive') Then type the following information:

Name	Value
node name for your product server	Specify a single node or enter an asterisk (*) to choose all nodes.
Tivoli Access Manager Policy Server	Type the name of the Tivoli Access Manager policy server and the connection port. Use the format, <i>policy_server:port</i> . The policy server communication port is set at the time of Tivoli Access Manager configuration. The default port is 7135.

Name	Value
Tivoli Access Manager Authorization Server	Type the name of the Tivoli Access Manager authorization server. Use the format <i>auth_server:port:priority</i> . The authorization server communication port is set at the time of Tivoli Access Manager configuration. The default port is 7136. You can specify more than one authorization server by separating the entries with commas. Having more than one authorization server configured is useful for failover and performance. The priority value is the order of authorization server use. For example: auth_server1:7136:1,auth_server2:7137:2. A priority of 1 is still required when configuring against a single authorization server.
administrator distinguished name for your product server	Type the full distinguished name of the security administrator ID for your product server. For example: cn=wasadmin,o=organization,c=country. For more information, see the related link.
Tivoli Access Manager user registry distinguished name suffix	For example: o=organization, c=country
Tivoli Access Manager administrator user name	Type the Tivoli Access Manager administration user ID, as created at the time of Tivoli Access Manager configuration. This ID is typically sec_master.
Tivoli Access Manager administrator user password	Type the password for the Tivoli Access Manager administrator.
Tivoli Access Manager security domain	Type the name of the Tivoli Access Manager security domain that is used to store users and groups. If a security domain is not already established at the time of Tivoli Access Manager configuration, click Return to accept the default.
Embedded Tivoli Access Manager listening port set	The product server listens on a TCP/IP port for authorization database updates from the policy server. Because more than one process can run on a particular node and machine, a list of ports is required for the processes. Specify the ports that are used as listening ports by Tivoli Access Manager clients, separated by a comma. If you specify a range of ports, separate the lower and higher values by a colon. For example, 7999, 9990:9999.
Defer	Set to yes, this option defers the configuration of the management server until the next restart. Set to no, configuration of the management server occurs immediately. Managed servers are configured on their next restart.

- 5) After you enter all the required information, select **F** to save the configuration properties or **C** to cancel from the configuration process and discard the entered information.

Example with SVTM TAM60 server:

```
wsadmin>$AdminTask configureTAM -interactive
Configure embedded Tivoli Access Manager
```

This command configures embedded Tivoli Access Manager on the WebSphere Application Server node or nodes specified.

```
WebSphere Application Server Node Name (nodeName): *
*Tivoli Access Manager Policy Server (policySvr):
  windomain3.rtp.raleigh.ibm.com:7135
*Tivoli Access Manager Authorization Servers (authSvrs):
  windomain3.rtp.raleigh.ibm.com:7136:1
*WebSphere Application Server administrator's distinguished name (wasAdminDN):
  cn=was6ladmin,o=ibm,c=us
*Tivoli Access Manager user registry distinguished name suffix (dnSuffix):
  o=ibm,c=us
Tivoli Access Manager administrator's user name (adminUid):
  [sec_master]
*Tivoli Access Manager administrator's user password (adminPasswd):
  domino123
Tivoli Access Manager security domain (secDomain): [Default]
Embedded Tivoli Access Manager listening port set (portSet): [9900:9999]
Defer (defer): [no]
```

Configure embedded Tivoli Access Manager

F (Finish)
C (Cancel)

Select [F, C]: [F] F

```
WASX7278I: Generated command line: $AdminTask configureTAM {-policySvr
  windomain3.rtp.raleigh.ibm.com:7135 -authSvrs
  windomain3.rtp.raleigh.ibm.com:7136:1 -wasAdminDN cn=wa
Embedded Tivoli Access Manager configuration action parameters saved successfully.
  Restart all WebSphere Application Server instances running on the target node or
  nodes to
wsadmin>
```

- 6) In the administrative console, select **Security** → **Global Security** → **External authorization providers**. Then select **External authorization using a JACC provider**, and click **OK**.
 - 7) Go to the main security screen and click **OK**. Save and synchronize your changes.
 - 8) Restart all processes in your cell.
- b. If you installed applications before you enabled Tivoli Access Manager (for example, you enabled LDAP security and installed some secured applications and mapped users and groups to security roles), propagate the security roles mapping information from the deployment descriptors to the Tivoli Access Manager policy server. Perform one of the following steps, depending on whether you want to use the administrative console, or the wsadmin commands.
- If you want to use the propagatePolicyToJACCProvider wsadmin command, see Propagating security policy of installed applications to a JACC provider using wsadmin scripting.
 - If you want to use the administrative console, see Propagating security policies and roles for previously deployed applications.
2. Configure WebSEAL with Tivoli Access Manager.
 - a. Ensure that WebSEAL is installed and configured properly.

- b. Create the junction between WebSEAL and your product application server using the **-c iv_creds** option for TAI++ and **-c iv_user** for TAI. Enter either of the following commands as one line, using the variables that are appropriate for your environment:

For TAI++

```
server task webseald-server create -t tcp -b supply -c iv_creds
-h host_name -p websphere_app_port_number junction_name
```

- c. To create a trusted user account in Tivoli Access Manager, which can be used for configuring TAI, issue the following commands:

```
pdadmin -a sec_master -p domino123
pdadmin sec_master> user create -gsouser -no-password-policy taiuser
"cn=taiuser,ou=websphere,o=ibm,c=us" taiuser taiuser ptaiuser
pdadmin sec_master> user modify taiuser password-valid yes
pdadmin sec_master> user modify taiuser account-valid yes
```

- d. In the WebSEAL configuration file *webseal_install_directory/etc/webseald-default.conf*, set the following parameter:

```
basicauth-dummy-passwd=webseal_userid_passwd
```

For example, if you set the taiuser/ptaiuser in Tivoli Access Manager, set the following parameter: `basicauth-dummy-passwd = ptaiuser`

If you are using a form-based authentication, set the following parameters:

```
forms-auth=both
ba-auth=none
```

3. Configure WebSEAL with your product application server by enabling the TAI++ interceptor on the server.
 - a. In the administrative console, select **Global security** → **Authentication mechanisms and expiration**.
 - b. Expand **Web and SIP security**, and then select **Trust Association**. Select the check box and click **Apply**.
 - c. Select **Interceptors** → **TAMTrustAssociationInterceptorPlus** → **custom properties**, and add the following properties:

Name	Value
com.ibm.websphere.security.webseal.configURL	\${WAS_INSTALL_ROOT}/java/jre/PdPerm.properties
com.ibm.websphere.security.webseal.id	iv-creds
com.ibm.websphere.security.webseal.loginId	taiuser (if the user taiuser/ptaiuser was created in the Tivoli Access Manager)

- d. Restart the cell.
- e. To access the client, go to `https://webseal_server_name:webseal_port/junction_name/web_uri_for_client`.
4. Configure the host junctions for your environment, so that the Business Space widgets appear. Complete one of the following steps, depending on whether you are using virtual host junctions or transparent host junctions.
 - If you are using virtual host junctions, create a virtual host junction. A virtual host junction eliminates the need to create separate junctions.
 - a. Make sure that a virtual host has been configured. Virtual host junctions match a host and port number and forward addresses to the target host. No URL filtering occurs, and all requests that match are forwarded to the target host.

- b. Make sure that the following applications are available to the same virtual host. You may have some or all of the applications, based on which products you are using with Business Space.
- BPMAdministrationWidgets_*nodename_servername* (for WebSphere Enterprise Service Bus and WebSphere Process Server)
 - BusinessSpaceHelpEAR_*nodename_servername* (for all products)
 - BSpaceEAR_*nodename_servername* (for all products)
 - BSpaceWebformsEnabler_*nodename_servername* (for all products)
 - HumanTaskManagementWidgets_*nodename_servername* (for WebSphere Process Server and WebSphere Business Monitor)
 - REST Services Gateway (for all products)
 - REST Services Gateway Dmgr (for WebSphere Enterprise Service Bus and WebSphere Process Server)
 - mm.was_*nodename_servername* (for all products)
 - WBMDashboardWeb_*nodename_servername* (for WebSphere Business Monitor)
 - webWidgets_*nodename_servername* (for WebSphere Enterprise Service Bus)
 - widgets_busleader_*nodename_servername* (for WebSphere Business Compass)
 - widgets_pubserver_*nodename_servername* (for WebSphere Business Compass)
 - widgets_fabric_*nodename_servername* (for WebSphere Business Services Fabric)

Note: This list of applications covers only the applications required by Business Space. You might need to add other applications to the list for non-Business Space scenarios using Tivoli Access Manager WebSEAL.

- c. Run the following command using pdadmin: `server task webseal server virtualhost create -t transport -h target_host [-p port] [-v virtual_host_name] virtual_host_label`

Use the following information:

- *webseal server* is the name of the WebSEAL server where you will create the virtual host entry.
- *transport* is the type of transport. Valid entries are `tcp`, `ssl`, `tcpproxy`, and `sslproxy`.
- *target_host* is the host of the required application.
- *virtual_host_name* is used to match HTTP requests to a virtual host junction. If no value is entered, it is made up of the target host and port by default. For example, if you set the *virtual_host_name* to `myvirthost.ibm.com:80`, WebSEAL matches the URLs containing `myvirthost.ibm.com:80` and routes it to the host provided in the pdadmin command.
- *virtual_host_label* is the label used to identify the entry in WebSEAL. It must be unique.

For Business Space to run as expected, both `ssl` and `tcp` entries must be created for the type of transport. When you need both Secure Sockets Layer (SSL) and Transmission Control Protocol (TCP) to be supported in the same virtual host junction, you must use the `-g vhost_label` option, where *vhost_label* is the original virtual host label to share configuration. This option finds a previously created virtual host junction (one created

earlier, where the *virtual_host_label* matches the label provided in the `-g` option), and will share that configuration. The second entry still needs its own *virtual_host_label*, but it can share the target host, port, and other values. If you do not provide this `-g` option, a second virtual host cannot be created because WebSEAL will see the target host and port as being identical to a previously create junction (which is not allowed).

- If you are using transparent host junctions, create a series of transparent path junctions for the widgets for each product.
 - a. Run the following command using `pdadmin`: `server task webseal server create -t transport type (ssl) or (tcp) -x -h hostname path`
For example, type: `server task webseald-default create -t tcp -x -h monServer.ibm.com /BusinessSpace`.
 - b. Create the following context roots for your product: Mapping Business Space URLs for a reverse proxy server.
- 5. Complete additional configuration steps to resolve issues with browser cookies and virtual hosts.
 - a. To resolve renaming of the Business Space cookie, add the following content to the WebSEAL configuration file:
[preserve-cookie-names]
name = com.ibm.bspace.UserName
name = com.ibm.wbimonitor.UserName
 - b. Optional: If you are using non-default virtual hosts with a context root, you might encounter issues with Business Space pages. You might need to stop the junction from rewriting the JavaScript™ on the Business Space pages by adding the `-j` junction to the context root. Run the following command:
`server task default-webseald create -f -h hostname -p portnumber -t tcp -b supply -c iv-user,iv-creds,iv-groups -x -s -j -J trailer/root context`

Assigning the Business Space superuser role

In Business Space, you can assign users to be superusers (or Business Space administrators). A superuser can view, edit, and delete all spaces and pages, can manage and create templates, and can change ownership of a space by changing the owner ID.

Before you begin

If administrative security is enabled when you configure Business Space, consider the following information about groups and superusers:

- Users belonging to the special user group, **administrators**, have a superuser role by default. As a result, the superuser role assignment is handled by user group membership.
- In a single-server environment, the Business Space server creates the **administrators** user group in the default user registry. The administrator ID provided during configuration is automatically added as member of this group.
- In a network deployment environment, the **administrators** user group is not created automatically. Use the `createSuperUser.py` script to create the user group and add members to that group in the default user registry.
- If another user registry (for example, LDAP) is used instead of the default user registry, or if the default user registry is used but you do not want to use the **administrators** user group, you must identify the user group that you are using for the Business Space superusers. Make sure that the value you provide can be understood by the user registry. For example, for LDAP, you might provide a

name like `cn=administrators,dc=company,dc=com`. For more information about identifying this user group, see the instructions for changing the administrators group in the What to do next section.

- For Business Space in WebSphere Portal, the default group **wpsadmins** is also used for the superuser role. Members of this group are granted the superuser role for Business Space.

Note: Security must be enabled if you want to use Business Space in WebSphere Portal.

If administrative security is not enabled when you configure Business Space, only the special user ID **BPMAdministrator** has the Business Space superuser role.

If you have a network deployment environment, you must run the `createSuperUser.py` script to assign the superuser role: to create the user group and add members. Before you run the script, complete the following steps:

- Make sure the default **administrators** group name is not changed.
- Use the default repository for the user registry.
- Start the server or the deployment manager for your Business Space environment for the profile where is Business Space installed.

Procedure

1. Locate the script `install_root\BusinessSpace\scripts\createSuperUser.py` for assigning the superuser role to a user.
2. Open a command prompt, and change directories to the following directory: `profile_root\bin`, where `profile_root` represents the directory for the profile where is Business Space installed.
3. Type the following command: `wsadmin -lang jython -f install_root\BusinessSpace\scripts\createSuperUser.py user_short_name password` where `user_short_name` is the unique identifier for a user in Virtual Member Manager (VMM), and `password` is the VMM password for that user. If that user exists in VMM, the user is added to the administrator group.

Note: When the path contains a space, for example, if `install_root` is `My install dir`, you must enclose the path names in quotation marks. For example, type the following command: `wsadmin -lang jython -f "\My install dir\BusinessSpace\scripts\createSuperUser.py" user_short_name_in_VMM`.

What to do next

To open Business Space, use the following URL: `http://host:port/BusinessSpace`, where `host` is the name of the host where your server is running and `port` is the port number for your server.

You can change the default special user group named **administrators**. Perform the following steps to check the current group name or change it to other name.

Inspect the value for the metric `com.ibm.mashups.adminGroupName` in the configuration file:

- `profile_root\BusinessSpace\node_name\server_name\mm.runtime.prof\config\ConfigService.properties` on a stand-alone server, or
- `deployment_manager_profile_root\BusinessSpace\cluster_name\mm.runtime.prof\config\ConfigService.properties` on a cluster.

If you want to change an administrative group, perform the following steps on a stand-alone server:

1. Modify the metric `com.ibm.mashups.adminGroupName` in the configuration file `profile_root\BusinessSpace\node_name\server_name\mm.runtime.prof\config\ConfigService.properties`.
2. Run the command `updatePropertyConfig` in the `wsadmin` environment of the profile:
`$AdminTask updatePropertyConfig {-serverName server_name -nodeName node_name -propertyFileName "profile_root\BusinessSpace\node_name\server_name\mm.runtime.prof\config\ConfigService.properties" -prefix "Mashups_"}` and run `$AdminConfig save`.
3. Restart the server.

If you want to change an administrative group, perform the following steps on a cluster:

1. Modify the metric `com.ibm.mashups.adminGroupName` in the configuration file `deployment_manager_profile_root\BusinessSpace\cluster_name\mm.runtime.prof\config\ConfigService.properties`.
2. Run the command `updatePropertyConfig` in the `wsadmin` environment of the deployment environment profile:
`$AdminTask updatePropertyConfig {-clusterName cluster_name -propertyFileName "deployment_manager_profile_root\BusinessSpace\cluster_name\mm.runtime.prof\config\ConfigService.properties" -prefix "Mashups_"}` and run `$AdminConfig save`.
3. Restart the deployment manager.

If you want to change the superuser when security is not enabled, perform the following steps on a stand-alone server:


1. Modify the metric `noSecurityAdminInternalUserOnly` in the configuration file `profile_root\BusinessSpace\node_name\server_name\mm.runtime.prof\config\ConfigService.properties`.
2. Run the command `updatePropertyConfig` in the `wsadmin` environment of the profile:
`$AdminTask updatePropertyConfig {-serverName server_name -nodeName node_name -propertyFileName "profile_root\BusinessSpace\node_name\server_name\mm.runtime.prof\config\ConfigService.properties" -prefix "Mashups_"}` and run `$AdminConfig save`.
3. Restart the server.

If you want to change the superuser when security is not enabled, perform the following steps on a cluster:

1. Modify the metric `noSecurityAdminInternalUserOnly` in the configuration file `deployment_manager_profile_root\BusinessSpace\cluster_name\mm.runtime.prof\config\ConfigService.properties`.
2. Run the command `updatePropertyConfig` in the `wsadmin` environment of the deployment environment profile:
`$AdminTask updatePropertyConfig {-clusterName cluster_name -propertyFileName "deployment_manager_profile_root\BusinessSpace\cluster_name\mm.runtime.prof\config\ConfigService.properties" -prefix "Mashups_"}` and run `$AdminConfig save`.
3. Restart the deployment manager.

Commands (wsadmin scripting) for configuring Business Space

Look up a scripting object or command class to find details about its command syntax.

To open the information center table of contents to the location of this reference information, click the **Show in Table of Contents** button () on your information center border.

configureBusinessSpace command

Use the configureBusinessSpace command to configure the database for Business Space powered by WebSphere.

This command configures the data source for Business Space and generates the scripts that create and configure database tables.

Required parameters

-serverName *server_name*

A parameter that specifies the server name for the configuration. For configuring Business Space on a server, you must specify both a **serverName** and a **nodeName**.

-nodeName *node_name*

A parameter that specifies the node name for the configuration. For configuring Business Space on a server, you must specify both a **serverName** and a **nodeName**.

-clusterName *cluster_name*

A parameter that specifies the cluster name for the configuration. For configuring Business Space on a cluster, you must specify a **clusterName**.

Optional parameters

-schemaName *schema_name*

An optional parameter that specifies the database schema for the Business Space database configuration. The default value is IBMBUSSP.

-tablespaceDir *table_space_path*

An optional parameter that specifies a directory path or file name prefix for the files used as the physical locations of table spaces. The default value is BSP. Valid for DB2, Oracle and SQL Server (otherwise ignored). For SQL Server, this parameter applies to the primary data file and log files.

-tablespaceNamePrefix *table_space_prefix*

An optional parameter that specifies a prefix string added to the beginning of table space names to make them unique. The default value is BSP. If a table space name prefix is longer than four characters, it is truncated to four characters. Valid for DB2, DB2 z/OS V8, DB2 z/OS V9, and Oracle (otherwise ignored).

-dbLocationName *database_location_name*

An optional parameter that specifies the database location name on z/OS. The default value is BSP or the product database name. Valid on DB2 z/OS V8 and V9 (otherwise ignored).

-storageGroup *storage_group*

An optional parameter that specifies the storage group on z/OS for Business Space. If you are using z/OS, you must update the database scripts that are

generated before running them. For more information about the scripts, see "Configuring Business Space database tables."

-bspacedbDesign *database_design_file_name*

An optional parameter that specifies a database design file that you are using to define all database configuration information, including the schema, and the table space directory. If you designate a database design file using the **-bspacedbDesign** parameter, you do not need to designate the **-schemaName**, **-tablespaceDir**, or **-storageGroup** parameters, unless you want to override what is in the database design file for particular database configuration information.

Note: The JNDI name of jdbc/mashupDS is always used for a Business Space data source, so the JNDI name in the database design file is not used. If a data source with a JNDI name of jdbc/mashupDS exists, this command stops without configuring the profile unless you also specify the **-replaceDatasource true** parameter.

-productTypeForDatasource *product_database*

An optional parameter that specifies properties to use to create the data source to use with Business Space. Designating a **productTypeForDatasource** creates a data source for Business Space with a JNDI name of jdbc/mashupDS that is modeled on the data source of an installed product, such as WebSphere Process Server, WebSphere Enterprise Service Bus, WebSphere Business Monitor, and WebSphere Business Compass. Valid values are WPS (to designate WebSphere Process Server or WebSphere Enterprise Service Bus), WPBS (to designate WebSphere Business Compass), and WBM (to designate WebSphere Business Monitor). If the **bspacedbDesign** parameter is also specified, the **productTypeForDatasource** overrides the database type and JDBC provider, and the JNDI name in the database design file is not used.

Note: If a data source with a JNDI name of jdbc/mashupDS exists, this command stops without configuring the profile unless you also specify the **-replaceDatasource true** parameter.

-replaceDatasource true | false

An optional parameter that specifies whether the **configureBusinessSpace** command runs if the profile has already been configured. The default value is **false**. When a profile is configured for Business Space, a data source with a JNDI name of jdbc/mashupDS is created. If the data source exists and you run the **configureBusinessSpace** command without specifying **-replaceDatasource true**, the command does not change the configuration. If you specify **true**, the command deletes the data source and its JDBC provider, creates new ones, and creates new DDL scripts.

-save true | false

A parameter that indicates saving your configuration changes. The default value is **false**.

Examples

The following example uses the **configureBusinessSpace** command to configure a Business Space data source on a server.

- Jython example:

```
AdminTask.configureBusinessSpace(['-nodeName myNode -serverName myServer'])
```

- Jacl example:

```
$AdminTask configureBusinessSpace {-nodeName myNode -serverName  
myServer}
```

The following example uses the `configureBusinessSpace` to configure a Business Space data source on a cluster and save the changes.

- Jython example:

```
AdminTask.configureBusinessSpace('[-clusterName myCluster -save  
true]')
```

- Jacl example:

```
$AdminTask configureBusinessSpace {-clusterName myCluster -save  
true}
```

The following example uses the `configureBusinessSpace` to configure a Business Space data source on a cluster, with a schema name and a product data source designated for WebSphere Process Server.

- Jython example:

```
AdminTask.configureBusinessSpace('[-clusterName myCluster  
-schemaName myCluster -productTypeForDatasource WPS -save true]')
```

- Jacl example:

```
$AdminTask configureBusinessSpace {-clusterName myCluster  
-schemaName myCluster -productTypeForDatasource WPS -save true}
```

The following example uses the `configureBusinessSpace` to configure a Business Space data source on a cluster using database information that is in the database design file.

- Jython example:

```
AdminTask.configureBusinessSpace('[-clusterName myCluster  
-bspacedbDesign "C:\Bspace_dbDesign.properties" -save true]')
```

- Jacl example:

```
$AdminTask configureBusinessSpace {-clusterName myCluster  
-bspacedbDesign "C:\Bspace_dbDesign.properties" -save true}
```

configureLotusWebformServer command

Use the `configureLotusWebformServer` command to configure Business Space to use IBM Lotus WebForm Server. Lotus Webform Server works with Human Task Management widgets and applies to WebSphere Process Server servers and clusters and any business process management product installation that includes WebSphere Process Server.

The `configureLotusWebformServer` command configures Business Space to use IBM Lotus WebForm Server to work with Human Task Management widgets. Webform Server must be installed on the same machine where you are running the script.

Required parameters

-serverName *server_name*

A parameter that specifies the server name for the configuration. For configuring Business Space widgets on a server, you must specify both a **serverName** and a **nodeName**.

-nodeName *node_name*

A parameter that specifies the node name for the configuration. Either a **serverName**, **nodeName**, or **clusterName** is required. For configuring on a server, you must specify both a **serverName** and a **nodeName**.

-clusterName *cluster_name*

A parameter that specifies the cluster name for the configuration. For configuring Business Space on a cluster, you must specify a **clusterName**.

-translatorHTTPLocation *URL*

A parameter that specifies the location of the Webform Server Translator. The default URL for the location is `http://localhost:8085/translator`.

-serverInstallRoot *Webform_Server_install_root*

A parameter that specifies the full path where Lotus Webform Server is installed. For example, the Lotus Webform Server install root might be `C:/IBM/LotusWebForms/3.5/WebFormServer`

Optional parameters

-save true | false

A parameter that indicates saving your configuration changes. The default value is true.

Examples

The following example uses the `configureLotusWebformServer` to configure Business Space to use Lotus WebForm Server with the Human Task Management widgets.

- Jython example:

```
AdminTask.configureLotusWebformServer(['-nodeName node_name  
-serverName server_name -translatorHTTPLocation  
http://localhost:9080/translator -serverInstallRoot  
C:/IBM/LotusWebForms/3.5/WebFormServer'] )
```

- Jacl example:

```
AdminTask configureLotusWebformServer {-nodeName node_name  
-serverName server_name -translatorHTTPLocation  
http://localhost:9080/translator  
-serverInstallRoot C:/IBM/LotusWebForms/3.5/WebFormServer}
```

getBusinessSpaceDeployStatus command

Use the `getBusinessSpaceDeployStatus` command to check whether Business Space powered by WebSphere is configured on a particular deployment target.

This command checks whether Business Space is configured on a server, node, or cluster that you specify. If you don't set any parameters, it checks if Business Space is configured in the cell.

Required parameters

-serverName *server_name*

A parameter that specifies the server name to check for Business Space.

-nodeName *node_name*

A parameter that specifies the node name to check for Business Space.

-clusterName *cluster_name*

A parameter that specifies the cluster name to check for Business Space.

Examples

The following example uses the `getBusinessSpaceDeployStatus` command to check whether Business Space is configured on a server.

- Jython example:

```
AdminTask.getBusinessSpaceDeployStatus(['-nodeName myNode -serverName myServer'])
```

- Jacl example:

```
$AdminTask getBusinessSpaceDeployStatus {-nodeName myNode -serverName myServer}
```

The following example uses the `getBusinessSpaceDeployStatus` command to check whether Business Space is configured on a cluster.

- Jython example:

```
AdminTask.getBusinessSpaceDeployStatus(['-clusterName myCluster'])
```

- Jacl example:

```
$AdminTask getBusinessSpaceDeployStatus {-clusterName myCluster}
```

The following example uses the `getBusinessSpaceDeployStatus` command to return a list of all deployment targets (server and clusters) configured for Business Space in a cell.

If you run the command from the profile root `bin` directory, the command returns a list of all deployment targets (server and clusters) configured for Business Space in a cell.

If you run the command from the installation root `bin` directory, the command returns a list of all deployment targets (server and clusters) configured for Business Space in the same installation root directory.

- Jython example:

```
AdminTask.getBusinessSpaceDeployStatus()
```

- Jacl example:

```
$AdminTask getBusinessSpaceDeployStatus
```

installBusinessSpace command

Use the `installBusinessSpace` command to set up Business Space powered by WebSphere on your runtime environment.

The `installBusinessSpace` command installs the Business Space enterprise archive (EAR) files in your runtime environment.

Required parameters

-serverName *server_name*

A parameter that specifies the server name for the configuration. For configuring Business Space on a server, you must specify both a **serverName** and a **nodeName**.

-nodeName *node_name*

A parameter that specifies the node name for the configuration. Either a **serverName**, **nodeName**, or **clusterName** is required. For configuring Business Space on a server, you must specify both a **serverName** and a **nodeName**.

-clusterName *cluster_name*

A parameter that specifies the cluster name for the configuration. For configuring Business Space on a cluster, you must specify a **clusterName**.

Optional parameters

-noWidgets *true | false*

An optional parameter that if set to `true` prevents the product widgets from

being installed on the deployment target. Then, if you want to install widgets, you must use the `installBusinessSpaceWidgets` command after the Business Space configuration has completed successfully. The default value is `false`.

-save true | false

An optional parameter that indicates saving your configuration changes. The default value is `false`.

Examples

The following example uses the `installBusinessSpace` command to install Business Space EAR files on a server.

- Jython example:

```
AdminTask.installBusinessSpace('[-nodeName myNode -serverName  
myServer -save true]')
```

- Jacl example:

```
$AdminTask installBusinessSpace {-nodeName myNode -serverName  
myServer -save true}
```

The following example uses the `installBusinessSpace` to install Business Space EAR files on a cluster.

- Jython example:

```
AdminTask.installBusinessSpace('[-clusterName myCluster -save true]')
```

- Jacl example:

```
$AdminTask installBusinessSpace {-clusterName myCluster -save true}
```

installBusinessSpaceWidgets command

Use the `installBusinessSpaceWidgets` command to install, deploy and register widgets for use with Business Space powered by WebSphere.

The `installBusinessSpaceWidgets` command installs, deploys, and registers designated widgets contained in a compressed file or an enterprise archive (EAR) file. If widgets are already deployed, the `installBusinessSpaceWidgets` command refreshes the binary and registration information.

The structure of the widget compressed file contains the following items:

- [ear\widgets_*name*.ear] one or more EAR files.
- [catalog\catalog_*name*.xml]
- [endpoints*.xml] widget endpoints
- [templates*.zip] Templates must be in a compressed file and follow IBM Lotus Mashups template format.
- [help\eclipse\plugins*]

All folders are not required. Empty folders are valid.

Required parameters

-serverName *server_name*

A parameter that specifies the server name for the configuration. For configuring Business Space on a server, you must specify both a **serverName** and a **nodeName**.

-nodeName *node_name*

A parameter that specifies the node name for the configuration. Either a

serverName, nodeName, or clusterName is required. For configuring Business Space widgets on a server, you must specify both a **serverName** and a **nodeName**.

-clusterName *cluster_name*

A parameter that specifies the cluster name for the configuration. For configuring Business Space widgets on a cluster, you must specify a **clusterName**.

-widgets *widgets_path*

A parameter that specifies one of the following:

- the full path for the directory that contains the compressed files or the EAR files that contain the widgets. If you specify a directory, all widgets will be installed for all compressed files and EAR files in that directory.
- the full path to an individual compressed file that contains the widgets.
- the full path to an individual EAR file that contains the widgets.

-save true | false

A parameter that indicates saving your configuration. The default value is true.

Optional parameters

-save true | false

A parameter that indicates saving your configuration. The default value is true.

Examples

The following example uses the `installBusinessSpaceWidgets` to install, deploy, and register widgets on a server.

- Jython example:

```
AdminTask.installBusinessSpaceWidgets(['-nodeName node_name  
-serverName server_name -widgets  
install_root\BusinessSpace\widgets\MyWidget.zip'])
```

- Jacl example:

```
$AdminTask installBusinessSpaceWidgets {-nodeName node_name  
-serverName server_name -widgets  
install_root\BusinessSpace\widgets\MyWidget.zip}
```

The following example uses the `installBusinessSpaceWidgets` to install, deploy, and register widgets on a cluster.

- Jython example:

```
AdminTask.installBusinessSpaceWidgets(['-clusterName cluster_name  
-widgets X:\WPS\Temp'])
```

- Jacl example:

```
$AdminTask installBusinessSpaceWidgets {-clusterName cluster_name  
-widgets X:\WPS\Temp}
```

Manual steps are required for updating Business Space templates and spaces after running the `installBusinessSpaceWidgets` or `updateBusinessSpaceWidgets` command. For more information, see [Updating Business Space templates and spaces after installing or updating widgets](#).

registerRESTServiceEndpoint command

Use the registerRESTServiceEndpoint command to register configured and enabled Representational State Transfer (REST) endpoints so that your team can use the widgets in Business Space.

This command registers the REST service endpoints so that Business Space is properly connected to widgets for your product. This command registers the endpoints of the REST services that are in the same cell as Business Space.

Required parameters

-clusterName *name_of_rest_services_cluster*

A parameter that specifies the cluster name for the REST service. When registering REST services endpoints for a cluster, you must specify a **clusterName**.

-nodeName *name_of_rest_services_node*

A parameter that specifies the node name for the REST service. When registering REST services endpoints for a server, you must specify both a **serverName** and a **nodeName**.

-serverName *name_of_rest_services_server*

A parameter that specifies the server name for the REST service. When registering REST services endpoints for a server, you must specify both a **serverName** and a **nodeName**.

-businessSpaceClusterName *name_of_business_space_cluster*

The Business Space cluster name. If Business Space is configured on a cluster, you must specify a **businessSpaceClusterName**.

-businessSpaceNodeName *name_of_business_space_node*

The Business Space node name. If Business Space is configured on a server, you must specify both a **businessSpaceServerName** and a **businessSpaceNodeName**.

-businessSpaceServerName *name_of_business_space_server*

The Business Space server name. If Business Space is configured on a server, you must specify both a **businessSpaceServerName** and a **businessSpaceNodeName**.

Optional parameters

-appName *name_of_provider_application*

The application name of the REST service provider.

-type *name_of_service_type*

The type of the service. This parameter is optional. If this parameter is not specified, all unique REST service endpoints configured for a specified REST service provider on a specified deployment target are registered. If you want to specify a specific service endpoint, use the `<tns:type>` value that is in the endpoints file for a widget. The endpoints files are located at `install_root\BusinessSpace\registryData\endpoints` directory. For example, `bpmAdministrationEndpoints.xml` contains all service endpoint types that are used by Administration widgets. The value of the `<tns:type>` element is `{com.ibm.bpm}SCA`:

```
<tns:Endpoint>
  <tns:id>{com.ibm.bpm}SCA</tns:id>
  <tns:type>{com.ibm.bpm}SCA</tns:type>
  <tns:version>6.2.0.0</tns:version>
  <tns:url>/rest/sca/v1</tns:url>
```

```
<tns:description>Location backend SCA REST Services
for Module Administration widgets and Service Monitoring widget
</tns:description>
</tns:Endpoint>
```

For Jacl, make sure to use double quotes around the value, for example: ...
-type "{com.ibm.bpm}SCA"

-webModuleName *name_of_web_module*
The web module name of the REST service provider.

-version *name_of_version*
The version of the REST service provider.

Examples

The following example uses the registerRESTServiceEndpoint command. It registers all configured and enabled REST services on the cluster with Business Space.

- Jython example:

```
AdminTask.registerRESTServiceEndpoint('[-clusterName
name_of_rest_services_cluster -businessSpaceClusterName
name_of_business_space_cluster]')
```

- Jacl example:

```
$AdminTask registerRESTServiceEndpoint {-clusterName
name_of_rest_services_cluster -businessSpaceClusterName
name_of_business_space_cluster}
```

uninstallBusinessSpaceWidgets command

Use the uninstallBusinessSpaceWidgets command to remove widgets and widget definitions from the profile, including removing individual widget assets (application, catalog, endpoints, spaces, templates, help).

The uninstallBusinessSpaceWidgets command removes widget files in a designated compressed file or an enterprise archive (EAR) file. The structure of the widget compressed file contains the following items:

- [ear\widgets_*name*.ear] one or more EAR files.
- [catalog\catalog_*name*.xml]
- [endpoints*.xml] widget endpoints
- [templates*.zip] Templates must be in a compressed file and follow IBM Lotus Mashups template format.
- [help\eclipse\plugins*]

All folders are not required. Empty folders are valid.

Note: If you customized REST endpoint information outside of using the updateBusinessSpaceWidgets command, those endpoint changes are lost after running the uninstallBusinessSpaceWidgets command.

Required parameters

-serverName *server_name*
A parameter that specifies the server name for the configuration. For configuring Business Space on a server, you must specify both a **serverName** and a **nodeName**.

-nodeName *node_name*

A parameter that specifies the node name for the configuration. For configuring Business Space on a server, you must specify both a **serverName** and a **nodeName**.

-clusterName *cluster_name*

A parameter that specifies the cluster name for the configuration. For configuring Business Space on a cluster, you must specify a **clusterName**.

-widgets *widgets_path*

A parameter that specifies one of the following:

- the full path for the directory that contains the compressed files or the widget EAR files that contain the widgets. If you specify a directory, all widgets will be installed for all compressed files and EAR files in that directory.
- the full path to an individual compressed file that contains the widgets.
- the full path to an individual EAR file that contains the widgets.

Optional parameters

-save **true** | **false**

A parameter that indicates saving your configuration changes. The default value is true.

Example

The following example uses the `uninstallBusinessSpaceWidgets` command to remove widgets from a cluster.

- Jython example:

```
AdminTask.uninstallBusinessSpaceWidgets(['-clusterName  
cluster_name -widgets X:\WPS\Temp'])
```

- Jacl example:

```
$AdminTask uninstallBusinessSpaceWidgets {-clusterName  
cluster_name -widgets X:\WPS\Temp}
```

updateBusinessSpaceWidgets command

Use the `updateBusinessSpaceWidgets` command to update previously configured Business Space widgets and their endpoints, catalogs, templates, and help plugins.

The `updateBusinessSpaceWidgets` command updates widget binary files, catalog files, endpoint files, templates, and help plug-ins for widgets that have been previously installed and configured for Business Space.

The `updateBusinessSpaceWidgets` command updates widget files in a designated compressed file or an enterprise archive (EAR) file. The structure of the widget compressed file contains the following items:

- [ear\widgets_name.ear] one or more EAR files.
- [catalog\catalog_name.xml]
- [endpoints*.xml] widget endpoints
- [templates*.zip] Templates must be in a compressed file and follow IBM Lotus Mashups template format.
- [help\eclipse\plugins*]

All folders are not required. Empty folders are valid.

Required parameters

-serverName *server_name*

A parameter that specifies the server name for the configuration. For configuring Business Space widgets on a server, you must specify both a **serverName** and a **nodeName**.

-nodeName *node_name*

A parameter that specifies the node name for the configuration. Either a **serverName**, **nodeName**, or **clusterName** is required. For configuring Business Space widgets on a server, you must specify both a **serverName** and a **nodeName**.

-clusterName *cluster_name*

A parameter that specifies the cluster name for the configuration. For configuring Business Space on a cluster, you must specify a **clusterName**.

Optional parameters

-widgets *widget_path*

A parameter that specifies the full path for the directory where widget enterprise archive (EAR) files or widget compressed files are located or the full path to a specific EAR file or widget compressed file.

-endpoints *endpoint_path*

A parameter that specifies the full path for the directory where the widget endpoint files are located or the full path to a specific endpoint file.

-catalogs *catalog_path*

A parameter that specifies the full path for the directory that contains the widget catalog files or the full path to a specific catalog file.

-templates *template_path*

A parameter that specifies the full path for the directory that contains the widget template files or the full path to a specific template file.

-helpplugins *help_path*

A parameter that specifies the full path for the directory that contains the widget online help plugin files or the full path to a specific widget online help plugin file.

-noWidgets true | false

Specifies that you do not want to update the widget EAR files that are contained within the widgets compressed file.

-noEndpoints true | false

Specifies that you do not want to update the specified endpoint files that are contained in the widgets compressed file.

-noCatalogs true | false

Specifies that you do not want to update the catalog definition files that are contained in the widgets compressed file.

-noTemplates true | false

Specifies that you do not want to update the templates that are contained in the widgets compressed file.

-noHelp true | false

Specifies that you do not want to update the help files that are contained in the widgets compressed file.

-save true|false

A parameter that indicates saving your configuration. The default value is true.

Examples

The following example uses the `updateBusinessSpaceWidgets` to update widgets on a cluster.

Jacl example: Jython example:

```
AdminTask.updateBusinessSpaceWidgets(['-clusterName cluster_name  
-widgets widget_path'])
```

```
$AdminTask updateBusinessSpaceWidgets {-clusterName cluster_name  
-widgets widget_path}
```

The following example uses the `updateBusinessSpaceWidgets` to update widgets on a server.

Jython example:

```
AdminTask.updateBusinessSpaceWidgets(['-nodeName node_name  
-serverName server_name -widgets widget_path'])
```

Jacl example:

```
$AdminTask updateBusinessSpaceWidgets {-nodeName node_name  
-serverName server_name -widgets widget_path}
```

Manual steps are required for updating Business Space templates and spaces after running the `installBusinessSpaceWidgets` or `updateBusinessSpaceWidgets` command. For more information, see [Updating Business Space templates and spaces after installing or updating widgets](#).

updateRESTGatewayService command

Use the `updateRESTGatewayService` command to update a Representational State Transfer (REST) gateway service so that REST services are configured and enabled.

This command updates the REST Gateway service so that REST services are configured and enabled. The deployment of the REST services is performed automatically in a stand-alone server profile. For other types of configurations, the REST Services administrative console page or the `updateRESTGatewayService` allows you to configure REST services for all of your product's widgets in Business Space.

Note: For WebSphere Process Server, Business Process Choreographer and Human Task Management REST services are configured when you configure the Business Process Choreographer and Human Task Management containers.

Required parameters

-clusterName *cluster_name*

A parameter that specifies the cluster name for the REST service. For configuring REST services on a cluster, you must specify a **clusterName**.

-nodeName *node_name*

A parameter that specifies the node name for the REST service. For configuring REST services on a server, you must specify both a **serverName** and a **nodeName**.

-serverName *server_name*

A parameter that specifies the server name for the REST service. For configuring REST services on a server, you must specify both a **serverName** and a **nodeName**.

-enable true | false

Indicates if the REST service is enabled. Valid values include true or false.

Optional parameters

-type *name_of_service_type*

The type of the REST service.

-version *name_of_version*

The version of the REST service.

Examples

The following example uses the `updateRESTGatewayService` command to update the REST Gateway service so that REST services are configured and enabled.

- Jython example:

```
AdminTask.updateRESTGatewayService(['-nodeName node1 -serverName
server1 -type "{com.ibm.bpm}TimeTable" -version 6.2.0.0 -enable
true'])
```

- Jacl example:

```
$AdminTask updateRESTGatewayService {-nodeName node1 -serverName
server1 -type "{com.ibm.bpm}TimeTable" -version 6.2.0.0 -enable true}
```

Updating Business Space templates and spaces after installing or updating widgets

Manual steps are required for updating Business Space templates and spaces after running the `installBusinessSpaceWidgets` or `updateBusinessSpaceWidgets` commands in a clustered environment.

Before you begin

You must complete the following additional steps if you have previously used the `installBusinessSpaceWidgets` command or the `updateBusinessSpaceWidgets` command.

Procedure

1. If Business Space is configured in a cluster, perform the following steps:
 - a. Identify the custom profile for `oobLoadedStatus` properties file:
 - 1) In deployment manager profile, open the `deployment_manager_profile_root\BusinessSpace\cluster_name\mm.runtime.prof\config\ConfigService.properties` file.
 - 2) Look for the name of cell, node and server in the `com.ibm.mashups.directory.templates` or `com.ibm.mashups.directory.spaces` properties.
For example, in `com.ibm.mashups.directory.templates = config/cells/Cell01/nodes/Node01/servers/Server1/mm/templates`, you can locate the custom profile by the Cell01 cell name and the Node01 node name.
 - 3) Use the name of cell, node and server to locate the custom profile.

- b. In the custom profile, open the *custom_profile_root\BusinessSpace\cluster_name\mm.runtime.prof\public\oobLoadedStatus.properties* file and update the importTemplates.txt or importSpaces.txt properties:


```
importTemplates.txt=true
importSpaces.txt=true
```
 - c. Resynchronize the custom profile.
 - 1) Open the administrative console and click **System administration** → **Nodes**.
 - 2) Click **Full Resynchronize**.
 - d. Restart the cluster.
2. If Business Space is configured in a managed server, perform the following steps:
 - a. In the custom profile where the managed server is located, open the *custom_profile_root\BusinessSpace\node_name\server_name\mm.runtime.prof\public\oobLoadedStatus.properties* file and update the importTemplates.txt or importSpaces.txt properties:


```
importTemplates.txt=true
importSpaces.txt=true
```
 - b. Resynchronize the custom profile.
 - 1) Open the administrative console and click **System administration** → **Nodes**.
 - 2) Click **Full Resynchronize**.
 - c. Restart the server.

Configuring business rules and selectors

Business rules and selectors provide flexibility in a business process by changing the results of a process based on a criteria. Before installing applications that contain business rules and selector components, you must install the business rules dynamic repository. You can install the business rules dynamic repository for a stand-alone server or for network deployment.

Configuring the business rule and selector audit log

You can configure the server to use different values than the default values for the log that keeps track of new, changed, and deleted business rules and selectors. Changing the configuration can help you conserve resources on your server.

Before you begin

You must be at the administrative console to perform this task.

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

About this task

After you have run your server in production for a while, you may have determined that the default values the server uses for the business rules and selectors audit log need adjustment.

To configure the business rule and selector audit log, perform the following steps.

Procedure

1. Navigate to the **Business Rules and Selectors Auditing** page by clicking **Servers > Application servers > *servername* Business Rules > Business Rules and Selectors Auditing**.
2. Do one of the following depending on the type of change you want.

Type of change	Actions
Immediate	<ol style="list-style-type: none">1. Select the Runtime tab.2. Enter the desired changes.3. Optional: To make the changes permanent, copy them to the repository by selecting Save runtime changes to configuration as well.4. Click OK to make the changes and return to the previous page or Apply to make the changes and remain on this page.
Delayed	<ol style="list-style-type: none">1. Select the Configuration tab.2. Enter the desired changes.3. Click OK to make the changes and return to the previous page or Apply to make the changes and remain on this page.4. When you want the changes to take effect, restart the server.

Results

The audit log takes the attributes you specified.

Note: You may need to modify the configuration for business rules and selector auditing due to the way the server user identity is specified when security is enabled with WebSphere Application Server 6.1. If the default value is used for the server user identity, an automatically generated server identity value is recorded in the audit record for the user when any auditable action involving business rules or selectors is performed when the application containing the business rules or selectors is started after business rule or selector installation. An auditable action occurs when a business rule or selector artifact is changed through application startup after install, management clients, or import or export through the administrative console. The generated value may not match the format of other user IDs used in other audit records, and you may want a more consistent value.

You can specify a server identity by selecting the option to use a "Server identity that is stored in the repository," which will associate a user ID that is in the user repository with the server process. The audit records will use this identity when auditable actions involving business rules or selectors are performed when the application containing the business rules or selectors is started after the business rule or selector artifacts are installed in the repository.

The server identity value has no effect on audit actions involving changes through management clients such as the business rules manager or other administrative actions such as exporting or importing business rule groups. For these actions, the audit record will use the authenticated user.

For more information on changing the server user identity, see the topics under Securing applications and their environment and the WebSphere Application Server WebSphere Application Server Network Deployment Security documentation.

Configuring business rule and selector auditing using commands

Use commands to configure business rule and selector auditing when you need to change any of the characteristics while a server is running.

Before you begin

You must run these commands from a command line environment for the server.

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

About this task

There may be occasions when you need to change how many servers audit business rules and selectors and cannot restart the servers involved. Using the command line, you can automate configuring the servers in a batch mode. The following tasks show how to use commands to configure one server.

Important: These settings are not saved if you restart the server. To save the configuration after entering these commands, you must use the administrative console. Select **Servers > Server Types > WebSphere application servers > *servername* > Business Rules > Business Rules and Selectors Auditing > Runtime** or **Servers > Server Types > WebSphere application servers > *servername* > Selectors > Business Rules and Selector Auditing > Runtime**.

To configure business rule and selector auditing using commands, perform the following steps.

Note: The following task configures server `server1`. If the server is not named `server1`, replace `server1` below with the name of the server. All of the steps beginning at step 3 could be placed in a `jacl` script and run that way.

Procedure

1. Enter the administrative environment.
`wsadmin`
2. Decide whether you are configuring audit logging or changing an existing configuration.

Task	Command
Configuring audit logging	<code>set mbean [\$AdminControl queryNames *:*,name=CustomizationAuditMBean,process=server1]</code>
Changing audit logging configuration	<code>set auditconfig [\$AdminConfig list AuditLog]</code>

3. Enter the appropriate commands.

Commands to configure or change audit logging

Important: When entering commands that change an existing configuration, you must save the changes. The changes do not take effect until you restart the server.

The following are the commands you can enter:

\$AdminControl invoke \$mbean getSeparateAuditLogEnabled

Use to determine whether logging is occurring to a separate audit log.

\$AdminControl invoke \$mbean setSystemOutAuditLogEnabled {boolean}

Use to enable or disable logging to the SystemOut.log file. *Boolean* can either be true or false.

\$AdminControl invoke \$mbean getSeparateAuditLogFileName

Use to determine the file name of the separate audit log.

\$AdminControl invoke \$mbean setSeparateAuditLogFileName {filename}

Use to set the name of the new log file, for example, MyAudit.log.

\$AdminControl invoke \$mbean getSeparateAuditLogFileRolloverSize

Use to determine the size of the audit log.

\$AdminControl invoke \$mbean setSeparateAuditLogFileRolloverSize integer

Use to set the size of the audit log before the system rolls it over into a history file. The size is in megabytes.

\$AdminControl invoke \$mbean

getSeparateAuditLogFileMaxNumberOfBackupFiles

Use to determine the number of audit log history files.

\$AdminControl invoke setSeparateAuditLogFileMaxNumberOfBackupFiles

integer Use to set the number of audit log history files.

\$AdminControl invoke \$mbean setSeparateAuditLogEnabled {boolean}

Use to start or stop logging to a separate log file. *Boolean* can either be true or false.

\$AdminConfig showall \$auditconfig

Use to show the current audit log configuration.

\$AdminConfig modify \$auditconfig {{separateAuditLogEnabled true}}

Use to enable logging to a separate audit log.

\$AdminConfig modify \$auditconfig {{systemOutAuditLogEnabled false}}

Use to disable auditing to the system.Out file.

\$AdminConfig modify \$auditconfig {{customAuditLog

{{maxNumberOfBackupFiles 7} {rolloverSize 7}}}}

Use to change the number of audit log history files and the size of the audit log file.

\$AdminConfig modify \$auditconfig {{customAuditLog {{fileName

MyAudit.log}}}}

Use to change the name of the audit log file.

\$AdminConfig save

Use to save the configuration.

What to do next

Save these changes by opening the administrative console and selecting **Servers > Server Types > WebSphere application servers > servername > Business Rules >**

Business Rules and Selectors Auditing > Runtime or Servers > Server Types > WebSphere application servers > *servername* > Selectors > Business Rules and Selector Auditing > Runtime. Alternatively, enter \$AdminConfig save.

Note: You may need to modify the configuration for business rules and selector auditing due to the way the server user identity is specified when security is enabled with WebSphere Application Server 6.1. If the default value is used for the server user identity, an automatically generated server identity value is recorded in the audit record for the user when any auditable action involving business rules or selectors is performed when the application containing the business rules or selectors is started after business rule or selector installation. An auditable action occurs when a business rule or selector artifact is changed through application startup after install, management clients, or import or export through the administrative console. The generated value may not match the format of other user IDs used in other audit records, and you may want a more consistent value.

You can specify a server identity by selecting the option to use a "Server identity that is stored in the repository," which will associate a user ID that is in the user repository with the server process. The audit records will use this identity when auditable actions involving business rules or selectors are performed when the application containing the business rules or selectors is started after the business rule or selector artifacts are installed in the repository.

The server identity value has no effect on audit actions involving changes through management clients such as the business rules manager or other administrative actions such as exporting or importing business rule groups. For these actions, the audit record will use the authenticated user.

For more information on changing the server user identity, see the topics under Securing applications and their environment and the WebSphere Application Server WebSphere Application Server Network Deployment Security documentation.

Considerations for installing the business rules manager

If you are planning to use the business rules manager in a distributed environment, you must understand the concepts of cells, nodes, and clusters and how to set up the business rules manager for best performance during run time.

The application server is organized on the concept of cells, nodes and servers. In a stand-alone server configuration, a cell contains one node, and each node contains one server. System administration applications and user applications all run in the same server. In a stand-alone server configuration, you can install the business rules manager in the same application server, and it can be accessed by the default URL.

In a distributed server configuration, you can configure a cell to contain multiple nodes, and each node can contain multiple application servers. Each cell constitutes a single administrative domain. With this configuration, you can use central administration, workload management, and failover configuration for the entire domain.

For best performance in a distributed server configuration, install the business rules manager on the administrative deployment target, an application server in the cell where business administration services are centrally hosted. This server is typically the same server that hosts the Common Event Infrastructure service.

Within a cell, all servers use and share a single business rules repository. When you access the business rules repository, you can access all dynamic business rule artifact definitions regardless of the exact location where the business application is installed.

Because of this central storage for all business rules in the cell at run time, you can deploy the business rules manager to any application servers in the cell, and the business rules manager gives a consistent view of all business rules within the cell. However, because of high-availability considerations, it is recommended that system administrators deploy the business rules manager into the administrative deployment target, a dedicated application server in the cell where business administration services are centrally hosted. The administrative deployment target server is the same server where the Common Event Infrastructure service and other business administrative applications are installed. With this configuration, when you require high availability, you can cluster the administrative deployment target server to provide a scalable solution to the application users.

Installing the business rules manager using the administrative console

You can install the business rules manager as an enterprise application on WebSphere Process Server to manage business rules during run time. For WebSphere Process Server 6.1 and higher, you can install the business rules manager simultaneously when creating a WebSphere Process Server profile by selecting the check box on the Business Rules Manager Configuration page of the Profile Management tool. Alternatively, you can install the business rules manager using three other methods: through the configuration page of the administrative console, by using the JACL command for your operating system, or by using the Admin Tasks command (this method is for WebSphere Process Server 6.1 and higher). For more information, see the individual topics for each installation method.

Before you begin

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

About this task

To install the business rules manager using the administrative console, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane click **Servers > Server Types > WebSphere application servers** or **Servers > Clusters > WebSphere application server clusters**.
3. Select the name of your server or cluster target.
4. On the Configuration-tabbed page, under **Business Integration**, expand **Business Rules** and click **Business Rules Manager Configuration**.
5. Under **General Properties** select the **Install business rules manager** check box.

Note: If the business rules manager has already been installed, the check box will be checked but grayed out as it is not possible to uninstall the business rules manager from this page. However, you can uninstall it manually by going to the list of applications and uninstalling it from there.

6. In the **Context root** field either accept the default context root of `/br` or type a custom context root for the business rules manager URL.
7. Click **OK**.
8. Save the configuration.

What to do next

In the navigation pane click **Applications > Application Types > WebSphere enterprise applications** and select **Start Business Rules Manager**.

Installing the business rules manager using the JACL command

You can use a JACL command as an alternative to using the administrative console for installing the business rules manager. Using a JACL command is possible if you did not already install the business rules manager when you installed WebSphere Process Server and created profiles.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, you must be authenticated with a user ID that has been assigned to the administrator or configurator role to perform this task.

About this task

To install the business rules manager using the JACL command, perform the following steps.

Procedure

1. Ensure that WebSphere Process Server is started.
2. Open the shell environment or command prompt for your operating system, and go to the `install_root/bin` directory.
3. Run the following installation command: `wsadmin.sh -f ./installBRManager.jacl [-s servername -n nodename | -cl clustername] -ce cellname -r rootname`

To install and map the business rules manager to more than one target, run the following command: `wsadmin.sh -f installBRManager.jacl -m "{{target1} {target2} ... {targetn}}" -ce cellname -r rootname`

Note: The parameter “-m” (implying “multiple”) allows you to install and map the business rules manager to many targets at the same time. A pair of double quotation marks encloses the targets.

where:

servername

The name of the application server.

The pair of arguments “-s *servername*” is required in the Network Deployment configuration if a cluster is not specified. If missing, the default value of *servername* is “server1”.

nodename

The name of the installation node.

The pair of arguments “-n *nodename*” is required in the Network Deployment configuration if a cluster is not specified.

clustername

The name of the cluster where you want to install the application.

The pair of arguments "-cl clustername" is required in the Network Deployment configuration if a server name and a node name are not specified.

Note: You must either specify the node and server or specify the cluster. Do not specify both.

cellname

The name of the installation cell.

The pair of arguments "-ce cellname" is optional.

rootname

The name of the application root directory.

The pair of arguments "-r rootname" is optional. If missing, the default value of *rootname* is "/br".

target_i The target (where *i* is 1, 2, ..., *n*) to which you want to install and map the business rules manager.

The target can be either (-s *servername* and -n *nodename*) or -cl *clustername*.

Important: If WebSphere Process Server is configured in a single-server environment, all of these pairs of arguments are optional. If WebSphere Process Server is configured for a Network Deployment environment, one of the following argument pairs is required:

- either (-s *servername* and -n *nodename*)
- or -cl *clustername*
- or -m "{{target1} {target2} ... {targetn}}"

The other argument pairs are optional.

Example

Example: Suppose that you want to map the business rules manager application to the following targets:

- cluster "BofACluster"
- Web server "RedirectorServer" and node "AIXNode01"
- application server "LinuxServer" and node "LinuxNode02"

on context root "bofa/brm"

You would run the command, as follows:

```
install_root/bin/wsadmin -f installBRManager.jacl -m "{{-cl BofACluster}
{-n AIXNode01 -s RedirectorServer} {-s LinuxServer -n LinuxNode02}}" -r
bofa/brm
```

Installing the business rules manager using the AdminTask command

With WebSphere Process Server 6.1 and higher, you can install the business rules manager using the Admin Task command. Similar to using the administrative console or the JACL command, use the Admin Task command if you did not install the business rules manager when you installed WebSphere Process Server.

Before you begin

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

About this task

To install the business rules manager using the admin task command, perform the following steps.

Procedure

1. Ensure that WebSphere Process Server is started.
2. In a command window, go to the WebSphere Process Server home directory and change to the directory `/bin`.
3. Run the `wsadmin` command to enter the `wsadmin` mode.
4. Enter one of the following commands to install the business rules manager:

```
wsadmin> $AdminTask configBusinessRulesManager {-serverName <serverName>
-nodeName <nodeName> -contextRoot <contextRoot>}
```

Or

```
wsadmin> $AdminTask configBusinessRulesManager {-clusterName
<clusterName> -contextRoot <contextRoot>}
```

where

serverName

The name of the application server.

nodeName

The name of the installation node.

clusterName

The name of the cluster where you want to install the application.

contextRoot

The context root used to launch the application. The default value is `/br`.

5. Run `wsadmin> $AdminConfig save` to save the configuration.

Tip: You can run `$AdminTask help configBusinessRulesManager` to learn more about its parameters.

Example

Example: To install the business rules manager on server "cvuServer" and node "cvuNode01" with context root "br", you should enter the following command:

```
wsadmin> $AdminTask configBusinessRulesManager {-serverName cvuServer
-nodeName cvuNode01 -contextRoot br}
```

Then to save the configuration, enter:

```
wsadmin> $AdminConfig save
```

Configuring server security for the business rules manager

If you want to use security with your server, you must configure the server that is using the business rules manager. On a server where security is not enabled, you can use the business rules manager without additional configuration.

About this task

If you have different roles or user IDs, you must set administrative security when configuring your server. To set security for your server, perform the following steps.

Procedure

1. Set administrative security on user IDs by assigning a role to each ID when creating the user IDs. Create each user ID and map each user ID to the role `BusinessRuleUser`.

To set the role, navigate to the business rule manager application (**Applications > Enterprise Applications**), select the business rule manager application, select the Security role to user/group mapping and update the `BusinessRuleUser` role.

In addition to the `BusinessRuleUser` role, two other roles are defined: `NoOne` and `AnyOne`. `NoOne` is used by developers to explicitly set the resources that should not be accessed directly. `AnyOne` is used by Tivoli Access Manager to obtain authorization for a WebSphere Process Server environment.

Note: In an ND environment with administrative security turned on, if you plan to run the business rules manager on port `908n`, where `n` is a positive integer, you should make sure that port `"944(n+3)"` with the host value of `"*"` was configured. If there is no such port, manually configure it before you launch the business rules manager.

2. Set the session tracking mechanism to use cookies to track sessions.
3. At a minimum, set an appropriate session timeout value.

Configuring a Web browser for the business rules manager

The server configures a client automatically while installing the business rules manager, but you must ensure that the Web browser is configured correctly for the business rules manager to work properly.

About this task

To ensure that the Web browser is configured correctly for the business rules manager, perform the following steps.

Procedure

1. Make sure that scripting is enabled in the Web browser.

The business rules manager requires scripting to function.

2. Make sure that cookies are enabled.

When necessary, cookies are used to track the session when you are using the business rules manager. Therefore, enable cookies on your browser when tracking sessions. Contact your system administrator if you enable cookies.

Configuring the relationship service

After installing the product, you need to set the configuration properties for the relationship service.

Before you begin

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as a configurator or an administrator to perform this task. Any WebSphere security role can view this configuration.

About this task

To set the data source and query block size (relationship instance count) properties for the relationship service, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. In the navigation pane, click **Integration Applications > Relationship Manager**.
3. Click **Relationship Services configuration**.
The configuration tabbed page displays, showing the name and version (read-only) of the currently installed relationship service.
4. In the **Query block size (relationship instance count)** field, specify the maximum cache that the relationship service should set aside for relationship queries. This setting determines the size of the query results set. By default, 5000 relationship instances are read at once. This field controls the server size memory usage and provides the administrator with a level of control over how much memory resource is consumable by any given query.
5. In the **Data source** field, specify the default data source for the relationship service by entering the Java Naming and Directory Interface (JNDI) name of a data source defined at the cell level. This is where the tables for the relationship service are stored. Each relationship-related schema is created in this data source by default.
6. You then have the following options:
 - Click **OK** to save your changes and return to the previous page.
 - Click **Reset** to clear your changes and restore the currently configured values or most recently saved values.
 - Click **Cancel** to discard any unsaved changes on the page and return to the previous page.

DB2 setup required for the relationships function

The following DB2 configuration steps are required to enable the WebSphere Process Server relationships function.

The relationships function requires customization of DB2 stored procedures. The following steps outline this procedure:

1. Ensure REXX enablement in DB2.
2. Enable DSNTPSMP.
3. Define WLM environment names to WLM.
4. Define WLM environment names to DB2.
5. Create WLM JCL.

For more information about these steps, see "Setting up the environment for the SQL procedure processor (DSNTPSMP)" in DB2 for z/OS Application Programming & SQL Guide .

Before running DSNTPSMP/WPS setup, ensure that the ID you are using has the authority to run DSNTPSMP. See "Invoking the SQL procedure processor (DSNTPSMP) in an application" outlined in DB2 for z/OS Application Programming & SQL Guide.

You can also provide overrides for DSNTPSMP by using CFGTPSMP. For details on all of the options that you can set in this file and how to set them, see the DSNTPSMP CLIST comments in DSN.SDSNCLST(DSNTPSMP).

Tip: Reference CFGTPSMP, when debugging CREATE PROCEDURE change DSNTPSMP_TRACELEVEL to MEDIUM.

Configuring extended messaging resources

Use the administrative console to configure resources needed by the extended messaging service and the applications that use the service. You can enable the extended messaging service, configure listener port extensions to handle late responses, and add or modify input and output ports for applications that use extended messaging.

Important: The Extended Messaging Service feature was deprecated in WebSphere Process Server 6.0.1 and is no longer available for application use as of WebSphere Process Server 6.2, except when managing any 6.0.x nodes that exist in a cell during migration. Replace any existing applications which depend on Extended Messaging services with ones that use standard JMS APIs, or equivalent messaging technologies.

Extended messaging enables container-managed messaging. It extends the base Java Message Service (JMS) support, the Enterprise Java Bean (EJB) component model, and support for EJB 2.0 message-driven beans to allow use of the existing container-managed persistence and transactional behavior.

Extended messaging uses the bean-managed messaging implementation to provide the JMS interfaces, which ensures that both bean-managed and extended messaging use consistent JMS support. JMS usage is simplified since its support is managed by the extended messaging service.

For a complete description of extended messaging, see the following articles in the WebSphere Business Integration Server Foundation information center:

- Extended messaging: Overview
- Using extended messaging in applications

Enabling the Extended Messaging Service

Enable the Extended Messaging Service to provide runtime support for container-managed messaging (extended messaging). Use the Extended Messaging Service page to specify whether this service starts automatically when the application server starts or whether it must be started manually.

About this task

Important: The Extended Messaging Service feature was deprecated in WebSphere Process Server 6.0.1 and is no longer available for application use as of WebSphere Process Server 6.2, except when managing any 6.0.x nodes that exist in a cell during migration. Replace any existing applications which depend on Extended Messaging services with ones that use standard JMS APIs, or equivalent messaging technologies.

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

To enable the Extended Messaging Service, perform the following steps.

Procedure

1. Ensure that the administrative console is running.
2. Click **Servers** → **Server Types** → **WebSphere application servers** → *server_name* → **Extended Messaging Service** to display the Extended Messaging Service page.
3. If you want to enable the Extended Messaging Service to start automatically with server startup, select the **Enable service at server startup** check box. If you want to start the service manually, ensure the check box is cleared.
4. Click **OK**.
5. When prompted, click **Save** on the console task bar to save your changes to the master repository.
6. If you are using the WebSphere MQSeries bindings transport in your configuration, set the value of the MQ_INSTALL_ROOT environment variable as follows:
 - a. From within the administrative console, click **Environment** → **WebSphere Variables**.
 - b. Click MQ_INSTALL_ROOT to display the configuration page for the environment variable.
 - c. In the **Value** field, delete the default value (`${WAS_INSTALL_ROOT}/lib/WMQ`) and replace it with the explicit installation path (for example, `D:/IBM/WebSphereMQ` on a Windows system).
 - d. Click **OK**.
7. Stop and restart the application server in order for the changes to take effect.

Configuring listener port extensions to handle late responses

To enable a listener port to handle late responses, configure an extension that specifies how often the port checks for responses and how long it waits for those responses.

About this task

Important: The Extended Messaging Service feature was deprecated in WebSphere Process Server 6.0.1 and is no longer available for application use as of WebSphere Process Server 6.2, except when managing any 6.0.x nodes that exist in a cell during migration. Replace any existing applications which depend on Extended Messaging services with ones that use standard JMS APIs, or equivalent messaging technologies.

Late responses occur when the messaging infrastructure delays a response to a message sent by a sender bean, thereby preventing the application from receiving that response. Extended messaging can retrieve these late response messages and pass them to a message-driven bean provided by the application to handle late responses.

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

To create and enable a listener port extension that handles late responses, perform the following steps.

Procedure

1. Ensure you have a listener port defined and configured, and that you have deployed the sender bean with the **Handle late responses** option enabled.

Note: For more information about deploying a sender bean with this option enabled, refer to the WebSphere Business Integration Server Foundation Information Center.

2. From the administrative console, click **Servers** → **Server Types** → **WebSphere application servers** → *server_name* → **Extended Messaging Service** → **Listener Port Extensions** .
3. From the Listener Port Extensions page, click **New** to create a new listener port extension.
4. From the New Listener Port Extension page, select the **Enabled** check box to enable the extension and late response handling.
5. In the **Request Interval** field, either accept the default value or specify a new value to indicate how often the listener port checks for late responses.
6. In the **Request Timeout** field, either accept the default value or specify a new value to indicate how long the listener port waits for a late response. The listener port discards any responses received after the specified timeout value.
7. Use the **Listener Ports** drop-down menu to specify the listener port to use for the extension.
8. Click **OK**.
9. When prompted, click **Save** on the console task bar to save your changes to the master repository.
10. Stop and restart the application server in order for the changes to take effect.

What to do next

After you create a listener port extension, you can modify its configuration as necessary by using the Listener Port Extensions settings page.

Selecting extended messaging providers

Select the extended messaging provider you want to administer by clicking the appropriate scope on the Extended Messaging Provider page. Each scope (cell, node, and server) that contains applications that use extended messaging has its own extended messaging provider to manage resources. You can create, modify or delete input ports, output ports, or other custom properties for each provider.

About this task

Important: The Extended Messaging Service feature was deprecated in WebSphere Process Server 6.0.1 and is no longer available for application use as of WebSphere Process Server 6.2, except when managing any 6.0.x nodes that exist in a cell during migration. Replace any existing applications which depend on Extended Messaging services with ones that use standard JMS APIs, or equivalent messaging technologies.

Required security role for this task: When security and role-based authorization are enabled, you must be logged in as administrator, operator, configurator, or monitor to perform this task.

To select the extended messaging provider you want to administer, perform the following steps.

Procedure

1. From the administrative console, click **Resources > Extended Messaging Provider**.
2. From the Extended Messaging Provider page, select the appropriate scope for the extended messaging provider you want to administer.
 - **Cell:** The most general scope. Extended messaging resources defined at the cell scope are visible from all nodes and servers, unless they have been overridden.
 - **Node:** Extended messaging resources defined at the node scope override any duplicates defined at the cell scope. They are visible to all servers on the same node, unless they have been overridden at a server scope on that node.
 - **Server:** Extended messaging resources defined at the server scope override any duplicate definitions defined at the cell or parent node scope. They are visible only to a specific server.

For detailed information about scopes, see the WebSphere Application Server for z/OS Information Center.

3. Click **Apply**.

Results

The administrative console updates the **Scope**, **Name**, and **Description** fields on the bottom of the page to reflect the values for the selected resource provider.

What to do next

You can now create, modify or delete input ports, output ports, or other custom properties for the selected extended messaging provider.

Configuring input ports

Use the administrative console to create new or modify existing input ports for each receiver bean that is constructed from a session bean. Input ports define properties for the receiving Java Message Service (JMS) destination, specify how to select and handle messages, and provide details for any required reply destinations.

About this task

Important: The Extended Messaging Service feature was deprecated in WebSphere Process Server 6.0.1 and is no longer available for application use as of WebSphere Process Server 6.2, except when managing any 6.0.x nodes that exist in a cell during migration. Replace any existing applications which depend on Extended Messaging services with ones that use standard JMS APIs, or equivalent messaging technologies.

You do not need to create input ports for receiver beans that are constructed from message-driven beans; the necessary details are associated with the deployed message-driven bean and the Message Listener Service.

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

To add or modify an input port, perform the following steps.

Procedure

1. From the administrative console, click **Resources > Extended Messaging Provider**.
2. From the Extended Messaging Provider page, select the appropriate scope for the resource provider you want to work with.
3. Click **Apply**.
4. Click **Input Ports** from the Additional Properties table.
5. From the Input Port collection page, do one of the following:
 - If you are creating a new input port, click **New**.
 - If you want to modify an existing input port, click the port name.
6. From the Input Port settings page, specify the appropriate properties for the input port.
7. Click **OK**.
8. When prompted, click **Save** on the console task bar to save your changes to the master repository.
9. Stop and restart the application server in order for the changes to take effect.

Input port settings:

When you create a new input port or modify an existing input port, you must specify certain properties. Use the information in this topic to determine whether a property is optional or required and what data type it accepts.

Important: The Extended Messaging Service feature was deprecated in WebSphere Process Server 6.0.1 and is no longer available for application use as of WebSphere Process Server 6.2, except when managing any 6.0.x nodes that exist in a cell during migration. Replace any existing applications which depend on Extended Messaging services with ones that use standard JMS APIs, or equivalent messaging technologies.

An input port has the following configuration properties:

Scope The scope at which the extended messaging provider is defined. The value represents the location of the configuration file. The administrative console automatically populates this field. You cannot edit the value.

Name The name of the input port, used for administrative purposes. This field requires a string value.

JNDI Name

The Java Naming and Directory Interface (JNDI) name for the input port. This field requires a string value.

Description

A description of the input port, used for administrative purposes. This field is optional, and it accepts a string value.

Category

A category string to use when classifying or grouping the resource. This field is optional, and it accepts a string with a maximum of 30 ASCII characters.

JMS Connection Factory JNDI Name

The JNDI name for the Java Message Service (JMS) connection factory used by the input port. This field requires a string value (for example, `jms/connFactory1`).

JMS Destination JNDI Name

The JNDI name for the JMS destination used by the input port. This field requires a string value (for example, `jms/destn1`).

JMS Acknowledgement Mode

The JMS mode that is used to acknowledge messages. This field is required for message-driven beans that use bean-managed transaction demarcation (in other words, the transaction type is set to Bean).

The following are valid values for this field:

- **Auto Acknowledge:** The session automatically acknowledges a message in either of the following cases:
 - When the session successfully returns from a call to receive a message
 - When the session calls a message listener to process the message and receives a successful response from that listener
- **Dups OK Acknowledge:** The session acknowledges only the delivery of messages. This can result in the delivery of duplicate messages if JMS fails.

The default mode is Auto Acknowledge.

Destination Type

The JMS resource type. This field requires one of the following values:

- **Queue:** The receiver bean receives messages from a queue destination.
- **Topic:** The receiver bean receives messages from a topic destination.

The default value is Queue.

Subscription Durability

Specifies whether a JMS topic subscription is durable. This field is required if the JMS destination type is a topic. The following are valid values for this field:

- **Durable:** A subscriber registers a durable subscription with a unique identity that is retained by JMS. Subsequent subscriber objects with the same identity resume the subscription in the state in which it was left by the previous subscriber. If there is no active subscriber for a durable subscription, JMS retains the subscription's messages until they are received or they expire.
- **NonDurable:** Nondurable subscriptions last for the lifetime of their subscriber. A client sees the messages published on a topic only while its subscriber is active. If the subscriber is inactive, the client misses the messages published on its topic.

The default value is NonDurable.

Reply JMS Connection Factory JNDI Name

The JNDI name of the JMS connection factory that is used for replies. This field requires a string value (for example, `jms/connFactory1`).

Reply JMS Destination JNDI Name

The JNDI name of the JMS destination that is used for replies. This field requires a string value (for example, `jms/destn1`).

Configuring output ports

Use the administrative console to create new or modify existing output ports for sender beans. Output ports specify the properties sender beans need to define the destinations for sent messages. They also specify optional properties when responses are expected. Output ports are associated with sender beans at deployment time.

About this task

Important: The Extended Messaging Service feature was deprecated in WebSphere Process Server 6.0.1 and is no longer available for application use as of WebSphere Process Server 6.2, except when managing any 6.0.x nodes that exist in a cell during migration. Replace any existing applications which depend on Extended Messaging services with ones that use standard JMS APIs, or equivalent messaging technologies.

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

To add or modify an output port, perform the following steps.

Procedure

1. From the administrative console, click **Resources > Extended Messaging Provider**.
2. From the Extended Messaging Provider page, select the appropriate scope for the extended messaging provider you want to modify.
3. Click **Apply**.
4. Click **Output Ports** from the Additional Properties table.
5. From the Output Port collection page, do one of the following:
 - If you want to add a new output port, click **New**.
 - If you want to modify an existing output port, click the port name.
6. From the Output Port settings page, specify the appropriate properties for the output port.
7. Click **OK**.
8. When prompted, click **Save** on the console task bar to save your changes to the master repository.
9. Stop and restart the application server in order for the changes to take effect.

Output port settings:

When you create a new output port or modify an existing output port, you must specify certain properties. Use the information in this topic to determine whether a property is optional or required and what data type it accepts.

Important: The Extended Messaging Service feature was deprecated in WebSphere Process Server 6.0.1 and is no longer available for application use as of WebSphere Process Server 6.2, except when managing any 6.0.x nodes that exist in a cell during migration. Replace any existing applications which depend on Extended Messaging services with ones that use standard JMS APIs, or equivalent messaging technologies.

An output port has the following configuration properties:

Scope The extended messaging provider scope; the value represents the location of the configuration file. The administrative console automatically populates this field. You cannot edit the value.

Name The name of the output port, used for administrative purposes. This field requires a string value.

JNDI Name

The Java Naming and Directory Interface (JNDI) name for the output port. This field requires a string value.

Description

A description of the output port, used for administrative purposes. This field is optional, and it accepts a string value.

Category

A category string to use when classifying or grouping the resource. This field is optional. It accepts a string value with a maximum of 30 ASCII characters.

JMS Connection Factory JNDI Name

The JNDI name for the Java Message Service (JMS) connection factory used by the output port. This field requires a string value (for example, `jms/connFactory1`).

JMS Destination JNDI Name

The JNDI name for the JMS destination used by the output port. This field requires a string value (for example, `jms/destn1`).

JMS Delivery Mode

The JMS mode used to deliver messages. You must select one of the following values for this field:

- Persistent: Messages put onto the destination are persistent.
- Nonpersistent: Messages put onto the destination are not persistent.

The default value is Persistent.

JMS Priority

The message priority for the queue destination. This field requires an integer value from 0 to 9. The default value is 4.

JMS Time To Live

The time, in milliseconds, a message remains in the queue. After the specified time elapses, the message expires.

This field requires an integer with a value from 0 to *n*:

- 0: Messages never time out.
- *n*: Messages time out after *n* milliseconds.

The default value is 0.

JMS Disable Message I.D.

Specifies whether the system generates a JMS message ID. This is a required field; select one of the following values:

- Selected: The system does not generate JMS message IDs.
- Cleared: The system generates JMS message IDs automatically.

By default, JMS message IDs are generated.

JMS Disabled Message Timestamp

Specifies whether the system generates a JMS message timestamp. This is a required field; select one of the following values:

- Selected: The system does not add message timestamps to sent messages.
- Cleared: The system automatically adds message timestamps to sent messages.

By default, the system adds message timestamps to sent messages.

Response JMS Connection Factory JNDI Name

The JNDI name of the JMS connection factory that is used for responses handled by the output port. This field requires a string value (for example, `jms/connFactory1`).

Response JMS Destination JNDI Name

The JNDI name of the JMS destination that is used for responses handled by the output port. This field requires a string value (for example, `jms/destn1`).

Setting up the messaging server environment

Before running any XMS applications, including the sample applications provided with XMS, you must set up the messaging server environment.

About this task

The steps that you need to complete to set up the messaging server environment depend on the artifacts that an application connects to, and whether you are using the Message Service Client for .NET or the Message Service Client for C/C++. The steps are described in the documentation for the type of client.

Procedure

- Setting up for Message Service Client for .NET
- Setting up for Message Service Client for C/C++

What to do next

You can use the sample applications provided with the Message Service clients to verify your installation and messaging server setup. For more information about using the sample applications, see the following topics:

- Using .NET sample XMS applications
- Using C/C++ sample XMS applications

Configuring the JNDILookup Web Service

If you are using the administered JMS objects provided by WebSphere Process Server with Message Service Clients for C/C++ and .NET, you must configure the JNDILookup Web Service that WebSphere Process Server provides to enable non-Java clients to access administered JMS objects from a non-Java environment.

Before you begin

Before starting this task, make sure that the JNDILookup Web Service application has been installed.

About this task

Administratively defined `ConnectionFactory` and `Destination` objects provide a separation between a JMS implementation and the JMS interfaces, which makes JMS client applications more portable since they are sheltered from the implementation details of a JMS provider. Administered objects enable an administrator to manage the connection settings for client applications from a central repository. For example, the specific queue that an application uses can be altered by changing the administered `Destination` object that the application obtains via JNDI.

Non-Java clients such as Message Service Clients for C/C++ and .NET can also use administered objects. However, since the administered JMS objects provided by WebSphere Process Server are serialized Java objects accessed via JNDI, non-Java clients are not able to interpret them properly without the use of the JNDILookup Web Service. This web service provides a lookup operation that allows Message Service Clients for C/C++ and .NET to request the retrieval of a JNDI object by specifying the name of the object. The properties of the administered object are returned to the application using a Map of name/value pairs.

Procedure

Define the JNDILookup Web Service URL within the Message Service Client for C/C++ or Message Service Client for .NET application. To define the web service URL within an application, set the `XMSC_IC_URL` property of the `InitialContext` object to the web service endpoint URL. This property can alternatively be specified as an argument on constructing the `InitialContext` object.

Configuring Common Event Infrastructure

You can configure Common Event Infrastructure resources, or change existing resources, using the server `AdminTask` object

About this task

Common Event Infrastructure (CEI) can be installed with a default configuration that is fully functional on a stand-alone server configuration. You would perform this task only when you create a stand-alone server. In all other cases, you use the administrative console to configure CEI — such as when you are installing it in a network deployment environment or in a cluster — to ensure that the configuration is appropriate on your system.

You can also use the `wsadmin` command to configure CEI, or you can use the command to alter an existing CEI configuration. In either case, you would change the configuration of CEI by using the server `AdminTask` object to run administrative commands.

After changing CEI configuration, you must restart the server or cluster.

Common Event Infrastructure components

Common Event Infrastructure components are installed as a set of applications, services, and resources on the server.

When you configure Common Event Infrastructure, a number of components are created and deployed on your server.

Common Event Infrastructure service

A service installed into the server, that enables applications and clients to use Common Event Infrastructure. You can view the configuration of the Common Event Infrastructure service in the administrative console, as follows:

- For a server, select **Servers > Application Servers > *server_name* > Business Integration > Common Event Infrastructure > Common Event Infrastructure Service**.
- For a cluster, select **Servers > Clusters > *cluster_name* > Business Integration > Common Event Infrastructure > Common Event Infrastructure Service**.

If the check box labeled "Enable the event infrastructure server" is selected, then the service is installed and running or it will start after you restart your server or cluster. If it is cleared, then the service is not installed or will be uninstalled after you restart your server or cluster

Event service settings

A set of properties used by the event service that enable event distribution and persistence using the data store. Typically, no configuration is necessary for this resource, but you might need to create additional event service settings if you want to set up multiple event services in the same cell. To view the event service settings, click **Service integration > Event service > Event service settings**.

Event messaging configuration

The resources that support asynchronous event transmission to the event service using the Java Messaging Service (JMS). The default messaging configuration uses the server embedded messaging. You can optionally configure an external JMS provider for event messaging.

Event database

The event database is used to persistently store events received by the event service. The Derby database is included as part of the server, but is not recommended for use in production environments. Instead, you can configure an external event database on the following products: DB2, Oracle, SQLServer, and Informix®.

Event filter plug-in

A filter plug-in is used to filter events at the source using XPath event selectors. To configure the filter properties, click **Service Integration > Common Event Infrastructure > Event Emitter Factories > Event Filter Settings**.

Emitter factory

An emitter factory is an object used by event sources to create emitters; an emitter is used to send events to the event service. The properties of an emitter factory affect the behavior of any emitter that is created using that emitter factory. To view the available emitter factories, click **Service Integration > Common Event Infrastructure > Event Emitter Factories**.

Event service transmission

An event service transmission is an object defining properties that determine how emitters access the event service synchronously using EJB calls; these properties are used by emitter factories when creating new emitters. You can view or change the available event service transmissions from the emitter factory settings.

JMS transmission

A JMS transmission is an object that defines properties that determine how emitters access the event service asynchronously using a JMS queue; these properties are used by emitter factories when creating new emitters. You can view or change the available JMS transmissions from the emitter factory settings.

Event group

An event group is a logical collection of events used to categorize events according to their content. When querying events from the event service or subscribing to event distribution, an event consumer can specify an event group to retrieve only the events in that group. Event groups can also be used to specify which events should be stored in the persistent data store. To view the available event groups in the administrative console, click **Service integration > Common Event Infrastructure > Event service > Event services > *event_service* > Event groups**.

Configuring the Common Event Infrastructure using the administrative console

Configure Common Event Infrastructure by using the server administrative console.

About this task

Open the Common Event Infrastructure Server panel of administrative console:

If you are configuring a server, select **Servers > Server Types > WebSphere application servers > *server_name* > Business Integration > Common Event Infrastructure > Common Event Infrastructure Server**.

If you are configuring a cluster, click **Servers > Clusters > WebSphere application server clusters > *cluster_name* > Business Integration > Common Event Infrastructure > Common Event Infrastructure Server**.

Procedure

1. Enable the deployment of the Common Event Infrastructure enterprise application by selecting the check box labeled **Enable the event infrastructure server**. If the server has already been configured, then you can enable or disable it by selecting or clearing the check box. If the enable check box is cleared then Common Event Infrastructure has not been configured, or has had a previous configuration disabled but the server has not been restarted. An information message shows you whether this deployment target has Common Event Infrastructure configured. If the server has already been configured, you can change the data source settings for the event database, the message store, or both.

Note: If you select the check box to enable the Common Event Infrastructure server and the server has not yet been configured, then the parameters shown is used to configure it unless you change them.

- If you are performing the configuration the first time, then the event data source tables are created on the common database. If there is already a Common Event Infrastructure server configuration, then you need to create a database.
- The messaging service is created under a unique schema under the common database.

When the server/cluster on which Common Event Infrastructure has been configured is restarted, then the new changes take effect.

2. Configure (or change the current settings for an existing configuration of) the event database by using one of the following methods to populate the fields with the appropriate settings.

- Click **Edit** for a database configuration panel with a more extensive list of options than the ones listed on the panel.
- Use the fields on the panel to enter the information:
 - a. **Database name** – the name of the database you use to store events.
 - b. **Create Tables** – select this check box if you want to create the database tables on the event database.

Note: If you are configuring Common Event Infrastructure to use a database on another server, then you are not be able to create the tables using this control. Instead, you will have to use the database scripts that will be generated after you complete the rest of this configuration. In this case, you can click **Edit** to show the data source detail panel, which tells you the location of the database creation scripts.

- c. **Username** and **Password** – for authenticating into the event database.
- d. **Server** – name of the server where the event database is located.
- e. **Provider** – choose a provider for your database from the menu.

Note: The **Schema** field is activated only if the database is created using DB2 on an iSeries® or z/OS platform. In all other cases, the schema field is disabled.

Important: If the tables exist on the target database, then the configuration can fail.

3. Select whether the Common Event Infrastructure bus is to be **Local** on the server, or **Remote** and reside on another server. If you choose remote, then select the remote location from the menu or click **New** to create a new remote bus.
4. Configure Common Event Infrastructure support for messaging.
 - Click **Edit** for a database configuration panel with a more extensive list of options than the ones listed on the panel.
 - Use the fields on the panel to enter the information:
 - a. **Database name** – enter the name of the database you use to store messages.
 - b. **Schema** – enter a name for the schema, or accept the default name given.
 - c. **Username** and **Password** – for authenticating into the messaging database.
 - d. **Server** – name of the server where the messaging database is located.
 - e. **Provider** – choose a provider for your database from the menu.
5. Click **OK** or **Apply**.
6. Restart your server or cluster.

Results

All the major parts of Common Event Infrastructure are now configured and running on your server or cluster. The configuration includes the event data store, the messaging engine, and the event application. This single panel can be used in place of many commands and steps you would otherwise use to configure Common Event Infrastructure.

What to do next

After you have restarted your server or cluster, you will be able to store service component events that are emitted from your applications. You can now change the runtime properties of the Common Event Infrastructure server by selecting the **Common Event Infrastructure Destination** panel. You can choose whether to start the Common Event Infrastructure server at startup, and specify the emitter factory JNDI name where the events are sent.

Deploying the Common Event Infrastructure application

Before you can use Common Event Infrastructure, you must first deploy the event service and associated resources in the server runtime environment.

About this task

The Common Event Infrastructure enterprise application includes the runtime components of the event service and the default messaging configuration used for asynchronous event submission.

To deploy the event service:

Procedure

From the wsadmin tool, run the **deployEventService** administrative command in batch or interactive mode. The parameters of the **deployEventService** administrative command are as follows:

nodeName

The name of the node where the event service to be deployed. This parameter is optional; if you do not specify a node name, the default is the current node. If you specify a node name, then you must also specify the server name using the **serverName** parameter. This parameter is not valid if you are deploying the event service in a cluster.

serverName

The name of the server where the event service to be deployed. This parameter is required only if you specify a node; it is not valid if you are deploying the event service in a cluster.

clusterName

The name of the cluster where the event service to be deployed. This parameter is optional and must not be specified if you are deploying at the node or server scope.

enable

Indicates whether the event service to be started automatically when the server starts. The default value is true.

Results

After the administrative command completes, the Common Event Infrastructure event service and default messaging configuration are deployed at the specified scope.

What to do next

If WebSphere security is enabled, you must also configure the JMS authentication alias and password using the `setEventServiceJmsAuthAlias` administrative command.

If you are deploying the event service in a cluster, you must also manually configure the event database.

Deploying Common Event Infrastructure in a cluster

There are several ways you can deploy Common Event Infrastructure resources in a cluster environment.

Deploying Common Event Infrastructure in an existing cluster:

You can deploy the event service application in an existing cluster.

About this task

Deploying the event service application in a cluster is essentially the same as deploying the application on a stand-alone server. However, in a cluster environment, no default event database is configured.

To deploy and configure Common Event Infrastructure in a cluster environment:

Procedure

1. Run the `deployEventService` administrative command as you would for a stand-alone server, but specifying the name of the cluster. Use the `clusterName` parameter to specify the cluster.
2. On the deployment manager system, run the database configuration administrative command. Specify the cluster name using the `clusterName` parameter. This command generates the database configuration script.
3. Copy the generated database configuration script to the database system.
4. Run the database configuration script on the database system to create the event database.
5. On the deployment manager system, run the `enableEventService` command to enable the event service. Use the `clusterName` parameter to specify the name of the cluster.

Creating a cluster by converting an existing Common Event Infrastructure server:

You can create a cluster by converting an existing stand-alone server that is already configured with Common Event Infrastructure.

Before you begin

Before you can convert the existing server, make sure that it is fully configured for Common Event Infrastructure. The configuration includes deploying the event service application and configuring the event database.

About this task

To create the cluster:

Procedure

1. Follow the typical WebSphere process for converting a stand-alone server into the first member of a new cluster. When the server is converted, the following steps take place:

- Common Event Infrastructure resources available at the scope of the server are moved to the new cluster scope.

Default database: If the existing server is configured with the default Derby database, the database resources are not moved to the cluster scope. Instead, these resources are removed. The default database configuration is not supported in a cluster. In this situation, the event service in the cluster is disabled by default.

- The deployed event service application target list is modified to remove the converted server and add the new cluster.
2. Optional: If the converted server was configured with the default Derby database, you must configure a new event database for the cluster and then enable the event service:
 - a. On the deployment manager system, run the database configuration administrative command. Specify the cluster name using the `clusterName` parameter. This command generates the database configuration script.
 - b. Copy the generated database configuration script to the database system.
 - c. Run the database configuration script on the database system to create the event database.
 - d. On the deployment manager system, run the **enableEventService** command to enable the event service. Use the `clusterName` parameter to specify the name of the cluster.

Creating a cluster by using an existing Common Event Infrastructure server as a template:

You can create a cluster by specifying an existing Common Event Infrastructure server as a template.

Before you begin

Before you can create a cluster using this method, you must have an existing server that is fully configured for Common Event Infrastructure. The configuration includes deploying the event service application and configuring the event database.

About this task

To create the cluster:

Procedure

1. Follow the typical WebSphere process for creating new cluster, using the existing Common Event Infrastructure server as a template for the first cluster member. When the first member is created, the following steps take place:

- Common Event Infrastructure resources available at the scope of the existing server are copied to the new cluster scope.

Default database: If the existing server is configured with the default Derby database, the database resources are not copied to the cluster scope. The

default database configuration is not supported in a cluster. In this situation, the event service in the cluster is disabled by default.

- The deployed event service application target list is modified to include the new cluster.
2. Optional: If the existing server was configured with the default Derby database, you must configure a new event database for the cluster and then enable the event service:
 - a. On the deployment manager system, run the database configuration administrative command. Specify the cluster name using the `clusterName` parameter. This command generates the database configuration script.
 - b. Copy the generated database configuration script to the database system.
 - c. Run the database configuration script on the database system to create the event database.
 - d. On the deployment manager system, run the **enableEventService** command to enable the event service. Use the `clusterName` parameter to specify the name of the cluster.

Configuring event messaging

You can modify the messaging configuration used for JMS transmission of events to the event service.

About this task

You will create the messaging infrastructure for Common Event Infrastructure when you use the administrative console panel to configure Common Event Infrastructure on a server. Generally, the messaging configuration will use the default messaging provider and create a single JMS queue for asynchronous transmission of events to the event service. You can, if necessary, modify this messaging configuration.

Configuring additional JMS queues

If you are using the default event messaging configuration, you can add additional JMS queues for transmission of events to the event service.

About this task

To configure an additional JMS queues using the default messaging configuration, you can set up multiple JMS queues that are routed to the service integration bus queue destination. The Common Event Infrastructure service integration bus queue destination depends upon the scope at which the event service is deployed:

Scope	Service integration bus queue destination
Server	<code>node.server.CommonEventInfrastructureQueueDestination</code>
Cluster	<code>cluster.CommonEventInfrastructureQueueDestination</code>

For more information about service integration bus configuration, refer to the documentation.

Configuring event messaging using an external JMS provider

If you do not want to use the default embedded messaging configuration for event transmission, you can configure asynchronous message transport to use an external Java Messaging Service (JMS) provider.

Before you begin

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. You must also create a listener port or activation specification.

About this task

To configure event messaging using an external JMS provider:

Procedure

From the wsadmin tool, run the **deployEventServiceMdb** administrative command in batch or interactive mode. The parameters of the **deployEventServiceMdb** command are as follows:

applicationName

The application name of the event service message-driven bean to be deployed. This parameter is required.

nodeName

The name of the node where the event service message-driven bean is to be deployed. If you specify a node name, you must also specify a server name. The node name is an optional parameter; the default value is the current node. Do not specify this parameter if you are deploying the application in a cluster.

serverName

The name of the server where the event service message-driven bean is to be deployed. This parameter is required if you are deploying the application at server scope; otherwise it is optional. Do not specify a server name if you are deploying the application in a cluster.

clusterName

The name of the cluster where the event service message-driven bean is to be deployed. Specify this parameter only if you are deploying the application in a cluster.

listenerPort

The name of the listener port used by the event service message-driven bean to publish events. The specified listener port must exist. You must specify either a listener port or an activation specification, but not both.

activationSpec

The JNDI name of the activation specification used by the event service message-driven bean to publish events. The specified activation specification must exist. You must specify either a listener port or an activation specification, but not both.

qcfJndiName

The JNDI name of the JMS queue connection factory to be used by the event service message-driven bean. This parameter is required if you specify an activation specification; otherwise it is optional. If you specify a queue connection factory and a listener port, the queue connection factory must match the one configured for the listener port.

Results

The **deployEventServiceMdb** administrative command deploys the message-driven bean for the event service, configured for the specified listener port or activation specification. It also creates an emitter factory and JMS transmission using the external JMS configuration. Applications can use either the default emitter factory (which is configured to use the default messaging configuration) or the new emitter factory (which uses the external JMS provider).

What to do next

If you want to set up more than one JMS queue to the event service, you can run this command 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.

Configuring the JMS authentication alias

If WebSphere security is enabled and you want to use asynchronous JMS messaging to submit events to the event service, you must configure the JMS authentication alias.

About this task

To configure the JMS authentication alias:

Procedure

From the wsadmin tool, run the **setEventServiceJmsAuthAlias** administrative command in batch or interactive mode. The parameters of the **setEventServiceJmsAuthAlias** command are as follows:

userName

The name of the user to be used for the JMS authentication alias. This parameter is required.

password

The password of the user to be used for the JMS authentication alias. This parameter is required.

nodeName

The name of the node where you want to update or create the JMS authentication alias. If you specify a node name, you must also specify a server name. Do not specify a node name if you are configuring the authentication alias in a cluster.

serverName

The name of the server where you want to update or create the JMS authentication alias. This parameter is required only if you specify a node; it is not valid if you are configuring the authentication alias in a cluster.

clusterName

The name of the cluster where you want to update or create the JMS authentication alias. Specify this parameter only if you are configuring the authentication alias in a cluster; if you specify a cluster name, do not specify a node or server name.

Results

The JMS authentication alias used by the event service objects is updated at the specified scope; if the authentication does not exist, it is created using the specified values.

Populating the event database

The event database is required to support persistence of events. If you ran the createDB.sh script, you have already created and populated the event database. If you did not use the createDB.sh script, run the generated DDL scripts to populate the event database.

Before you begin

Before you start, you must have already created the event database; see Creating the DB2 databases and storage groups using DBUtility.sh, SPUFI, or DSNTEP2.

About this task

Populate the event database using the DDL scripts that were generated by the administrative console.

Procedure

1. Copy, to your working directory, the following generated DDL script files from `/WebSphere/V6R1M0/DeploymentManager/profiles/default/databases/event/cluster_name/dbscripts/db2zos/ddl` (where *cluster_name* is the name of the cluster to which the network deployment cell belongs):
 - catalogSeed.ddl
 - cr_db.ddl
 - cr_db_catalog.ddl
 - cr_tbl.ddl
 - cr_tbl_catalog.ddl
 - ins_metadata.ddl
2. Assign the appropriate permissions to the copy of each file:

```
chmod 755 catalogSeed.ddl
```
3. Edit the values in the file to suit your needs. Change the database names and the storage group names to meet your naming requirements. The names that you specify in the file must match the values that you entered in the response file that provided input to the configuration script.

Note: The files are provided in ASCII format. If the tools that you use to view, edit, and run the scripts require the scripts to be in EBCDIC format, use the `iconv` command to convert the file to EBCDIC. For example:

```
iconv -t IBM-1047 -f IS08859-1 catalogSeed.ddl >
catalogSeed_EBCDIC.ddl
```

If you convert the file from ASCII format to EBCDIC but need to run the file in ASCII format, use `iconv` to convert the file back to ASCII. For example:

```
iconv -t IS08859-1 -f IBM-1047 catalogSeed_EBCDIC.ddl >
catalogSeed.ddl
```

4. Run the customized scripts using the tool of your choice. For example, DBUtility.sh or SPUFI.

Results

The event database is now populated.

Upgrading a DB2 for z/OS event database from a previous version

If you have an existing DB2 event database from Version 5.1 of Common Event Infrastructure on a z/OS system, you must upgrade it to the current version.

About this task

To upgrade a DB2 event database on a z/OS system:

Procedure

1. Make a backup copy of the existing event database.
2. Go to the *profile_root/bin* directory.
3. Run the DB2 for z/OS upgrade script:
 - `eventUpgradeDB2ZOS.sh runUpgrade=[true|false] dbUser=user`
`[dbName=name] [dbPassword=pw]`
`[scriptDir=dir] storageGroup=group`
`bufferPool4K=4kbufpool bufferPool8k=8kbufpool`
`bufferPool16K=16kbufpool`

The typical required parameters are as follows:

runUpgrade

Indicates whether you want the upgrade script to automatically run the generated DDL scripts to complete the database upgrade. This parameter is required. Specify `false` if you want to manually upgrade the database at a later time or on a different system.

z/OS systems: This parameter is ignored on a native z/OS system. Automatically running the generated DDL scripts is supported only on a client system.

dbUser

Specifies the DB2 user ID to use. This parameter is required.

dbName

Specifies the DB2 database name. The default name for the event database is `event`. This parameter is required if you specified `runUpgrade=true`.

dbPassword

Specifies the password for the specified DB2 user ID. This parameter is optional; if you do not specify a password, DB2 prompts you to type it.

scriptDir

Specifies the directory you want to contain the generated DDL scripts. This parameter is optional; if you do not specify a directory, the scripts are stored in the `.\eventDBUpgrade\db2zos` directory.

storageGroup

Specifies the name of the storage group. This parameter is required.

bufferPool4K

Specifies the name of the 4K buffer pool. This parameter is required.

bufferPool8K

Specifies the name of the 8K buffer pool. This parameter is required.

bufferPool16K

Specifies the name of the 16K buffer pool. This parameter is required.

To see a complete list of parameters and usage information, run the **eventUpgradeDB2ZOS** script with no parameters.

Results

The upgrade script generates the required DDL scripts for upgrading the event database. If you specified `runUpgrade=true` on a client system, the DDL scripts are automatically run, completing the upgrade.

What to do next

You must now manually run the generated DDL scripts using the SQL Processor Using File Input (SPUFI) facility. This step completes the database upgrade.

Configuring WebSphere Business Integration Adapters

You must perform installation and configuration procedures for the WebSphere Business Integration Adapter to work with WebSphere Process Server.

Procedure

1. Install the adapter.
 - a. Follow the procedures outlined at *Installing WebSphere Business Integration Adapters* products, which describe how to install WebSphere Business Integration Adapters.
 - b. Determine whether there are any additional required procedures that are specific to your adapter by going to the WebSphere Business Integration Adapters documentation and expanding the navigation under **Adapters**. If any additional installation tasks are listed for your adapter, perform those tasks.
2. Configure your adapter by going to the WebSphere Business Integration Adapters documentation, expanding the navigation under **Adapters**, and following the configuration instructions for your adapter. The configuration procedure generates the required artifacts.
3. Install the application EAR file by following the instructions for Deploying a mediation module.

Setting up administration of WebSphere Business Integration Adapters

You must perform several administrative functions before you can manage a WebSphere Business Integration Adapter.

Before you begin

- You must be familiar with the procedures outlined in *Installing WebSphere Business Integration Adapters* products.
- You must have installed the application EAR file to create the artifacts required for the WebSphere Business Integration Adapter before you perform this task.

About this task

In order to have administrative control over a WebSphere Business Integration Adapter, perform the following administrative functions.

Procedure

1. Create a Queue Connection Factory.

From the top level of the administrative console, follow these steps:

- a. Expand **Resources**.
- b. Expand **JMS**.
- c. Select **Queue connection factories**.
- d. Select the scope level that matches the scope level of the Administration Input/Output Queues.
- e. Click **New** to create a new JMS queue connection factory.
- f. Choose the JMS resource provider. Select **Default messaging provider**, and click **OK**.
- g. Accept all the default values with the following exceptions:
 - **Name:** QueueCF
 - **JNDI Name:** jms/QueueCF
 - **BusName:** *Your bus name*
- h. Complete the creation of your new JMS queue connection factory by clicking **OK**.

A message window appears at the top of the JMS queue connection factory panel.
- i. Apply the changes that you have made at the local configuration level to the master configuration by clicking **Save** in the message window.

2. Create a WebSphere Business Integration Adapter resource.

From the top level of the administrative console, follow these steps:

- a. Expand **Resources**.
- b. Open the WebSphere Business Integration Adapters page.

Select **WebSphere Business Integration Adapters**.
- c. Create a new WebSphere Business Integration Adapter by clicking **New**.
- d. Accept all the default values with the following exceptions:
 - **Name:** EISConnector
 - **Queue connection factory JNDI name:** jms/QueueCF
 - **Administration input queue JNDI name:** *connectorName/AdminInQueue*
 - **Administration output queue JNDI name:** *connectorName/AdminOutQueue*
- e. Complete the creation of the WebSphere Business Integration Adapter by clicking **OK**.

A message window appears at the top of the WebSphere Business Integration Adapters panel.
- f. Apply the changes that you have made at the local configuration level to the master configuration by clicking **Save** in the message window.

3. Enable the WebSphere Business Integration Adapter Service.

From the top level of the administrative console, follow these steps:

- a. Expand **Servers**.
- b. Expand **Server types**.
- c. Select **WebSphere application servers**.
- d. From the list of servers, select a server where the WebSphere Business Integration Adapter Service is to be enabled.

Click the name of the server that hosts the resources of interest.

- e. From the **Business Integration** list on the Configuration tab, select **WebSphere Business Integration Adapter Service**.
- f. Ensure that the **Enable service at server startup** check box is selected.
- g. Click **OK**.
A message window appears at the top of the WebSphere Business Integration Adapters page.
- h. Repeat steps 3d on page 226 to 3g for each server on which the WebSphere Business Integration Adapter Service is to be enabled.
- i. Apply the changes that you have made at the local configuration level to the master configuration by clicking **Save** in the message window.

Note: When you enable or disable a WebSphere Business Integration Adapter service, you must restart the server in order for the changes to take effect.

Configuring WebSphere Process Server for Service Federation Management

You can enable a WebSphere Process Server as a connectivity server that can be administered by the Service Federation Management (SFM) console provided with WebSphere Service Registry and Repository version 7.0. The SFM console can then configure SFM proxies in WebSphere Process Server.

About this task

You might have separate enterprise service buses (ESBs) in different business units. Each ESB and associated service registry constitute a separate domain of connected service applications. This can result in expensive duplication of applications between domains and also in increased development effort to implement application connectivity across domains. SFM, provided in WebSphere Service Registry and Repository version 7.0, allows you to establish bridges between separate ESBs, allowing services and applications to be shared between domains.

SFM provides:

- A federation model which provides a unifying view of federation relevant content.
- A Service Connectivity Management Protocol, which accesses the service connectivity and registry components supporting a domain.
- A console for controlling service domains.

SFM allows the console user to configure services in one domain so that they are available to service consumers in another domain; the service endpoints in one domain are available as service proxy endpoints in another domain.

Configuring the Service Connectivity Management connectivity server

The Service Federation Management (SFM) console uses the Service Connectivity Management Protocol (SCMP) to communicate with WebSphere Process Server.

About this task

WebSphere Process Server exposes the Atom based protocol as a system REST service named SCM Connectivity Server. This service is enabled by default in the

REST service provider for stand-alone servers and the deployment manager of a Network Deployment environment.

Procedure

1. Configure the REST services. The Atom documents returned by the protocol contain absolute URLs which are retained by the SFM console. The protocol, host name, and port number used in those absolute URLs are taken from the REST service configuration. It is important to consider any load balancing and network components between the SFM console server and WebSphere Process Server.
 - a. Configure the protocol, fully-qualified host name, and port number, for the stand-alone server or deployment manager REST service provider as described in the Configuring REST services in a service provider topic.
2. Provide the SFM console user with details to access the connectivity server.
 - a. The URL of the Atom service document for the connectivity server can be found on the REST services panel. The service has the type *SCM Connectivity Server*.
 - b. If WebSphere Process Server administrative security is enabled, the SFM console user will also require a username and password to access the service endpoint. These credentials must be for a user in the RestServicesUser group who has sufficient administrative rights to install Service Connectivity Architecture modules.

Configuring the Service Connectivity Management connectivity provider

You can configure all Service Connectivity Management (SCM) connectivity providers for your environment by using the administrative console.

About this task

An SCM connectivity provider is a logical partition of the ESB that is exposed via the SCM Protocol. It defines the target (server or cluster) to which proxy gateway modules will be deployed when a SCM group proxy is created on that connectivity provider. It also defines properties that will be used for proxy targets created on those group proxies.

Procedure

Select **Service integration > SCM connectivity providers**. The SCM connectivity providers page opens, displaying all connectivity providers in your environment.

Results

SCM connectivity providers can be added, removed, or worked with from this page.

Adding a connectivity provider

You can add a server or a cluster as a Service Connectivity Management (SCM) connectivity provider using the administrative console.

Procedure

1. Click **Service integration > SCM connectivity providers**. The SCM connectivity providers page opens, displaying all connectivity providers in your environment.

2. Click **Add** to add a server or a cluster as a connectivity provider. The wizard for adding connectivity providers will open.
3. Complete **Step 1. Select a server or cluster** on the wizard to identify the server or cluster to which SCM group proxies for this connectivity provider should be deployed. Click **Next**.
4. Complete **Step 2. Specify SCM connectivity provider properties** on the wizard to specify the properties:

Option	Description
Name	The name of the SCM connectivity provider. This must be unique within the cell. An exception is thrown if the name already exists. The name, description, contact, organization and location will be visible to users of the Service Federation Management console.
Description	A brief description of the SCM connectivity provider. This is optional and defaults to an empty string. The name, description, contact, organization and location will be visible to users of the Service Federation Management console.
Contact	The name of a contact person for the SCM connectivity provider. This is optional and defaults to an empty string. The name, description, contact, organization and location will be visible to users of the Service Federation Management console.
Organization	The name of the owning organization for the SCM connectivity provider. This is optional and defaults to an empty string. The name, description, contact, organization and location will be visible to users of the Service Federation Management console.
Location	The location for the SCM connectivity provider. This is optional and defaults to an empty string. The name, description, contact, organization and location will be visible to users of the Service Federation Management console.
HTTP host	The host name that will be returned for the endpoint of an insecure proxy target. This should be the host that web service clients in another domain will use to access the proxy, taking in to account web servers and other network components.
HTTP port	The port that will be returned for the endpoint of an insecure proxy target. This should be the port that web service clients in another domain will use to access the proxy, taking in to account web servers and other network components.

Option	Description
HTTPS host	The host name that will be returned for the endpoint of a secure proxy target. This should be the host that web service clients in another domain will use to access the proxy, taking in to account web servers and other network components.
HTTPS port	The port that will be returned for the endpoint of a secure proxy target. This should be the port that web service clients in another domain will use to access the proxy, taking in to account web servers and other network components.
Authentication Alias	The name of the authentication alias that will provide the basic authentication credentials used to retrieve WSDL documents via HTTP from the service registry associated with the SCM connectivity provider's domain. This parameter need not be specified if basic authentication is not required to connect to the service registry.
SSL configuration	The name of the SSL configuration used to retrieve WSDL documents via HTTP from a secure service registry associated with the SCM connectivity provider's domain. This is optional and, if not specified, the server's default SSL configuration will be used.

5. Click **Finish**. The SCM connectivity provider page will open, with the new connectivity provider listed.
6. Review the **Messages** section to ensure that the connectivity provider and its properties are complete.
7. Click **Save** to save the connectivity provider to the master configuration.

Removing a connectivity provider

You can remove a server or a cluster as a Service Connectivity Management (SCM) connectivity provider using the administrative console.

Procedure

1. Click **Service integration > SCM connectivity providers**. The SCM connectivity providers page opens, displaying all connectivity providers in your environment.
2. Select the connectivity provider. Click **Remove** to remove the server or cluster as a connectivity provider.

Working with connectivity providers

You can list, show, and modify a Service Connectivity Management (SCM) connectivity provider using the administrative console.

Procedure

1. Click **Service integration > SCM connectivity providers**. The SCM connectivity providers page opens, displaying all connectivity providers in your environment.
2. Select a connectivity provider to display its details page.

3. Fields can be modified on this page, although you cannot modify the **Name**, **Author**, **Created**, or **Updated** fields.
4. Use the **Apply**, **OK**, **Reset**, and **Cancel** buttons in order to complete any modifications.

Service Connectivity Management usage of Service Component Architecture modules

A Service Component Architecture module is installed every time the Service Federation Management console creates a group proxy. These Service Component Architecture modules can be viewed on the enterprise application view and Service Component Architecture module list on the administration console.

A versioned Service Component Architecture module is used for the group proxy. The base module name is `ScmGroupProxy` and the version number is `v1_0_0`. The cell identifier is formed from the connectivity provider name and a unique identifier for the group proxy within the cell.

The name of the service module as it appears in the module list is `ScmGroupProxy` (*ConnectivityProviderName_UniqueId*), and the service application name is of the form `ScmGroupProxy_v1_0_0_ConnectivityProviderName_UniqueIDApp`. The same unique identifier also forms part of the URL and Atom identifier used to access the group proxy via the SCM protocol.

A group proxy created on the connectivity provider *ExampleConnectivityProvider* with the generated unique identifier *xot5*, would result in a module with the name `ScmGroupProxy` (*ExampleConnectivityProvider_xot5*) being deployed as the application `ScmGroupProxy_v1_0_0_ExampleConnectivityProvider_xot5App` to the server or cluster associated with the connectivity provider.

The URL to access the Atom document representing the group proxy resource would be of the form:

```
/rest/scmp/connectivity-provider/ExampleConnectivityProvider-g0jk9fzm/mediation/group-proxy-type/group-proxy/xot5-g0jkja19
```

The Atom identifier for that document would be of the form:

```
urn:wesb-scmp:cell/localhostNode01Cell/connectivity-provider/ExampleConnectivityProvider-g0jk9fzm/mediation/group-proxy-type/group-proxy/xot5-g0jkja19
```

Note: Attributes of the SCM group proxy appear as promoted properties of the module. These can be viewed via the administration console but must not be modified.

Service Connectivity Management mapping to proxy gateways

A Service Connectivity Management (SCM) group proxy module is implemented as a proxy gateway within WebSphere Process Server

The SCM proxy targets for the group proxy appear as virtual services of the proxy gateway and can be viewed in Business Space powered by WebSphere via the Proxy gateway widget. Properties of the proxy target appear as properties of the virtual service.

Note: Virtual services associated with SCM group proxy modules must not be added, removed, or modified, via the Proxy gateway widget.

Troubleshooting configuration

You can diagnose problems when the configuration of WebSphere Process Server is unsuccessful.

WebSphere Process Server errors

If you experience a problem with one of the configuration tasks then there will be three main sources of information about the problem:

1. The error messages issued by the task
2. Error messages in the WebSphere deployment manager or application server job logs. If you are federating a node you might also find messages in the node agent job logs
3. Log files in the UNIX file system

Wherever possible, the cause and solution to each problem is also documented with the symptoms. The problems described here were experienced when starting the server after completing the installation procedure for WebSphere Process Server. In the examples of error messages, the messages have been made easier to read by changing the places where line breaks occur. Therefore, if you see these errors in your system the messages will have a slightly different layout.

Related reference

“Failure in loading T2 native library db2jct2zos”

This error can occur when using the DB2 Universal Driver connector, and WebSphere Application Server cannot load some external DB2 modules from SDSNLOAD or SDSNLOAD2.

“DataSource has a null RelationalResourceAdapter property” on page 234

The error shown in the example below is caused by a redundant datasource that is left behind after running the augment script zWPSConfig.sh. You can safely delete this datasource using the WebSphere administrative console. Be careful not to delete the JDBC provider that has a very similar name.

“SQLCODE = -471” on page 235

This error can occur when the Universal Driver has not been properly configured in a DB2 system.

“SQL code -204 and -516” on page 236

This error can be caused if the currentSchema property does not match the schema name of the tables and indexes that you created. The error messages show the JCA authentication alias that is being used.

“Repeated SIB messages about acquiring and losing locks” on page 237

This error can occur after correcting the DB2 Universal Driver configuration and restarting the server. The error messages are repeated continuously in the adjunct region.

Failure in loading T2 native library db2jct2zos

This error can occur when using the DB2 Universal Driver connector, and WebSphere Application Server cannot load some external DB2 modules from SDSNLOAD or SDSNLOAD2.

Error message: BBOO0220E:

```
error message: BBOO0220E:
[SCA.APPLICATION.mdcell.Bus:mdnodea.mdsr01a-SCA.APPLICATION.mdcell.Bus]
CWSIS0002E: The messaging engine encountered an exception while
starting.
Exception: com.ibm.ws.sib.msgstore.PersistenceException:
CWSIS1501E: The data source has produced an unexpected exception:
java.sql.SQLException: Failure in loading T2 native library
db2jcct2zos, reason: java.lang.UnsatisfiedLinkError:
/pp/db2v8/UK14852/jcc/lib/libdb2jcct2zos.so:
EDC5157I An internal error has occurred. (errno2=0x0BDF03B2)DSRA0010E:
SQL State = null, Error Code = -99,999DSRA0010E: SQL State = null,
Error Code = -99,999
com.ibm.ws.sib.utils.ras.SibMessage
com.ibm.ws.sib.utils.ras.SibMessage
```

There are a number of possible causes of a failure to load libdb2jcct2zos.so. A common error is the absence of the DB2 libraries from the STEPLIB of the WAS server processes. A failure like this can also be a symptom of a larger problem such as the DB2 Universal Driver not being fully configured in the DB2 system you are accessing.

Check that all the steps for installing the DB2 Universal Driver have been performed for your DB2 system.

The installation instructions for the DB2 Universal Driver can be found in the DB2 Information Center at [Installing the DB2 Universal JDBC Driver](#)

Related reference

“WebSphere Process Server errors” on page 232

If you experience a problem with one of the configuration tasks then there will be three main sources of information about the problem:

“Message reference for WebSphere Process Server for z/OS installation and configuration” on page 32

The message reference for WebSphere Process Server for z/OS lists the message codes that display while running the install script or when running the configuration script.

“Log files” on page 246

Various log files are created during the product installation and configuration process.

“Verification errors” on page 241

When you verify the installation you may encounter some problems, which are described in this section.

“Resources not seen in the administrative console” on page 241

When you are checking that applications you have installed exist in the system, you may not see them listed under the installed applications section. If you do not see the applications listed log out of the administrative console and log back in.

“Troubleshooting the Common Base Event Browser verification” on page 242

There are a number of reasons why you might see an error when testing the Common Base Event Browser in a network deployment configuration:

“bpeconfig.jacl: An error occurred installing TaskContainer” on page 239

A bpeconfig.jacl error normally occurs if you enter invalid input. The example below is displayed if you make a mistake entering group names and if you use the delete key instead of the backspace key to make a correction. In the example below, the input appears to be MKHTSMG, but the value that was entered contained invalid characters "MKSMG[D[D[D[CHTSMG".

DataSource has a null RelationalResourceAdapter property

The error shown in the example below is caused by a redundant datasource that is left behind after running the augment script zWPSConfig.sh. You can safely delete this datasource using the WebSphere administrative console. Be careful not to delete the JDBC provider that has a very similar name.

```
error message: BB000222I: DSRA8208I: JDBC driver type : 2
com.ibm.ws.exception.RuntimeWarning:
com.ibm.ws.runtime.component.binder.ResourceBindingException:
invalid configuration passed to resource binding logic. REASON: Invalid
Configuration!
The DataSource: DB2 Universal JDBC Driver DataSource has a null
RelationalResourceAdapter property.
```

Perform the following steps to remove the redundant datasource:

1. Log in to the WebSphere administrative console and navigate to Resources → JDBC Providers.
2. Set the scope to Server and click **Apply**.
3. Click the JDBC provider called **DB2 Universal JDBC Driver Provider**.
4. Click the link to **Datasources** on the right.
5. You should see a list of three datasources. Check the box next to **DB2 Universal JDBC Driver DataSource** and click the **Delete** button.
6. Save your configuration changes and restart the WebSphere server.

Related reference

“WebSphere Process Server errors” on page 232

If you experience a problem with one of the configuration tasks then there will be three main sources of information about the problem:

“Message reference for WebSphere Process Server for z/OS installation and configuration” on page 32

The message reference for WebSphere Process Server for z/OS lists the message codes that display while running the install script or when running the configuration script.

“Log files” on page 246

Various log files are created during the product installation and configuration process.

“Verification errors” on page 241

When you verify the installation you may encounter some problems, which are described in this section.

“Resources not seen in the administrative console” on page 241

When you are checking that applications you have installed exist in the system, you may not see them listed under the installed applications section. If you do not see the applications listed log out of the administrative console and log back in.

“Troubleshooting the Common Base Event Browser verification” on page 242

There are a number of reasons why you might see an error when testing the Common Base Event Browser in a network deployment configuration:

“bpeconfig.jacl: An error occurred installing TaskContainer” on page 239

A bpeconfig.jacl error normally occurs if you enter invalid input. The example below is displayed if you make a mistake entering group names and if you use the delete key instead of the backspace key to make a correction. In the example below, the input appears to be MKHTSMG, but the value that was entered contained invalid characters "MKSMG[D[D[D[CHTSMG".

SQLCODE = -471

This error can occur when the Universal Driver has not been properly configured in a DB2 system.

In the SYSIBM.SYSROUTINES table, the WLM_ENVIRONMENT for SYSIBM.SYSTABLES has a WLM name that does not match that being used in the stored procedure address space JCL.

The installation instructions for the DB2 Universal Driver can be found in the DB2 Information Center .

```
ExtendedMessage: BB000220E:
[CommonEventInfrastructure_Bus:mdnodea.mdsr01a-CommonEventInfrastructure_Bus]
CWSIS0002E: The messaging engine encountered an exception while
starting.
Exception: com.ibm.ws.sib.msgstore.PersistenceException:
CWSIS1501E: The data source has produced an unexpected exception:
com.ibm.db2.jcc.t2zos.y:[IBM/DB2][T2zos/2.9.32]
v.readExecuteCallInternal: nativeExecuteCall:5587:
DB2 engine SQL error, SQLCODE = -471, SQLSTATE = 55023,
error tokens = SYSIBM.SQLTABLES;00E7900C
```

Related reference

“WebSphere Process Server errors” on page 232

If you experience a problem with one of the configuration tasks then there will be three main sources of information about the problem:

“Message reference for WebSphere Process Server for z/OS installation and configuration” on page 32

The message reference for WebSphere Process Server for z/OS lists the message codes that display while running the install script or when running the configuration script.

“Log files” on page 246

Various log files are created during the product installation and configuration process.

“Verification errors” on page 241

When you verify the installation you may encounter some problems, which are described in this section.

“Resources not seen in the administrative console” on page 241

When you are checking that applications you have installed exist in the system, you may not see them listed under the installed applications section. If you do not see the applications listed log out of the administrative console and log back in.

“Troubleshooting the Common Base Event Browser verification” on page 242

There are a number of reasons why you might see an error when testing the Common Base Event Browser in a network deployment configuration:

“bpeconfig.jacl: An error occurred installing TaskContainer” on page 239

A bpeconfig.jacl error normally occurs if you enter invalid input. The example below is displayed if you make a mistake entering group names and if you use the delete key instead of the backspace key to make a correction. In the example below, the input appears to be MKHTSMG, but the value that was entered contained invalid characters "MKSMG[D[D[D[CHTSMG".

SQL code -204 and -516

This error can be caused if the currentSchema property does not match the schema name of the tables and indexes that you created. The error messages show the JCA authentication alias that is being used.

```
error message: BB000220E: SCHD0125E: Unexpected exception
while processing the acquireLease operation:
com.ibm.ws.leasemanager.LeaseException: SCHD0300E:
Error during Database operation,
localized message is
```

```
_:nativePrepareInto:1377:
DB2 engine SQL error, SQLCODE = -204, SQLSTATE = 42704,
error tokens = MDDBU.WSCH_LMGR,
Vendor Error Code is -204, ANSI-92 SQLState is 42704, cause:
[IBM/DB2][T2zos/2.9.32]T2zosPreparedStatement.readDescribeInput_
:nativeDescribeInput:2006:
DB2 engine SQL error, SQLCODE = -516, SQLSTATE = 26501,
error tokens =

..
..
com.ibm.db2.jcc.t2zos.y:
[IBM/DB2][T2zos/2.9.32]T2zosPreparedStatement.readDescribeInput:2006:
DB2 engine SQL error, SQLCODE = -516, SQLSTATE = 26501, ...
```

Related reference

“WebSphere Process Server errors” on page 232

If you experience a problem with one of the configuration tasks then there will be three main sources of information about the problem:

“Message reference for WebSphere Process Server for z/OS installation and configuration” on page 32

The message reference for WebSphere Process Server for z/OS lists the message codes that display while running the install script or when running the configuration script.

“Log files” on page 246

Various log files are created during the product installation and configuration process.

“Verification errors” on page 241

When you verify the installation you may encounter some problems, which are described in this section.

“Resources not seen in the administrative console” on page 241

When you are checking that applications you have installed exist in the system, you may not see them listed under the installed applications section. If you do not see the applications listed log out of the administrative console and log back in.

“Troubleshooting the Common Base Event Browser verification” on page 242

There are a number of reasons why you might see an error when testing the Common Base Event Browser in a network deployment configuration:

“bpeconfig.jacl: An error occurred installing TaskContainer” on page 239

A bpeconfig.jacl error normally occurs if you enter invalid input. The example below is displayed if you make a mistake entering group names and if you use the delete key instead of the backspace key to make a correction. In the example below, the input appears to be MKHTSMG, but the value that was entered contained invalid characters "MKSMG[D[D[D[CHTSMG".

Repeated SIB messages about acquiring and losing locks

This error can occur after correcting the DB2 Universal Driver configuration and restarting the server. The error messages are repeated continuously in the adjunct region.

```
ExtendedMessage: BB000222I:
[CommonEventInfrastructure_Bus:mdnodea.mdsr01a-CommonEventInfrastructure e_Bus]
CWSIS1538I: The messaging engine, ME_UUID=68E9550CE7780888,
INC_UUID=5f244052b02f04b4,
is attempting to obtain an exclusive lock on the data store.
..
..
ExtendedMessage: BB000222I:
[CommonEventInfrastructure_Bus:mdnodea.mdsr01a-CommonEventInfrastructure e_Bus]
CWSIS1546I: The messaging engine, ME_UUID=68E9550CE7780888,
INC_UUID=5f244052b02f04b4,
has lost an existing lock or failed to gain an initial lock on the database
```

These error messages indicate that there is a problem accessing the data store. Check that the fixWPSvars.jacl had created (jdbc/MEdatasource) in the datasource. Check that the datasource has an JCA authentication alias associated with it. If there is no JCA authentication alias associated with the datasource the database access defaults to the user ID of the servant region and tries to find tables called MKASRU which do not exist.

This error can occur because the -sibauth option has not been coded when running fixWPSvars.jacl. You can fix this in a number of ways:

- Associate the JDBC datasources used by the SIBs with the JCA authentication alias called WPSDBAlias.
- Create a new JCA Authentication Alias and associate that with the JDBC. Re-run fixWPSvars.jacl using the -sibauth option to specify WPSDBAuth as the JCA authentication alias, or use the WebSphere administrative console to make the change, specifying WPSDBAlias as the alias name.

The following steps explain how to create a new JCA authentication alias to be used by the SIBs to access DB2:

1. Open the WebSphere administrative console and navigate to **Security** → **Global security**.
2. Click the link to **J2C Authentication data** under **Additional Properties**.
3. Click the **New** button.
4. Enter a name for the alias, and enter the user ID and the password for the alias.
5. Click **OK**.
6. Click **Save**.

The following steps explain how to associate the Service Integration Buses with the authentication alias that you have created:

1. Open the WebSphere administrative console and navigate to **Service integration** → **Buses**.
2. Click the first bus in the list.
3. On the next panel, click **Messaging engines**.
4. Click the hyperlink to the messaging engine.
5. Click the link to **Data store** under **Additional Properties**.
6. Expand the drop-down list box in the **Authentication alias** field and select the alias you created earlier.
7. Click **OK** and then save the change to the configuration.
8. Click the link at the top of the page to navigate back to the list of buses.
9. Select the next bus in the list and repeat the same procedure. Repeat for the remaining buses.
10. When all the buses have been updated to refer to a valid JCA authentication alias and stop and restart the server.

Related reference

“WebSphere Process Server errors” on page 232

If you experience a problem with one of the configuration tasks then there will be three main sources of information about the problem:

“Message reference for WebSphere Process Server for z/OS installation and configuration” on page 32

The message reference for WebSphere Process Server for z/OS lists the message codes that display while running the install script or when running the configuration script.

“Log files” on page 246

Various log files are created during the product installation and configuration process.

“Verification errors” on page 241

When you verify the installation you may encounter some problems, which are described in this section.

“Resources not seen in the administrative console” on page 241

When you are checking that applications you have installed exist in the system, you may not see them listed under the installed applications section. If you do not see the applications listed log out of the administrative console and log back in.

“Troubleshooting the Common Base Event Browser verification” on page 242

There are a number of reasons why you might see an error when testing the Common Base Event Browser in a network deployment configuration:

“bpeconfig.jacl: An error occurred installing TaskContainer”

A bpeconfig.jacl error normally occurs if you enter invalid input. The example below is displayed if you make a mistake entering group names and if you use the delete key instead of the backspace key to make a correction. In the example below, the input appears to be MKHTSMG, but the value that was entered contained invalid characters "MKSMG[D[D[D[CHTSMG".

bpeconfig.jacl: An error occurred installing TaskContainer

A bpeconfig.jacl error normally occurs if you enter invalid input. The example below is displayed if you make a mistake entering group names and if you use the delete key instead of the backspace key to make a correction. In the example below, the input appears to be MKHTSMG, but the value that was entered contained invalid characters "MKSMG[D[D[D[CHTSMG".

```
..
..
[] Install the task container [Yes/no]? Yes
[adminHTMUsers] User(s) to add to role TaskSystemAdministrator
(separator is pipe,
'|') []: MKADMIN
[adminHTMGroups] Group(s) to add to role TaskSystemAdministrator
(separator is pipe,
'|') []: MKSMADMG|MKCFG
[monitorHTMUsers] User(s) to add to role TaskSystemMonitor
(separator is pipe, '|')
[]:
[monitorHTMGroups] Group(s) to add to role TaskSystemMonitor
(separator is pipe, '|')
[]: MKSMG[D[D[D[CHTSMG
[jmsHTMRunAsUser] Run-as UserId for role EscalationUser
[MKADMIN]: MKHTSM
[jmsHTMRunAsPwd] MKHTSM's password []: *****
[auto:mqType] Use WebSphere default messaging or
WebSphere MQ? WPM
task.ear install options: -appname "TaskContainer_mkcl01"
-usedefaultbindings
-defaultbinding.ejbjndi.prefix ejb/htm -cluster
```

```

"mkc101"
-BindJndiForEJBMessageBinding {{"TaskContainer" "HTMScheduler"
"taskejb.jar,META-INF/ejb-jar.xml" ""
"eis/HTMInternalActivationSpec"
"jms/HTMIntQueue" ""}} -MapResRefToEJB {
{"TaskContainer"
"GenericHumanTaskManagerEJB" "taskejb.jar,META-INF/
ejb-jar.xml" "jdbc/BPEDB"
"javax.sql.DataSource" "jdbc/BPEDB_mkc101"} {"TaskContainer"
"TaskContainerStartupBean" "taskejb.jar,META-INF/ejb-jar.xml"
"jdbc/BPEDB"
"javax.sql.DataSource" "jdbc/BPEDB_mkc101"}} -MapResEnvRefToRes
{{"TaskContainer"
"TaskContainerStartupBean" "taskejb.jar,META-INF/ejb-jar.xml"
"jms/HTMHoldQueue"
"javax.jms.Queue" "jms/HTMHoldQueue"} {"TaskContainer"
"TaskContainerStartupBean"
"taskejb.jar,META-INF/ejb-jar.xml" "scheduler/BPCScheduler"
"com.ibm.websphere.scheduler.Scheduler" "BPEScheduler"}}
} -MapRolesToUsers
{{"TaskSystemAdministrator" "AppDeploymentOption.No"
"AppDeploymentOption.No"
"MKADMIN" "MKSMADMG|MKCFG"} {"TaskSystemMonitor"
"AppDeploymentOption.No"
"AppDeploymentOption.No" "" "MKSMG[D[D[D[D[CHTSMG]"
{"EscalationUser"
"AppDeploymentOption.No" "AppDeploymentOption.Yes"
"" ""}} -MapRunAsRolesToUsers
{{"EscalationUser" "*****" "MKHTSM"}}}
An error occurred installing TaskContainer_mkc101:
..
..
com.ibm.ws.scripting.ScriptingException: WASX7132E: Application install for
/wasmkconfig/mkcell/mkdmnode/DeploymentManager/installableApps/task.ear failed: see
previous messages for details. Discarding changes.

```

If you look in the log of the servant job in the Deployment Manager you could also see a related error like that shown below:

```

error message: FFDC0010I: FFDC closed incident stream file
/wasmkconfig/mkcell/mkdmnode/DeploymentManager/profiles/default/logs/ff
dc/mkcell_mkdmnode_dmgr_STC12532_MKDMGRS_06.11.13_04.05.37_1.txt
com.ibm.etools.j2ee.commonarchivecore.exception.ResourceLoadException: IWAE0007E
Could not load resource "META-INF/ibm-application-bnd.xmi" in archive
"/wasmkconfig/mkcell/mkdmnode/DeploymentManager/profiles/default/temp/app35301.ear"
!Stack_trace_of_nested_exce!
com.ibm.etools.j2ee.exception.WrappedRuntimeException:
Exception occurred loading META-INF/ibm-application-bnd.xmi
!Stack_trace_of_nested_exce!
Wrapped exception
org.xml.sax.SAXParseException: An invalid XML character (Unicode: 0x1b) was found in
the value of attribute "name" and element is "groups".

```

Related reference

“Failure in loading T2 native library db2jcc2zos” on page 232

This error can occur when using the DB2 Universal Driver connector, and WebSphere Application Server cannot load some external DB2 modules from SDSNLOAD or SDSNLOAD2.

“DataSource has a null RelationalResourceAdapter property” on page 234

The error shown in the example below is caused by a redundant datasource that is left behind after running the augment script zWPSConfig.sh. You can safely delete this datasource using the WebSphere administrative console. Be careful not to delete the JDBC provider that has a very similar name.

“SQLCODE = -471” on page 235

This error can occur when the Universal Driver has not been properly configured in a DB2 system.

“SQL code -204 and -516” on page 236

This error can be caused if the currentSchema property does not match the schema name of the tables and indexes that you created. The error messages show the JCA authentication alias that is being used.

“Repeated SIB messages about acquiring and losing locks” on page 237

This error can occur after correcting the DB2 Universal Driver configuration and restarting the server. The error messages are repeated continuously in the adjunct region.

Verification errors

When you verify the installation you may encounter some problems, which are described in this section.

Related reference

“Failure in loading T2 native library db2jcc2zos” on page 232

This error can occur when using the DB2 Universal Driver connector, and WebSphere Application Server cannot load some external DB2 modules from SDSNLOAD or SDSNLOAD2.

“DataSource has a null RelationalResourceAdapter property” on page 234

The error shown in the example below is caused by a redundant datasource that is left behind after running the augment script zWPSConfig.sh. You can safely delete this datasource using the WebSphere administrative console. Be careful not to delete the JDBC provider that has a very similar name.

“SQLCODE = -471” on page 235

This error can occur when the Universal Driver has not been properly configured in a DB2 system.

“SQL code -204 and -516” on page 236

This error can be caused if the currentSchema property does not match the schema name of the tables and indexes that you created. The error messages show the JCA authentication alias that is being used.

“Repeated SIB messages about acquiring and losing locks” on page 237

This error can occur after correcting the DB2 Universal Driver configuration and restarting the server. The error messages are repeated continuously in the adjunct region.

Resources not seen in the administrative console

When you are checking that applications you have installed exist in the system, you may not see them listed under the installed applications section. If you do not see the applications listed log out of the administrative console and log back in.

If you do not see Service Integration Buses that you have configured, log out of the administrative console and log back in.

Related reference

“Failure in loading T2 native library db2jcc2zos” on page 232

This error can occur when using the DB2 Universal Driver connector, and WebSphere Application Server cannot load some external DB2 modules from SDSNLOAD or SDSNLOAD2.

“DataSource has a null RelationalResourceAdapter property” on page 234

The error shown in the example below is caused by a redundant datasource that is left behind after running the augment script zWPSConfig.sh. You can safely delete this datasource using the WebSphere administrative console. Be careful not to delete the JDBC provider that has a very similar name.

“SQLCODE = -471” on page 235

This error can occur when the Universal Driver has not been properly configured in a DB2 system.

“SQL code -204 and -516” on page 236

This error can be caused if the currentSchema property does not match the schema name of the tables and indexes that you created. The error messages show the JCA authentication alias that is being used.

“Repeated SIB messages about acquiring and losing locks” on page 237

This error can occur after correcting the DB2 Universal Driver configuration and restarting the server. The error messages are repeated continuously in the adjunct region.

Troubleshooting the Common Base Event Browser verification

There are a number of reasons why you might see an error when testing the Common Base Event Browser in a network deployment configuration:

CWLCB0020E: Common Event Infrastructure is unavailable with NameNotFoundException

- A fully qualified JNDI name was not used on the Common Base Event Browser.
- The schema name for the CEI tables is not equal to the user ID in the JCA Authentication Alias that is being used by the event and eventcat datasources.

The following is an example of messages that can be displayed when verifying the Common Base Event browser in the network deployment configuration for the first time.

The following is an example of messages that can be displayed in the servant log of the WebSphere Application Server:

```
javax.naming.NameNotFoundException:  
Context: mkcell/nodes/mkdmnode/servers/dmgr,  
name: ejb/com/ibm/events/access/EventAccess:  
First component in name com/ibm/events/access/EventAccess not found.  
Root exception is org.omg.CosNaming.NamingContextPackage.NotFound:  
IDL:omg.org/CosNaming/NamingContext/NotFound:1.0
```

A fully qualified JNDI name is needed in the Event Data Store when CEI is configured in a network deployment configuration. To resolve this problem, specify a fully qualified JNDI name for the EventAccess EJB.

CWLCB0020E: Common Event Infrastructure is unavailable with CEIDS0035E

The following error messages can also be displayed when verifying the Common Base Event browser in the network deployment configuration for the first time:

```
error message: CEIDS0035E The implementation class that supports the
configured relational database system cannot be loaded.
```

```
Implementation class name:
```

```
com.ibm.events.datastore.impl.Db2UniversalDriverImpl
```

```
Relational database name: DB2
```

```
Database version: DSN08015
```

```
com.ibm.events.datastore.impl.DatabaseSpecificsFactory
handleCreateException(String, String, String, Exception)
```

These error messages indicate a problem loading a class, but the cause is usually a problem accessing the database. Sometimes there are also some error messages with DB2 return codes like -204 that can help you diagnose the problem, but sometimes you only see the CEIDS0035E message.

Solution 1

If you find a -551 error in the FFDC file, correct the authorization failure reported there by issuing appropriate GRANT statements in DB2.

For example, you could issue GRANT ALL ON TABLE MKCELL. TO MKDBU for all of the CEI tables, views, and indexes created in the CEI databases, then restart the WebSphere Application Server. You may still however receive error message CWLCB0020E on the Common Base Event Browser and CEIDCS0035E in the servant log.

The following is an example of a different DB2 error reported in the FFDC log, and the same error reported in the servant message log:

```
Exception = com.ibm.db2.jcc.t2zos.y
```

```
Source = com.ibm.ws.rsadapter.jdbc.WSJdbcConnection.prepareStatement
```

```
probeid = 1584
```

```
Stack Dump = com.ibm.db2.jcc.t2zos.y:
```

```
[IBM/DB2][T2zos/2.9.32]T2zosPreparedStatement.readPrepareDescribeOutput
```

```
_:nativePrepareInto:1377:DB2 engine SQL error, SQLCODE = -204, SQLSTATE
```

```
= 42704, error tokens = MKDBU.CEI_T_Common Base Event_MAP
```

A -204 code is resource allocation failure, that is, MKDBU.CEI_T_Common Base Event_MAP not found. The CEI tables were created with a schema name of MKCELL, but the failure shows that the Common Base Event Browser is attempting to access MKDBU.CEI_T_Common Base Event_MAP. The user ID MKDBU is the one in the JCA Authentication Alias being used by the event and eventcat datasources.

The problem is that the CEI component interrogates the alias associated with the datasource and uses that to issue fully qualified SQL. This means that any value you set in the currentSchema custom property on the datasources is ignored. Unfortunately, at this time CEI must be configured so that the schema of all CEI objects in DB2 is equal to the user ID in the JCA Authentication Alias used by the event and eventcat datasources.

There are two ways to resolve this problem:

- Make the schema in the database match the user ID in the JCA Authentication Alias user ID being used on the event and eventcat datasources (solution 2a).

- Use a new JCA Authentication Alias for the event and eventcat datasources that has a user ID that matches the schema of the existing CEI tables (solution 2b). You can use this approach if longer term you do not want to have the schema of CEI tables equal to the JCA Authentication Alias user ID.

Solution 2a

Drop the CEI databases, then re-create them. Then rerun the CEI DDL (which includes inserting the metadata and seeding the catalog), but specify a schema name equal to the JCA Authentication Alias user ID on all CREATE statements.

Solution 2b

Perform the following steps if you do not want to drop and re-create the CEI tables:

1. Create a RACF user ID equal to the schema name you are currently using.
2. Using the WebSphere administrative console, define a new JCA Authentication Alias and set the RACF user ID with a password in that alias. Set the schema of the CEI user ID and password in the JCA alias that you create.
3. Navigate to **Resources** → **JDDC Providers** and set the scope depending on whether CEI was deployed in a cluster or a server:
 - If the CEI EventService application is deployed in a cluster, set the scope to the cluster.
 - If the CEI Event Server application is deployed in a server, set the scope to the server.

Click **Event_DB2ZOS_JDBC_Provider**.

4. Click **Data sources** under **Additional properties**.
5. Click **Event**.
6. Scroll down and select the new JCA Authentication Alias you created from the drop-down list box in field Component-managed authentication alias.
7. Click **OK**.
8. Navigate to the eventcat datasource and make the same change.
9. Save your configuration changes then restart the server or cluster.

The advantage of using solution 2b rather than 2a is that the reason you experienced this problem in the first place is that you do not want to have tables in DB2 with a schema that matches the JCA Authentication Alias user ID. The CEI tables already have the schema you want to use, so it does not make sense to change the schema. When a fix is available to CEI that allows you to use the `currentSchema` property on the datasource, you can easily switch the datasources so that they return to using the original JCA Authentication Alias.

Of course, even with solution 2b you will be temporarily using a JCA Authentication Alias equal to the CEI schema. When there is a fix for CEI and you switch back to the original JCA Authentication Alias (the one that is not the same as the schema), you can delete the alias and the RACF user ID for the JCA Authentication Alias you created to temporarily match the CEI schema.

CWLCB0020E: Common Event Infrastructure is unavailable with CORBA NO MEMORY

In the servant log you may also see the following error:

```
java.rmi.RemoteException: CORBA NO_MEMORY exception
```

You are unlikely to experience this problem if you have just installed WebSphere Process Server, but it is useful to know that you can experience memory problems if you get events with a high number specified in the Maximum number of events field.

You can resolve this problem in either of two ways:

- Reduce the maximum number of events to retrieve from 500 to 100, then retrieve the next blocks of events using the time and date query.
- Increase the minimum and maximum heap sizes for the JVM of the servant region in the server running the EventService application.

Note: You might meet an out-of-memory exception if you have a very large CEI event, even though the number of events is not high. To resolve this problem, increase the initial heap and the maximum heap size.

Related reference

“Failure in loading T2 native library db2jct2zos” on page 232

This error can occur when using the DB2 Universal Driver connector, and WebSphere Application Server cannot load some external DB2 modules from SDSNLOAD or SDSNLOAD2.

“DataSource has a null RelationalResourceAdapter property” on page 234

The error shown in the example below is caused by a redundant datasource that is left behind after running the augment script zWPSConfig.sh. You can safely delete this datasource using the WebSphere administrative console. Be careful not to delete the JDBC provider that has a very similar name.

“SQLCODE = -471” on page 235

This error can occur when the Universal Driver has not been properly configured in a DB2 system.

“SQL code -204 and -516” on page 236

This error can be caused if the currentSchema property does not match the schema name of the tables and indexes that you created. The error messages show the JCA authentication alias that is being used.

“Repeated SIB messages about acquiring and losing locks” on page 237

This error can occur after correcting the DB2 Universal Driver configuration and restarting the server. The error messages are repeated continuously in the adjunct region.

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 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

For a listing of the WebSphere Process Server for z/OS installation error messages, see CWPIZ in the Messages portion of the Reference documentation.

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/V7R0/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/V7R0/AppServer/logs/wbi/zWESBConfig.log and /WebSphere/V7R0/AppServer/logs/wbi/zWPSConfig.log respectively.

Related reference

“Failure in loading T2 native library db2jcc2zos” on page 232

This error can occur when using the DB2 Universal Driver connector, and WebSphere Application Server cannot load some external DB2 modules from SDSNLOAD or SDSNLOAD2.

“DataSource has a null RelationalResourceAdapter property” on page 234

The error shown in the example below is caused by a redundant datasource that is left behind after running the augment script zWPSConfig.sh. You can safely delete this datasource using the WebSphere administrative console. Be careful not to delete the JDBC provider that has a very similar name.

“SQLCODE = -471” on page 235

This error can occur when the Universal Driver has not been properly configured in a DB2 system.

“SQL code -204 and -516” on page 236

This error can be caused if the currentSchema property does not match the schema name of the tables and indexes that you created. The error messages show the JCA authentication alias that is being used.

“Repeated SIB messages about acquiring and losing locks” on page 237

This error can occur after correcting the DB2 Universal Driver configuration and restarting the server. The error messages are repeated continuously in the adjunct region.

Log files

Various log files are created during the product installation and configuration process.

Purpose

Consult the applicable logs if problems occur during the product installation 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 installation script. However, you can *redirect* these messages to a file by using the redirect symbol and a file name at the end of the command line. For

example, specifying `>run.log` at the end of the installation 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 before the Log and Trace File being opened. For instance, the following message block displays if a required keyword (`-runtime`) was not included in the installation command.

```
parsing command arguments...
CWPIZ0101E -runtime keyword and value not specified on command line.
com.ibm.ws390.installer.InstallFailureException: -runtime 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/V7R0/AppServer/profiles/default
response property: nodeName=SY1
response property: scaSecurityPassword=ibmuser
response property: dbType=Derby
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/V7R0/AppServer/derby/databases/WBIDB
response property: mqPwd=ibmuser
response property: was.install.root=/WebSphere/V7R0/AppServer
response property: augment=
response property: ceiDbProduct=DERBY_V51_1
response property: wbi.install.root=/WebSphere/V7R0/AppServer
response property: ceiSampleServerName=server1
response property: templatePath=/WebSphere/V7R0/AppServer/profileTemplates/default.*
response property: dbName=WBIDB
set up configuration complete
creating the symbolic links...
Source=/usr/lpp/zWPS/V7R0

Target=/WebSphere/V7R0/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 `zWPSInstall.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 installation 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 `zWPSInstall.sh` run.

```
[8/16/05 17:00:45:380 EDT] 0000000a ManagerAdmin I BB0002221:
TRAS0017I: The startup trace state is *=info.
```

```

[8/16/05 17:00:48:230 EDT] 0000000a WPSInstaller I BB000222I:
CWPIZ0044I Begin install task.

[8/16/05 17:00:48:273 EDT] 0000000a WPSInstaller I BB000222I:
CWPIZ0117I WPS installer log data will be written to
/WebSphere/V7R0/AppServer/logs/wbi/zWPSInstall.log.

[8/16/05 17:00:48:282 EDT] 0000000a WPSInstaller I BB000222I:
CWPIZ0024I WPS installer trace data will be written to
/WebSphere/V7R0/AppServer/logs/wbi/zWPSInstall.trace.

[8/16/05 17:00:48:292 EDT] 0000000a WPSInstaller I BB000222I:
CWPIZ0014I Trace specification is "*all=disabled".

[8/16/05 17:00:48:298 EDT] 0000000a WPSInstaller I BB000222I:
CWPIZ0045I WPS SMP/E root directory is /zrockuser/wbi/Install.

[8/16/05 17:00:48:302 EDT] 0000000a WPSInstaller I BB000222I:
CWPIZ0052I WAS SMP/E root directory is /web/usr/lpp/zWebSphere/V7R0.

[8/16/05 17:00:48:307 EDT] 0000000a WPSInstaller I BB000222I:
CWPIZ0046I Destination application server root directory is
/WebSphere/V7R0/AppServer.

[8/16/05 17:00:48:314 EDT] 0000000a WPSInstaller E BB000220E:
CWPIZ0322E profileName property not specified in Response File.

[8/16/05 17:00:48:318 EDT] 0000000a WPSInstaller E BB000220E:
CWPIZ0017E install task failed.

```

A trace file from a zWPSInstall.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/V7R0/AppServer
user.install.root = /WebSphere/V7R0/AppServer/profiles/default
Java Home = /web/usr/lpp/zWebSphere/V7R0/java/J1.4
ws.ext.dirs = /WebSphere/V7R0/AppServer/java/lib:/WebSphere/V7R0/AppServer/java/lib/
ext:/WebSphere/V7R0/AppServer/classes:/WebSphere/V7R0/AppServer/lib:/WebSphere/V7R0/AppServer/
installedChannels:/WebSphere/V7R0/AppServer/lib/ext:/WebSphere/V7R0/AppServer/deploytool/ftp
/plugins/com.ibm.etools.ejbdpdeploy/runtime:/WebSphere/V7R0/AppServer/MQSeries/pubsubroot/lib
Classpath = /zrockuser/bbzconfig.jar:/WebSphere/V7R0/AppServer/lib/admin.jar:/WebSphere/V7R0
/AppServer/lib/ant.jar:/WebSphere/V7R0/AppServer/lib/bootstrapws390.jar:/WebSphere/V7R0
/AppServer/lib/bootstrap.jar:/WebSphere/V7R0/AppServer/lib/configmanager.jar:/WebSphere
/V7R0/AppServer/lib/emf.jar:/WebSphere/V7R0/AppServer/lib/ras.jar:/WebSphere/V7R0
/AppServer/lib/runtimefw.jar:/WebSphere/V7R0/AppServer/lib/utills.jar:/WebSphere/V7R0
/AppServer/lib/wasjmx.jar:/WebSphere/V7R0/AppServer/lib/wasproduct.jar:/WebSphere/V7R0
/AppServer/lib/wccm_base.jar:/WebSphere/V7R0/AppServer/lib/wjmxapp.jar:/WebSphere/V7R0
/AppServer/lib/wsanitasks.jar:/WebSphere/V7R0/AppServer/lib/wsexception.jar:/WebSphere
/V7R0/AppServer/lib/wsprofile.jar:/WebSphere/V7R0/AppServer/profiles/default/properties:
/WebSphere/V7R0/AppServer/properties:/WebSphere/V7R0/AppServer/lib/bootstrap.jar:/WebSphere
/V7R0/AppServer/lib/j2ee.jar:/WebSphere/V7R0/AppServer/lib/lmproxy.jar:/WebSphere/V7R0
/AppServer/lib/urllibprotocols.jar:/WebSphere/V7R0/AppServer/lib/bootstrapws390.jar
Java Library path = /web/usr/lpp/zWebSphere/V7R0/java/J1.4/bin/classic/libjvm.so:/web/usr
/lpp/zWebSphere/V7R0/java/J1.4/bin/classic:/web/usr/lpp/zWebSphere/V7R0/java/J1.4/bin/:
/WebSphere/V7R0/AppServer/lib:/WebSphere/V7R0/AppServer/lib:/WebSphere/V7R0/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/V7R0/AppServer/lib:/usr/lib
Current trace specification = *all
***** End Display Current Environment *****

```

```

[10/3/05 16:35:05:709 EDT] 0000000a ManagerAdmin I BB000222I: TRAS0017I:
The startup trace state is *all.
[10/3/05 16:35:08:638 EDT] 0000000a WPSInstaller > setup Entry
/web/usr/wbi/zWebSphere/V7R0
APPSEVER
zWPSInstall.sh
-smprot
/web/usr/wbi/zWPS/V7R0
-runtime
/WebSphere/V7R0/AppServer
-response
/web/usr/wbi/zWPS/V7R0/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 BB000222I: CWPIZ0044I:
Begin install task.
[10/3/05 16:35:08:702 EDT] 0000000a WPSInstaller I BB000222I: CWPIZ0117I:
WPS installer log data will be written to /WebSphere/V7R0/AppServer/logs/wbi/zWPSInstall.log.
[10/3/05 16:35:08:712 EDT] 0000000a WPSInstaller I BB000222I: CWPIZ0024I:
WPS installer trace data will be written to /WebSphere/V7R0/AppServer/logs/wbi/zWPSInstall.trace.

```

```

[10/3/05 16:35:08:722 EDT] 0000000a WPSInstaller I BB000222I: CWPIZ0014I:
Trace specification is "*all=enabled".
[10/3/05 16:35:08:726 EDT] 0000000a WPSInstaller I BB000222I: CWPIZ0052I:
WAS SMP/E root directory is /web/usr/lpp/zWebSphere/V7R0.
[10/3/05 16:35:08:730 EDT] 0000000a WPSInstaller > checkPathName Entry
/web/usr/wbi/zWPS/V7R0
[10/3/05 16:35:08:731 EDT] 0000000a WPSInstaller < checkPathName Exit
[10/3/05 16:35:08:732 EDT] 0000000a WPSInstaller I BB000222I: CWPIZ0045I:
WPS SMP/E root directory is /web/usr/wbi/zWPS/V7R0.
[10/3/05 16:35:08:736 EDT] 0000000a Symlink > isSymlink Entry
/web/usr/wbi/zWPS/V7R0
[10/3/05 16:35:08:737 EDT] 0000000a Symlink 3 absolute path
/web/usr/wbi/zWPS/V7R0
[10/3/05 16:35:08:737 EDT] 0000000a Symlink 3 canonical path
/web/usr/wbi/zWPS/V7R0
[10/3/05 16:35:08:738 EDT] 0000000a Symlink < isSymlink Exit
false
[10/3/05 16:35:08:738 EDT] 0000000a WPSInstaller I BB000222I: CWPIZ0046I:
Destination application server root directory is /WebSphere/V7R0/AppServer.
[10/3/05 16:35:08:744 EDT] 0000000a WPSInstaller I BB000222I: CWPIZ0247I:
Response file is /web/usr/wbi/zWPS/V7R0/zos.config/sample.rsp.
[10/3/05 16:35:08:764 EDT] 0000000a WPSInstaller 3 response property
profilePath=/WebSphere/V7R0/AppServer/profiles/default
[10/3/05 16:35:08:765 EDT] 0000000a WPSInstaller 3 response property
nodeName=SY1

```

Related reference

“Failure in loading T2 native library db2jct2zos” on page 232

This error can occur when using the DB2 Universal Driver connector, and WebSphere Application Server cannot load some external DB2 modules from SDSNLOAD or SDSNLOAD2.

“DataSource has a null RelationalResourceAdapter property” on page 234

The error shown in the example below is caused by a redundant datasource that is left behind after running the augment script zWPSConfig.sh. You can safely delete this datasource using the WebSphere administrative console. Be careful not to delete the JDBC provider that has a very similar name.

“SQLCODE = -471” on page 235

This error can occur when the Universal Driver has not been properly configured in a DB2 system.

“SQL code -204 and -516” on page 236

This error can be caused if the currentSchema property does not match the schema name of the tables and indexes that you created. The error messages show the JCA authentication alias that is being used.

“Repeated SIB messages about acquiring and losing locks” on page 237

This error can occur after correcting the DB2 Universal Driver configuration and restarting the server. The error messages are repeated continuously in the adjunct region.



Printed in USA