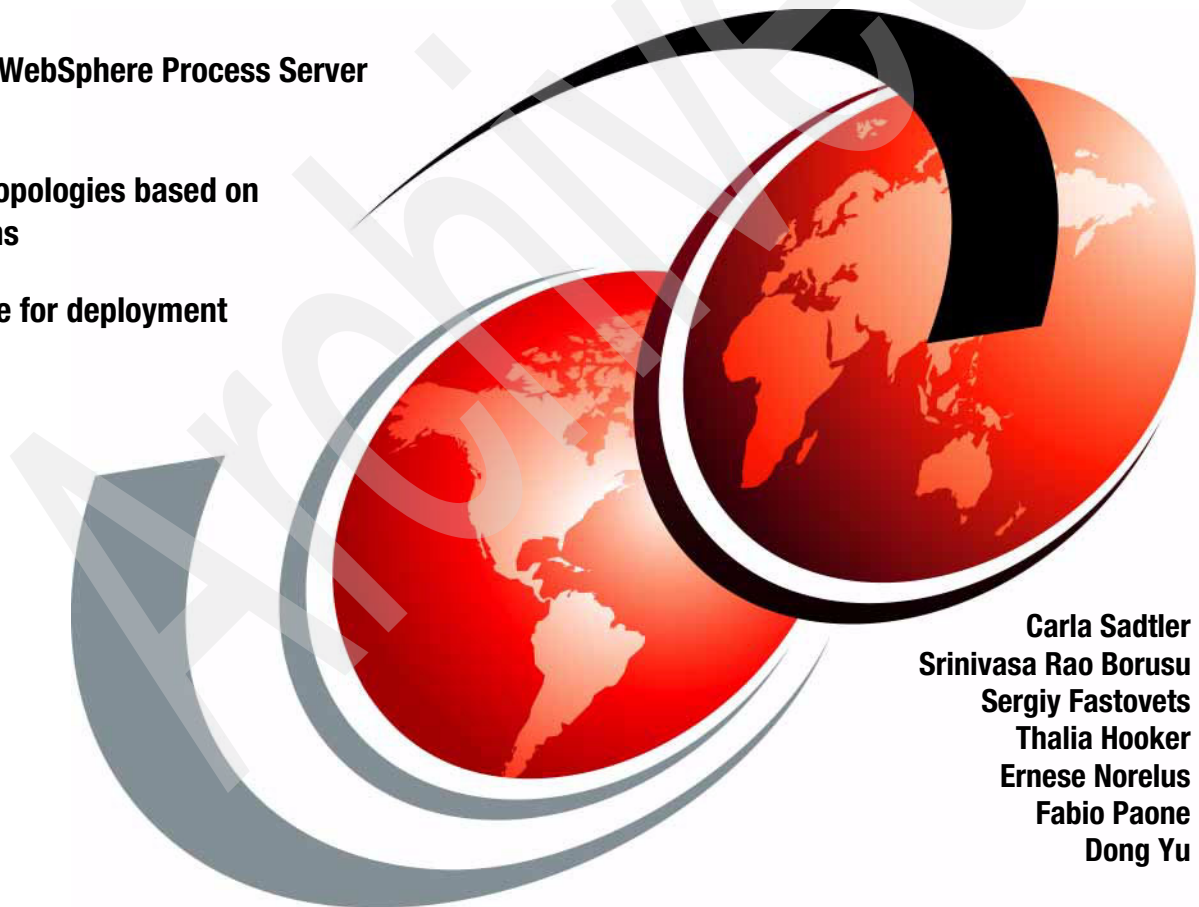


Getting Started with IBM WebSphere Process Server and IBM WebSphere Enterprise Service Bus Part 3: Run time

Install WebSphere Process Server

Build topologies based on patterns

Prepare for deployment



Carla Sadtler
Srinivasa Rao Borusu
Sergiy Fastovets
Thalia Hooker
Ernese Norelus
Fabio Paone
Dong Yu



International Technical Support Organization

**Getting Started with IBM WebSphere Process
Server and IBM WebSphere Enterprise Service Bus
Part 3: Run time**

July 2008

Archived

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

First Edition (July 2008)

This edition applies to IBM WebSphere Process Server V6.1 and IBM WebSphere Enterprise Service Bus V6.1.

© Copyright International Business Machines Corporation 2008. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	vii
Trademarks	viii
 Preface	ix
The team that wrote this book	ix
Become a published author	xi
Comments welcome	xii
 Chapter 1. Planning for installation	1
1.1 IBM WebSphere Application Server concepts	2
1.1.1 Profiles	3
1.1.2 Application server environments	4
1.1.3 IBM WebSphere Application Server for z/OS	7
1.2 Selecting a topology	10
1.2.1 Stand-alone server topology	10
1.2.2 Deployment environment patterns	11
1.2.3 Single cluster pattern	12
1.2.4 Remote messaging pattern	13
1.2.5 Remote messaging and remote support pattern	14
1.2.6 Web servers	15
1.3 Planning for installation	19
1.3.1 Prerequisites	19
1.3.2 User ID authorities for installation	19
1.3.3 Installation methods	20
1.3.4 Installation type (distributed and i5/OS)	22
1.3.5 Selecting additional features	22
1.3.6 Root directory	23
1.3.7 Creating profiles during installation	24
1.3.8 Installing recommended maintenance	24
1.4 Planning for profiles	24
1.4.1 Planning for databases	25
1.4.2 Security	30
1.4.3 Ports	32
1.5 Planning for the Profile Management Tool	34
1.5.1 Profile environment and type	34

Chapter 2. Installing on distributed and i5/OS systems	39
2.1 Overview of the installation	40
2.2 Typical installation	40
2.3 Installing on i5/OS	45
2.3.1 Installing WebSphere Process Server remotely from a Windows workstation	45
2.3.2 Installing WebSphere Process Server on i5/OS silently	49
2.4 Installing the IBM Update Installer	50
2.5 Downloading and installing maintenance	50
Chapter 3. Creating profiles on distributed and i5/OS systems	53
3.1 Methods used to build the profiles	54
3.1.1 Profile Management Tool	54
3.1.2 The manageprofiles command	56
3.2 Stand-alone WebSphere Process Server	58
3.3 Building a deployment environment using a pattern	69
3.3.1 Summary of steps	71
3.3.2 Building the Web server	72
3.3.3 Building the deployment manager	74
3.3.4 Building Node1 and the first process server of the cluster	78
3.3.5 Building Node2 and the second process server of the cluster	82
3.3.6 Creating the cluster for mediation modules	84
3.3.7 Defining the Web server to the cell	90
3.4 Augmenting a WebSphere ESB profile to WebSphere Process Server	94
Chapter 4. Creating and managing profiles on z/OS	97
4.1 Methods of creating profiles	98
4.2 Topology built in this chapter	100
4.3 Creating a deployment manager node	100
4.4 Creating a custom node	116
4.5 Configuring the deployment manager node	132
4.5.1 Running the zSMPInstall.sh	132
4.5.2 Preparing DmgrDB2.rsp	133
4.5.3 Running the zWPSCfg.sh	136
4.5.4 Creating DB2 database objects using the createDB.sh script	136
4.5.5 Backing up the WebSphere configuration HFS data sets	138
4.6 Configuring a custom node	138
4.6.1 Running the zSMPInstall script	139
4.6.2 Preparing ManagedDB2.rsp	140
4.6.3 Running zWPSCfg.sh	142
4.6.4 Federating a custom node	143
4.6.5 Configuring and federating a secondary custom node	144
4.6.6 Backing up the WebSphere configuration HFS data sets	145

4.7	Creating a WebSphere Process Server cluster	145
4.7.1	Adding DB2 libraries to the Servant and Adjunct JCL	145
4.7.2	Creating a cluster/server	146
4.7.3	Configuring Service Component Architecture	151
4.7.4	Configuring Common Event Infrastructure	156
4.7.5	Configuring the Business Flow and Human Task Managers	158
Chapter 5.	Managing the runtime environment	163
5.1	Administrative tools	164
5.1.1	Using commands	165
5.1.2	Using the administrative console	168
5.1.3	Using IBM Web Administration for i5/OS	171
5.1.4	WebSphere scripting client (wsadmin)	175
5.2	Managing stand-alone server environments	176
5.2.1	Managing application servers	176
5.3	Managing distributed environments	178
5.3.1	Managing the deployment manager	178
5.3.2	Managing nodes	180
5.3.3	Managing servers and clusters from the administrative console	182
5.4	Managing the Web server plug-in configuration	184
5.4.1	Generating the plug-in with administrative console	185
5.4.2	Enabling automated plug-in regeneration	185
5.4.3	Propagating the plug-in configuration file	186
5.4.4	Activating the new plug-in configuration	186
5.4.5	Enabling automated plug-in propagation	187
5.5	Viewing the JVM logs	187
Chapter 6.	Managing applications	189
6.1	Scenario: Order Management System	190
6.2	Planning for deployment	190
6.2.1	Setting up the prerequisites for Order Management System	195
6.3	Defining application resources	197
6.3.1	Setting the resource scope	198
6.3.2	Defining security credentials to access a resource	198
6.3.3	Defining JDBC resources	200
6.3.4	Defining users and groups	207
6.3.5	JMS resources	213
6.4	Deploying applications	214
6.4.1	Exporting applications for deployment	215
6.4.2	Installing applications	216
6.4.3	Configure and test Web service URLs	218
6.4.4	Updating endpoint URLs in Web service bindings	221
6.4.5	Example: Installing the Order Management System applications	221

6.5 Generating the Web server plug-in configuration	226
6.6 Testing the application	226
6.6.1 Starting and preparing your test clients.	227
6.6.2 Testing the mediation modules	229
6.6.3 Testing the business integration modules	231
Related publications	239
IBM Redbooks publications	239
Online resources	239
How to get IBM Redbooks publications	240
Help from IBM	240
Index	241

Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®
DB2®
i5/OS®
IBM®
Informix®
iSeries®
Lotus®

MVS™
OS/2®
OS/390®
RACF®
Rational®
Redbooks®
Redbooks (logo) ®

System i™
System z™
Tivoli®
WebSphere®
z/OS®

The following terms are trademarks of other companies:

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

EJB, J2EE, Java, JDBC, JMX, JVM, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Preface

This IBM® Redbooks® publication provides new users with information about how to install and configure IBM WebSphere® Process Server and IBM WebSphere Enterprise Service Bus runtime environment. It includes installation examples on distributed, i5/OS®, and z/OS® platforms. It also provides basic configuration information for deploying applications.

Some knowledge of IBM WebSphere Application Server Network Deployment is assumed for the readers of this book.

This book is the third book of a three-part series:

Getting Started with IBM WebSphere Process Server and IBM WebSphere Enterprise Service Bus:

- ▶ *Part 1: Development*, SG24-7608
- ▶ *Part 2: Scenario*, SG24-7642
- ▶ *Part 3: Run time*, SG24-7643

The team that wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), Raleigh Center.



Carla Sadtler is a Consulting IT Specialist at the ITSO, Raleigh Center. She writes extensively about WebSphere and Patterns for e-business areas. Before joining the ITSO in 1985, Carla worked in the Raleigh branch office as a Program Support Representative, supporting MVS™ customers. She holds a degree in mathematics from the University of North Carolina at Greensboro.



Srinivasa Rao Borusu is a Senior IT Specialist at IBM India Software Labs, Bangalore. He is currently working as a WebSphere Consultant with IBM Business Partner Technical Strategy and Enablement (BPTSE) Developer Services team, enabling and supporting worldwide business partners on WebSphere products. Since he joined IBM in 2000, he has played various roles, including Tech Lead for the IBM Java™ Virtual Machine Support during his tenure at OS/2® worldwide.

support center. His technical portfolio also includes one FILE rated patent, four IBM SOA and WebSphere product certifications, and several published articles. He holds a masters degree in Computer Applications from the Andhra University, Visakhapatnam, India.



Sergiy Fastovets joined IBM in 1996 in the Research Triangle Park. He has 10 years of experience in design, development, and sales support in the Host Integration area. In 2006, Sergiy transferred to the IBM UK where he is an IT Consultant for WebSphere Software Services. He holds a degree in mathematics from the University of Saint Petersburg, Russia. His native town is Poltava, Ukraine.



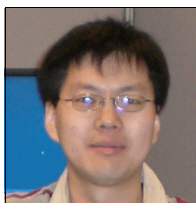
Thalia Hooker Ph.D., is a Consulting IT Specialist and member of the WebSphere Americas iPoC Team. She executes proof-of-concepts using the WebSphere platform to show customers how IBM solutions and products can help them meet their integration requirements. These solutions and products include Service Oriented Architecture (SOA) and WebSphere Business Process Management products such as, WebSphere Process Server and WebSphere Enterprise Service Bus.



Ernese Norelus is an IBM Certified Consulting IT Specialist with the ASEAN Software Services team in Singapore. He has been in IT for 10 years, with eight years of experience presenting, teaching, and proposing solution architectures to customers using the WebSphere business integration portfolio. He is also a well-known conference speaker, and he holds certifications in IM/DB2®, Lotus®, Rational®, Tivoli®, and WebSphere. He holds degrees in Biochemistry and Computer Science and Information Technology Management from the Université du Québec à Montréal, Québec, Canada.



Fabio Paone is in WebSphere technical sales supporting the Channel (Business Partners) in Italy. He has four years of experience as a developer in the Rome Tivoli Lab, primarily in J2EE™ development. He is certified as a WebSphere Application Server Administrator (versions 5, 6, and 6.1) and has participated in WebSphere Application Server certification test reviews.



Dong Yu is a Staff Software Engineer at IBM China Development Lab. He has four years of experience in the WebSphere Business Integration Field. His areas of expertise include WebSphere Process Server installation and Configuration. Dong holds a master degree in Software Engineering from Northwestern Polytechnical University.

Thanks to the following people for their contributions to this project:

Rich Conway
International Technical Support Organization, Raleigh Center

Margaret Ticknor
International Technical Support Organization, Raleigh Center

Stephen Cocks
IBM UK

Andrew Mead
IBM UK

Sharuff Morsa
IBM UK

Dean Barker
IBM UK

Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- ▶ Use the online **Contact us** review Redbooks form found at:

ibm.com/redbooks

- ▶ Send your comments in an e-mail to:

redbooks@us.ibm.com

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Planning for installation

This chapter discusses installation planning for IBM WebSphere Process Server installations. It is intended to provide the concepts and primary considerations for installation.

Although it is possible to install IBM WebSphere Process Server on an IBM WebSphere Application Server base, this chapter assumes IBM WebSphere Application Server Network Deployment as a base product.

This chapter also provides concepts for IBM WebSphere Enterprise Service Bus (ESB) within the context of an IBM WebSphere Process Server installation.

This chapter includes the following topics:

- ▶ IBM WebSphere Application Server concepts
- ▶ Selecting a topology
- ▶ Planning for installation
- ▶ Planning for profiles
- ▶ Planning for the Profile Management Tool

1.1 IBM WebSphere Application Server concepts

WebSphere Process Server and WebSphere ESB are built on top of WebSphere Application Server technology and inherit its runtime structure and configuration characteristics. The concepts that are associated with WebSphere Application Server are common for all of these products and are important to understand in order to plan for a WebSphere Process Server or WebSphere ESB environment.

WebSphere Application Server is organized based on the concept of cells, nodes, and servers. While all of these elements are present in each configuration, cells and nodes do not play an important role until you take advantage of the features that WebSphere Application Server Network Deployment provides.

The following concepts are key to understanding topologies based on WebSphere Application Server:

- ▶ The *application server* is the primary runtime component in all configurations and is where an application executes. Application servers have levels of capability that are tailored to the applications that they host. These levels are:
 - WebSphere Application Server and WebSphere Application Server Network Deployment is designed to run J2EE applications.
 - WebSphere ESB includes the functionality of WebSphere Application Server and can host mediation modules.
 - WebSphere Process Server includes the functionality of WebSphere ESB servers and can host business integration modules.

In a WebSphere Process Server installation, which includes WebSphere ESB, it is common to create all application servers with the WebSphere Process Server template.

- ▶ A *node* is a grouping of application servers for configuration and operational management on one system. Nodes are generally associated with a physical system. It is possible to have multiple nodes on a single system but nodes cannot span systems. A *node agent* in the node is the contact point for the deployment manager during cell administration.

As with application servers, nodes also have levels of capability that support their environment.

- ▶ A *cell* is a grouping of nodes into a single administrative domain. A cell can consist of multiple nodes, which are all administered from a single point (the *deployment manager*).

1.1.1 Profiles

Runtime environments are built by creating *profiles*. Each profile contains files that are specific to that run time, such as logs and configuration files. You can create profiles during installation and after. After you create a profile, you can further configure the runtime environment using WebSphere administrative tools that include scripting, commands, and the administrative console.

Each profile is stored in a unique directory path that is selected at profile creation time. The default is for the profiles to be stored in a subdirectory of the installation directory, but they can be located anywhere. All profiles share the product binaries.

A profile is created with one of the following levels of function:

- ▶ WebSphere Application Server Network Deployment
- ▶ WebSphere ESB
- ▶ WebSphere Process Server

Profiles for WebSphere Process Server and WebSphere ESB are actually profiles built for WebSphere Application Server Network Deployment and augmented with the process server or ESB function.

Profile types

There are three profile types that are used to define server environments:

- ▶ **Stand-alone server:** This type of profile creates an application server that is managed from its own administrative console. It functions independently from all other servers and deployment managers.

When the tool you use to create profiles offers the choice of creating a profile for multiple environments, the stand-alone server option is associated with a WebSphere Application Server Network Deployment profile. The profiles augmented for additional function are referred to as:

- **Stand-alone process server profile:** A stand-alone server profile that is augmented for WebSphere Process Server.
- **Stand-alone enterprise service bus profile:** A stand-alone service bus profile that is augmented for WebSphere ESB.
- ▶ **Deployment manager:** This type of profile creates a deployment manager that is used to administer nodes that are federated into its cell.
- ▶ **Custom:** A custom profile contains an empty node. No application server or deployment manager is defined by this profile. The typical use for a custom profile is to federate its node to a deployment manager. After federating the node, use the deployment manager to create a server or a cluster of servers within the node.

Augmenting profiles

You can augment WebSphere Application Server profiles (both base and Network Deployment) to have WebSphere ESB or WebSphere Process Server function. Likewise, You can augment WebSphere ESB profiles to have WebSphere Process Server function. However, the profile that you augment must exist on a system with a WebSphere Process Server installation.

On distributed and i5/OS systems, you can augment stand-alone servers or unfederated custom nodes using the Profile Management Tool or the **manageprofiles** command.

You must augment custom profiles that are federated to a cell manually. You must remove a federated profile from the deployment manager first, then augment it. Removing the node from the deployment manager restores the node to the configuration saved prior to using the **addNode** command. Data might be lost. You must also federate the deployment manager profile. Then, you can federate the custom profile back to the cell. You must also augment the deployment manager before federating the node back to the cell.

For more information, see *Augmenting existing profiles* (distributed and i5/OS platforms), which is available at:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/doc/tins_aug_profiles.html

For information about augmenting profiles on z/OS systems, see *zWPSCfg.sh* and *zWESBCfg.sh* (z/OS platforms), which is available at:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.z.610.doc/doc/rins_zos_configsript.html

1.1.2 Application server environments

This section describes the two basic topology types for WebSphere environments.

Stand-alone application servers

In a *stand-alone application server* environment, each application server acts as a unique entity. The cell consists of one node and one application server. The administration is managed through the application server.

Multiple stand-alone application servers can exist on a system, either by using independent installations of the WebSphere Process Server code or by creating multiple application server profiles within one installation. However, WebSphere Process Server does not provide centralized management or administration for

multiple stand-alone application servers. Stand-alone application servers do not provide workload management or failover capabilities.

Stand-alone servers are most often used for test environment and for non-production applications.

Profiles

A stand-alone server for WebSphere Process Server is created by defining a *stand-alone process server profile*. A stand-alone server for WebSphere ESB is created by defining a *stand-alone enterprise service bus profile*.

Distributed application servers

A *distributed server* configuration enables central administration, workload management, and failover. In this environment, you integrate one or more nodes into a cell that is managed by a *deployment manager*. Administration and management is handled centrally from the deployment manager.

Each node has one or more application servers. The nodes can reside on the same system as the deployment manager or on multiple separate systems. Each node has a *node agent* that works with the deployment manager to manage administration processes.

Figure 1-1 shows a runtime environment that consists of a single cell, a deployment manager that manages that cell, and two nodes, each with two application servers.

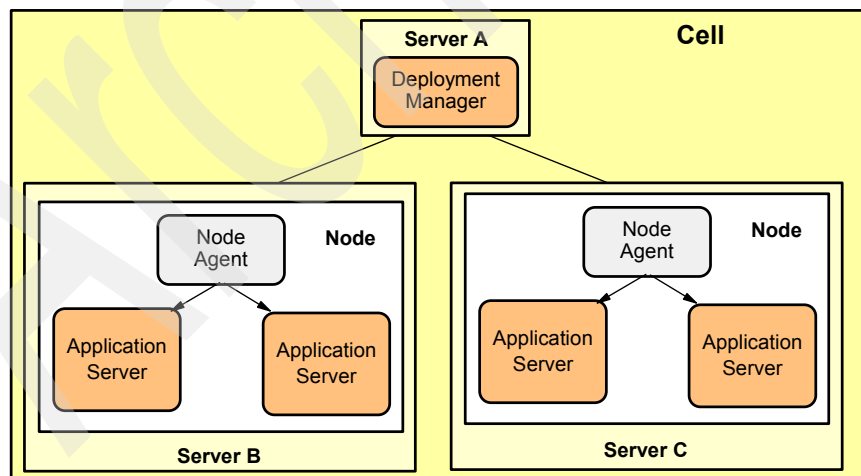


Figure 1-1 Distributed servers

With a distributed server configuration, you can create multiple application servers to run unique sets of applications and then manage those applications from a central location. However, more importantly, you can *cluster* application servers to allow for workload management and failover capabilities. Applications that you install in the cluster are replicated across the application servers. When one server fails, another server in the cluster continues processing. Workload is distributed among Web and EJB™ containers in a cluster using a weighted round-robin scheme.

Profiles

A distributed server environment consists of one deployment manager profile and one or more custom node profiles.

The easiest method to build a new WebSphere Process Server distributed environment is to use the deployment environment patterns. Each cell can have multiple deployment environments configured.

A typical process for creating a *new non-production* WebSphere Process Server environment is as follows:

1. Select a topology based on one of the available deployment environments.
2. Install WebSphere Process Server on the deployment manager host, and select the deployment environment that you want to build and the deployment manager option. This process builds a deployment manager profile and prepares the environment, including creating the databases that are required for WebSphere Process Server.
3. Install WebSphere Process Server on additional hosts (nodes), selecting the deployment environment and the custom profile option. This process creates a custom profile for each node, federates it to the cell, defines the application servers on the node, and adds the application servers to the appropriate cluster.

The result is fully functional WebSphere Process Server environment; however, this process gives you limited control over the database configuration.

In a production environment, it is recommended that you create the deployment manager first, then use the administrative console to create the deployment environment. This method provides the ability to use multiple databases for the WebSphere Process Server components.

A typical process for creating a *new production* WebSphere Process Server environment is as follows:

1. Install WebSphere Process Server on the deployment manager host, and create a deployment manager profile.
2. Install WebSphere Process Server, and create a custom profile on each node, allowing the federation to the deployment manager to occur during profile creation.
3. Use the administrative console to create a deployment environment. Using the administrative console allows you to select the nodes to add to each cluster and provides greater control over the database configuration options.

You can find examples of building topologies based on deployment environment patterns in Chapter 3, “Creating profiles on distributed and i5/OS systems” on page 53.

1.1.3 IBM WebSphere Application Server for z/OS

While WebSphere Application Server for z/OS has the same basic concepts that we discussed previously, it contains a unique process model that enables the product to manage many z/OS unique services and that provides Quality of Service (QoS). On z/OS, an application server is built using a federation of Java virtual machines (JVMs), each in a different process that together represents a single server instance. A server is composed of address spaces that actually run the code. WebSphere Process Server is built on this environment and takes advantage of this structure.

Figure 1-2 illustrates how WebSphere processes are structured in a z/OS environment.

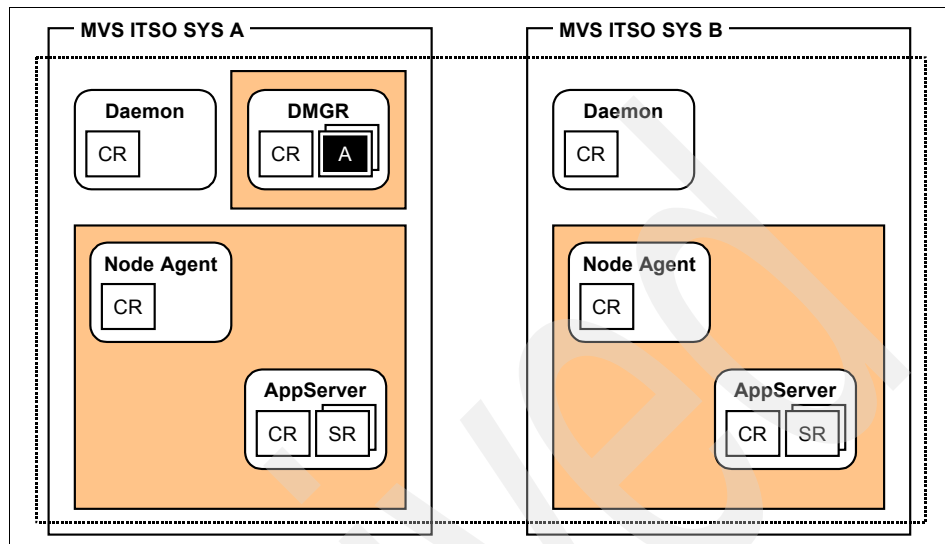


Figure 1-2 One cell, deployment manager, node on same LPAR, node separated

To understand how WebSphere on z/OS is unique from the distributed platforms, we briefly discuss some of the z/OS-specific concepts.

Address space

An *address space* is the area of successive virtual addresses that z/OS assigns to a user (or separately running program) for executing instructions and storing data. It is equivalent to a process on distributed platforms.

Control region

The *control region* (CR) is basically the only public interface to this collection of JVMs that, all together, represent a single application server. All requests go through the CR and the CR forwards them to one of the potentially many servant controllers for processing. In short, a CR is similar to a router or even an address space that binds the TCP ports used by the server. A CR does have an embedded JVM™, which is the only JVM that is allowed to receive connections from the outside world. Each server has only one CR that is started through a JCL start procedure.

The requests arrive in the CR process, which then works with the z/OS workload manager (WLM) to dispatch the work to the servant regions (Figure 1-3).

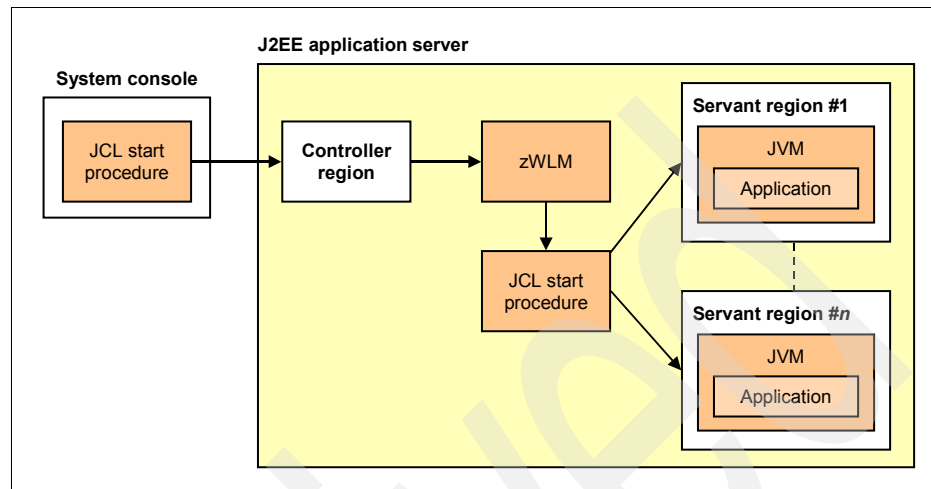


Figure 1-3 Inside the application server

Control region adjunct

The *control region adjunct* is a specialized servant that interfaces with new service integration buses to provide messaging services.

Servant region

The *servant region* (SR) is where the requests are actually processed and is equivalent to the application server on a distributed environment platform. All the SRs are identical and have the same J2EE level. The SR depends on the CR for many services such as communication, security and transaction control.

When multiple SRs are created, a copy of each application is found in each SR and the CR forwards the requests to the appropriate SR.

zWLM

The z/OS workload manager manages resources to ensure that performance goals are met. It is a part of z/OS. To differentiate this from the workload management of WebSphere, we refer to this as *zWLM*.

As the CR receives incoming requests, it works with zWLM to ensure that these requests are classified according to organization-defined rules and are dispatched appropriately to servant regions that can handle the load. zWLM can alter factors to ensure that performance goals are met, for example, by updating importance levels of services classes and starting additional servant regions.

Daemon

A *daemon* server provides the location name service for external clients. There is one daemon per cell per MVS image. If your cell consists of multiple MVS images, a daemon is created for each MVS image where your cell exists. If there are two cells on the same MVS image, two daemons are created. Each daemon server consists of a single CR.

Daemon servers are started automatically when the first server for the cell on that MVS image is started. If you kill a daemon, all the servers for that cell on that MVS image come down also.

1.2 Selecting a topology

To plan an installation, you need to design the topology that you want to build. Choosing the correct topology is important regarding future requirements for high availability, scalability, and performance. The topology that you select determines the number of systems on which you need to install the product, the profiles that you need to create, and your path through the installation and profile creation process.

This section describes common topologies, starting with a simple stand-alone server topology and progressing to more complex topologies that are identified as common deployment patterns.

Note: For a more in-depth discussion of topologies for WebSphere Process Server and WebSphere ESB topologies, see:

- ▶ *Production Topologies for WebSphere Process Server and WebSphere ESB V6*, SG24-7413
- ▶ *z/OS Getting Started: WebSphere Process Server and WebSphere Enterprise Service Bus V6*, SG24-7378

1.2.1 Stand-alone server topology

The simplest topology is one that consists of a stand-alone server for WebSphere Process Server and WebSphere ESB. All components run on a single server and both modules and mediation modules run in the application server, as shown in Figure 1-4.

With this topology, there is no central configuration management. The application server has its own administrative console. There are no workload management or high availability features with this topology.

This topology is most often seen for non-critical applications and in test environments.

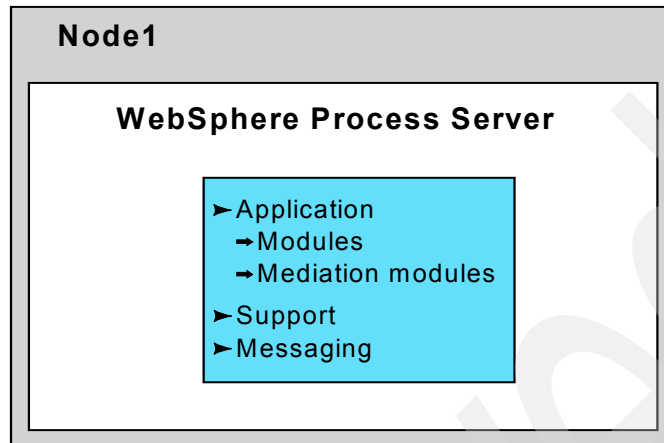


Figure 1-4 Stand-alone server topology

Example: For an example of this topology, see 3.2, “Stand-alone WebSphere Process Server” on page 58.

1.2.2 Deployment environment patterns

A set of deployment environment patterns are identified that represent common deployment scenarios. WebSphere Process Server provides built-in support for these patterns in the installation process, the profile creation tools, and the administrative console. These patterns assist in creating the profiles and configuration settings that define complex deployment environments, including the creation of the supporting databases and clusters.

The three patterns are:

- Single cluster pattern
- Remote messaging pattern
- Remote messaging and remote support pattern

Using this support is recommended on distributed and i5/OS systems. The patterns that are defined by this support are also legitimate patterns for a z/OS environment, but at the time of this publication, the use of the deployment environment options in the profile creation tool and the administrative console are not recommended.

Cluster pattern components

The patterns use application server clustering to provide failover and load balancing for components in the topology. The support for these components is broken down into the following categories:

- ▶ Messaging components that support the WebSphere Process Server and WebSphere ESB environment:
 - Service Component Architecture (SCA) application bus, which exists as a default location for the queues that are used by JMS export components
 - SCA system bus
 - Business Process Choreographer bus
 - Common Event Interface (CEI) bus
- ▶ Support components for the WebSphere Process Server and WebSphere ESB environment:
 - CEI server
 - Business Rules Manager
 - BPC Explorer and BPC Observer
- ▶ Application components
 - Application deployment target

Using patterns for WebSphere Process Server and WebSphere ESB:

When used to create a WebSphere Process Server environment, the application deployment target can host modules, mediation modules, or both. When used to create a WebSphere ESB environment, the application deployment target can host mediation modules only.

For examples of combining these patterns to create more complex deployment environments, see Chapter 3, “Creating profiles on distributed and i5/OS systems” on page 53.

1.2.3 Single cluster pattern

The *single cluster pattern* is the simplest of the clustering patterns. It defines one cluster that supports applications, messaging, and the support components. This pattern is recommended for the scenarios where most invocations are synchronous. Messaging use should be minimal and applications role should dominate the use cases.

The cluster can consist of multiple nodes and multiple servers on each node (as shown in Figure 1-5). A cluster consisting of WebSphere Process Server nodes and servers can host both business integration and mediation applications. Applications are installed to the cluster.

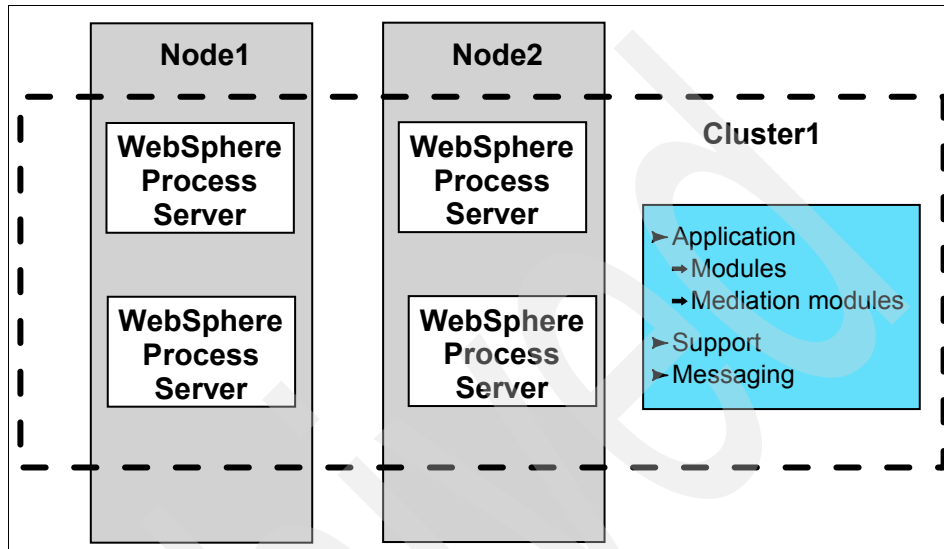


Figure 1-5 Single cluster pattern

Example: For an example of this topology, see 3.3, “Building a deployment environment using a pattern” on page 69.

1.2.4 Remote messaging pattern

The *remote messaging pattern* defines one cluster for application deployment and one remote cluster for the messaging infrastructure. The common event infrastructure and other support applications are configured on the application deployment cluster. This pattern is recommended for scenarios that involve heavy asynchronous invocations. However, this pattern is not ideal when other supporting runtime components are actively involved.

Applications are installed to the application deployment cluster.

In this pattern, the messaging cluster is assigned as a member of the service integration buses that provide support for the WebSphere Process Server and WebSphere ESB infrastructure. The SCA.Application bus has one bus member (the messaging cluster) by default. See Figure 1-6.

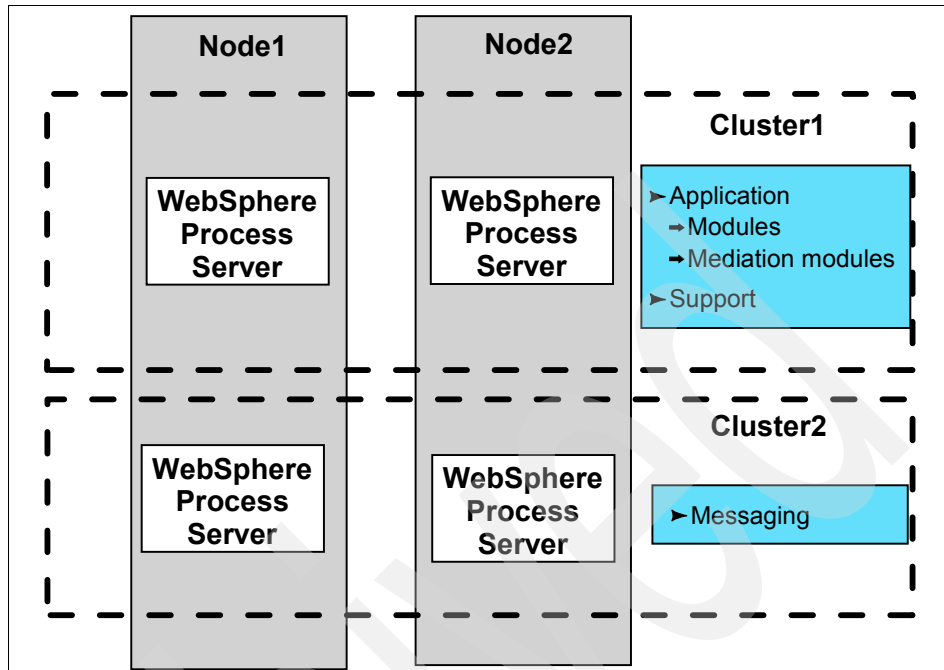


Figure 1-6 Remote messaging pattern

1.2.5 Remote messaging and remote support pattern

The *remote messaging and remote support pattern* defines one cluster for application deployment, one remote cluster for the messaging infrastructure, and one remote cluster for the common event infrastructure and other support applications (as shown in Figure 1-7). This pattern performs well and provides the most scalable, flexible, and versatile topology pattern. If in doubt, select this pattern.

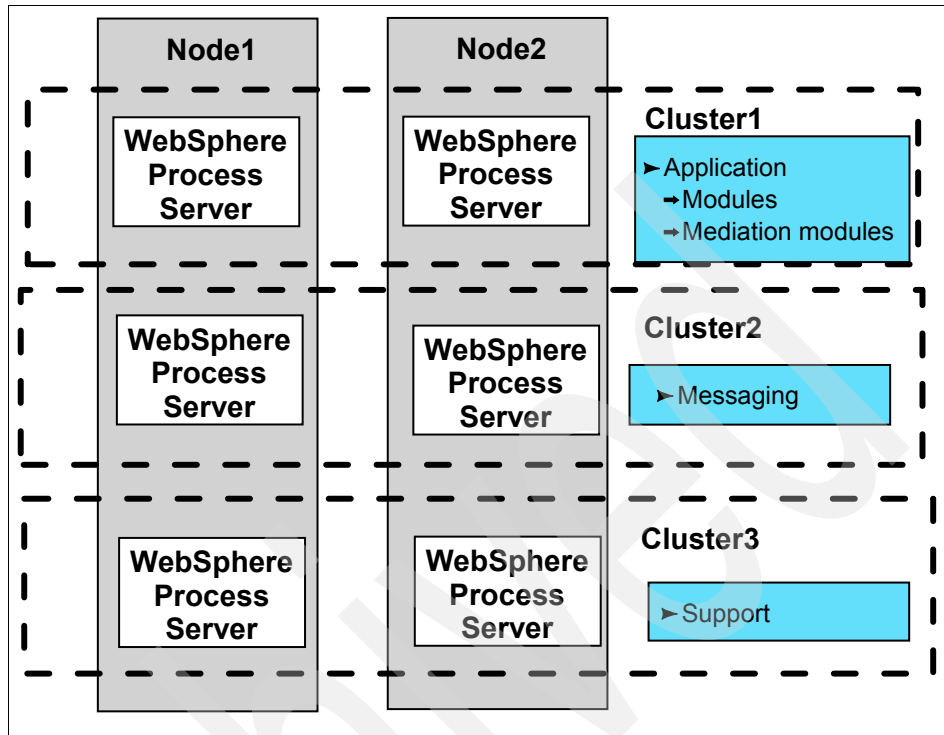


Figure 1-7 Remote messaging and remote support pattern

Example: For an example of this topology, see 3.3, “Building a deployment environment using a pattern” on page 69.

1.2.6 Web servers

WebSphere Application Server provides Web server plug-ins that work with a Web server to route requests from the Web server to the proper application server, as shown in Figure 1-8. A Web server plug-in is specific to the type of Web server. It is installed on the Web server system and configured in the Web server configuration.

A plug-in configuration file that is generated on the application server and that is placed on the Web server is used for routing information. To manage the generation and propagation of these plug-in configuration files, Web servers are defined to the WebSphere configuration repository. In some cases, Web server configuration and management features are also available from the WebSphere administrative tools.

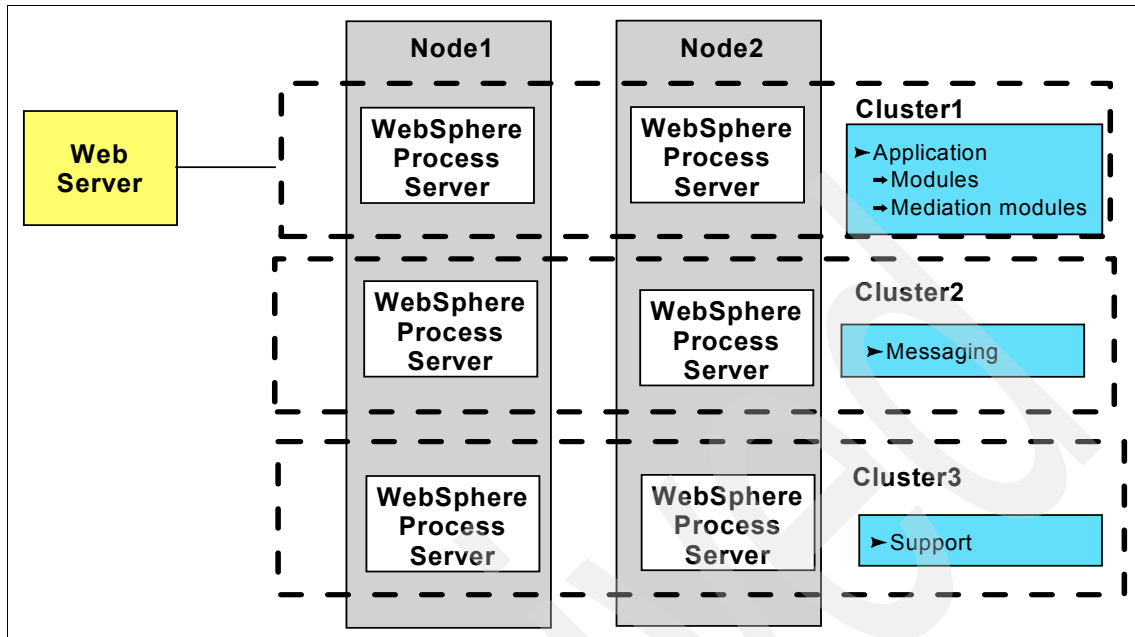


Figure 1-8 Remote messaging and remote support pattern

Managed Web servers versus unmanaged

A supported Web server can be on a managed node or an unmanaged node, depending on the environment on which you are running the Web server.

Basic administrative functions are provided in the WebSphere administrative tools for all supported Web servers. For example, you can generate a plug-in configuration for all Web servers. If the Web server is defined on a managed node, you can propagate the plug-in configuration automatically using node synchronization. If the Web server is defined on an unmanaged node, automatic propagation of a plug-in configuration is only supported for IBM HTTP Servers.

Some additional administrative console tasks are supported for IBM HTTP Servers on managed and unmanaged nodes. For example, you can start IBM HTTP Servers, stop them, terminate them, display their log files, and edit their configuration files.

Unmanaged nodes

An *unmanaged node* does not have a node agent to manage its servers. In a stand-alone server environment, you can define one Web server and it, by necessity, resides on an unmanaged node. In a distributed server environment, Web servers defined to an unmanaged node are typically remote Web servers.

If the Web server is defined to an unmanaged node, follow these steps:

1. Check the status of the Web server.
2. Generate a plug-in configuration file for that Web server.

If the Web server is an IBM HTTP Server and the IHS Administration server is installed and properly configured, you can also:

- a. Display the IBM HTTP Server Error log (error.log) and Access log (access.log) files.
- b. Start and stop the server.
- c. Display and edit the IBM HTTP Server configuration file (httpd.conf).
- d. Propagate the plug-in configuration file after it is generated.

You cannot propagate an updated plug-in configuration file to a non-IBM HTTP Server Web server that is defined to an unmanaged node. You must install an updated plug-in configuration file manually to a Web server that is defined to an unmanaged node.

Managed nodes

In a distributed server environment, you can define multiple Web servers. You can define these Web servers on managed or unmanaged nodes. A *managed node* has a node agent.

If the Web server is defined to a managed node, follow these steps:

1. Check the status of the Web server.
2. Generate a plug-in configuration file for that Web server.
3. Propagate the plug-in configuration file after it is generated.

If the Web server is an IBM HTTP Server and the IBM HTTP Server Administration server is installed and properly configured, you can also:

- a. Display the IBM HTTP Server Error log (error.log) and Access log (access.log) files.
- b. Start and stop the server.
- c. Display and edit the IBM HTTP Server configuration file (httpd.conf).

Defining nodes and servers

During the installation of the plug-in, the Plug-ins installation wizard creates a Web server configuration script named `configure<Web_server_name>`. You use this configuration script to define the Web server definition and, if necessary, the node definition to WebSphere.

If a Web server definition already exists for a stand-alone application server, running the script does not add a new Web server definition. Each stand-alone application server can have only one Web server definition. A distributed server environment, alternatively, can have multiple Web server definitions. The script creates a new Web server definition unless the Web server name is the same.

The Plug-ins installation wizard stores the script in the `<plug-in_home>/bin` directory on the Web server system. If the plug-in is installed locally (on the same machine as the application server), the configuration script runs automatically.

For remote installations, you must copy the script from the Web server system to the `WPS_install_root/bin` directory on the application server system for execution. The script runs against the default profile. If one system is running under Linux® or UNIX® and the other system is running under Windows®, use the script that is created in the `<plug-in_home>/bin/crossPlatformScripts` directory.

Note: Always open a new command window in which to execute the `configure<Web_server_name>` script. There is a potential conflict between a shell environment variable, the `WAS_USER_SCRIPT` variable, and the real default profile. The script always works against the default profile. However, if the `WAS_USER_SCRIPT` environment variable is set, a conflict arises because the script attempts to work on the profile that is identified by the variable.

If you are federating a stand-alone application server into a cell, any Web server definitions that are created for a stand-alone application server are lost when you federate them. If you are creating a distributed server environment this way, wait until after federating your application servers to create Web server definitions.

For more information about the installation of Web server plug-ins and how the Web server definitions scripts are generated and executed, see *WebSphere Application Server V6.1: Planning and Design*, SG24-7305.

1.3 Planning for installation

Note: In this book, we show examples of various installation and profile creation scenarios. However, this book is not intended as a comprehensive planning guide.

Refer to the WebSphere Process Server V6.1.0 information center for more in-depth planning and installation topics:

- ▶ *Planning for WebSphere Process Server (distributed and i5/OS)*
http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/doc/cpln_plancons.html
- ▶ *Planning for WebSphere Process Server (z/OS)*
http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.z.610.doc/doc/cpln_plancons.html

In addition, the following document provides valuable guidance in planning for a WebSphere for z/OS installation:

- ▶ *WebSphere z/OS V6—WSC Sample ND Configuration*
<http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100653>

1.3.1 Prerequisites

Before starting the installation process, check the following Web site to make sure that you have met the prerequisites:

- ▶ WebSphere Process Server detailed system requirements
<http://www-1.ibm.com/support/docview.wss?uid=swg27006205>

1.3.2 User ID authorities for installation

The user ID that is used for installation is preferably one with Administrator authorities. However, it is possible to use a “non-root” user ID. Non-root installation allows you to perform the installation as a non-root user on Linux or UNIX operating systems, or as a non-administrator on Windows.

If you decide to use a non-root user ID to perform the installation, be sure to review the following information center articles:

► *Limitations of non-root installers*

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rins_nonroot.html

► *Granting write permission of files and directories to a non-root user for profile creation*

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/doc/iins_nonrootprofile.html

To install on an i5/OS system, you must use a valid i5/OS user profile for that system. The user profile must have *ALLOBJ and *SECADM authorities. Without these authorities, the installation will fail.

To install on z/OS, you must have administrator authority.

1.3.3 Installation methods

Before starting the installation, you need to be aware of the options that you have for the method of installation. This section describes the different installation methods.

Distributed systems

You can install on distributed systems interactively or silently. To initiate an interactive installation, execute the launchpad interface and select the option to install WebSphere Process Server. For an example, see Chapter 2, “Installing on distributed and i5/OS systems” on page 39.

To perform a silent installation, build a response file and execute the **install** command.

Both installation methods install the binary files and, optionally, allow you to create one profile. You can create additional profiles after installation using the PMT or **manageprofiles** command.

For more information about installation, see *Installing and configuring WebSphere Process Server* (distributed and i5/OS), which is available at:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/doc/welcome_wps_ins.html

i5/OS

You can install on i5/OS using one of the following methods:

- ▶ Interactively from a Windows workstation
- ▶ Silently from a Windows workstation
- ▶ Silently on the i5/OS system

You can find examples of an interactive installation and a silent installation on i5/OS in Chapter 2, “Installing on distributed and i5/OS systems” on page 39.

The installation installs the binary files and, optionally, allows you to create one profile. You can create additional profiles after installation using the PMT on a Windows workstation or using the **manageprofiles** command on i5/OS.

For more information about installation, see *Installing and configuring WebSphere Process Server* (distributed and i5/OS), which is available at:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/doc/welcome_wps_ins.html

z/OS

You installation on z/OS in phases as follows:

1. Use SMPE to install and configure WebSphere Application Server for z/OS.
2. Use SMP/E to install the WebSphere Process Server product binaries into a hierarchical file structure (HFS) or zFS.
3. Create the WebSphere Application Server profiles that are required for the topology that you select using the customization dialog box or zPMT.
4. Run the zSMPInstall.sh script to install WebSphere Process Server to each WebSphere Application Server profile directory.
5. Create the appropriate databases and storage groups that are required by WebSphere Process Server.
6. Run the zWPSCfg.sh to configure the server as a process server and to enable the server to handle business processes and mediations.

Note: When you purchase WebSphere Process Server for z/OS, you get the WebSphere ESB run time that allows you to run mediations in a WebSphere Process Server configured server. You cannot build an ESB-only server. There is a zWESBConfig.sh script that ships with WebSphere Process Server, but it does not work.

To build an ESB-only server, you need to purchase, install, and configure the WebSphere ESB for z/OS product separately.

For information about these steps, see the information center, starting with the article *Installing and configuring WebSphere Process Server (z/OS)*, at:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.z.610.doc/doc/iins_zos_welcome.html

1.3.4 Installation type (distributed and i5/OS)

On distributed and i5/OS systems, you have three options for installation.

- ▶ **Typical Installation:** A full installation of WebSphere Process Server. This installation path allows you to define a WebSphere ESB or WebSphere Process Server environment consisting of one stand-alone server, a deployment manager, or a custom profile during the installation process. The common database is created using the default for the operating system. You can also elect to create no profiles during installation.
- ▶ **Deployment Environment Installation:** A full installation of WebSphere Process Server. This installation choice guides you through the process of setting up a deployment environment. You can choose a deployment manager based on a deployment environment pattern or continue to define a deployment environment that you have already created. This option allows you to select from a limited list of database providers for the common database. After installation, you create additional profiles using the Profile Management Tool or `manageprofiles` command.
- ▶ **Client Installation:** A partial installation of WebSphere Process Server that allows you to run a client application that interacts with a WebSphere ESB and WebSphere Process Server within the same cell.

1.3.5 Selecting additional features

During the installation, you have the option of selecting the following features that are not installed by default:

- ▶ Business rule beans (deprecated)

This option is available on all platforms with the exception of i5/OS.

This function is carried forward from WebSphere Business Integration Server Foundation, version 5.1.1. It will not be supported in future versions of WebSphere Process Server.

Business rule beans has been replaced by business rules.

For more information, see:

- Overview of business rules

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/doc/cadm_busiru_overview.html

- Deprecated features

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/doc/gmig_deprecationlist.html

- ▶ Extended messaging service (deprecated)

This option is available on all platforms with the exception of i5/OS.

This function is carried forward from WebSphere Business Integration Server Foundation, version 5.1.1. It will not be supported in future versions of WebSphere Process Server. Instead of using the Extended Messaging Service and its associated tools, you need to use the standard JMS APIs, or equivalent messaging technologies.

For more information, see:

- Deprecated features

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/doc/gmig_deprecationlist.html

- ▶ WebSphere Process Server samples

The Samples Gallery contains sample that demonstrate WebSphere technology. They are appropriate for demonstration and development environments, but are not recommended for production environments.

For more information about educational material, see:

- Samples and tutorials

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/doc/covw_sample_tutorial.html

1.3.6 Root directory

You need to specify a location for the installation. You can find the default root directories in:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/doc/rins_installation_paths.html

1.3.7 Creating profiles during installation

Determining the profiles that you need depends on the topology that you plan to run. With the exception of stand-alone topologies, every topology needs multiple profiles. With distributed and i5/OS installations, you have the option of creating a profile during installation. After installation, you can create additional profiles using the **manageprofiles** command or the Profile Management Tool.

Creating the first profile during installation is simply a shortcut to get you started.

1.3.8 Installing recommended maintenance

After installation, you need to install the Update Installer for WebSphere Software and all recommended fixes. You can install the Update Installer from the launchpad.

You can find the recommended fixes at:

- ▶ Recommended Fixes for WebSphere Process Server
<http://www-1.ibm.com/support/docview.wss?rs=2307&uid=swg27006649>
- ▶ Mandatory critical fixes for WebSphere Process Server for z/OS V6.1
http://www-1.ibm.com/support/docview.wss?rs=2307&context=SSQH9M&q1=P K59140&uid=swg27011675&loc=en_US&cs=utf-8&lang=en

1.4 Planning for profiles

You need to make some decisions before you create a profile. These decisions are related to the topology and runtime environment and are logically separate from those made for installation. However, because you can create profiles during installation, you need to consider these topics during initial planning.

You can create profiles on distributed and i5/OS systems using one of the following methods:

- ▶ Using the Profile Management Tool (a wizard-driven tool)
- ▶ During installation
- ▶ Using the **manageprofiles** command

On z/OS systems, the following path is recommended for creating profiles:

1. Use the zPMT to build JCL required to create the WebSphere Network Deployment profiles.
2. Augment the profiles for WebSphere Process Server or WebSphere ESB.

Configuring WebSphere Application Server for z/OS requires a fair amount of planning and coordination. For example, when defining multiple deployment managers or application servers on a single machine or LPAR, you need to ensure that the ports and names that you select for each are unique and that the z/OS environment variables, generated jobs, and so on are set up properly. We strongly recommend that you spend time planning the installation and, if possible, practice by configuring a stand-alone application server using the default options.

1.4.1 Planning for databases

WebSphere Process Server uses databases during run time to support the execution of business processes and mediations. The set of tables that make up each database can be stored in a separate physical database or can be a combined with others in a larger common database. The database configuration that you choose depends on the features that you use as well as performance considerations. You can create the databases when you create the profile, or you can choose to delay creating them until after the profile is created but before it is started.

Logical databases include:

- ▶ WebSphere Process Server components and the enterprise service bus use a common database. The default name for this database is WPRCSDB.
- ▶ A Common Event Infrastructure database, referred to as the *CEI database*. The default name for this database is EVENT.
- ▶ A Business Process Choreographer database, referred to as the Business Process Execution database. The default name for this database is BPEDB.
- ▶ A Business Process Execution Observer database, referred to as the observer database. The default name for this database is OBSRVDB.
- ▶ A database for the messaging engines. The default name for this database is MEDB.

i5/OS: On i5/OS systems, there is a single global database in which you define all schemas for all functional components. You must make sure that all schema names are unique within the logical partition (LPAR).

There are also unique considerations for selecting the driver that is needed to connect remotely to an i5/OS database server. For more information, see:

► *Database specifications*

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/doc/cins_db_specs.html

You can find detailed requirements for database products in:

► *WebSphere Process Server detailed system requirements*

<http://www-1.ibm.com/support/docview.wss?rs=2307&context=SSQH9M&uid=swg27006205>

Deciding when and how to create the databases

The minimal configuration for a WebSphere Process Server or WebSphere ESB configuration is the common database that contains all the database tables. *This is not recommended for a production system.* This is the configuration that you get if you use the Profile Management Tool to create the databases. In this series of books, we use this configuration.

For production systems, it is recommended that you create the databases and update the appropriate data sources manually. This gives you maximum control over the location and grouping of database tables. To do this, you choose the option to delay the creation of the databases when you create the profiles, which generates scripts to help you create the common and event databases and creates the data sources based on information that you provide during profile creation. Create the databases using the scripts or your own commands and update the appropriate data sources.

Finding the database scripts and specifications

You can find scripts for the common database, the BPEDB and the OBSRVDB in *WPS_install_root/dbscripts*. These scripts are available after install.

During the profile creation with the Profile Management Tool, a new set of scripts is generated into *profile_root/dbscripts*. These scripts create the common database with the attributes you specified during profile creation.

Scripts to create the Business Process Choreographer tables and schema are generated when you configure BPC. This can be during or after profile creation or as part of the deployment environment configuration. These are also stored in

profile_root/dbscripts. You need to create the BPEDB database and then run these scripts.

For the messaging engines, you can use the *sibDDLGenerator* script to generate the database scripts. You can find information about the messaging engine databases at:

- ▶ *Messaging engine database specifications*
http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/doc/cins_messaging_engine_db_specs.html
- ▶ *Messaging engine database specifications (z/OS)*
http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.z.610.doc/doc/cins_messaging_engine_db_specs.html

Common database (WPRCSDB)

WebSphere Process Server components and the enterprise service bus use a common database. The default name for this database is WPRCSDB.

The common database is used by the following product components:

- ▶ Recovery
- ▶ Relationship service
- ▶ Mediation
- ▶ Application scheduler
- ▶ Customization (selector and business rule group)
- ▶ EventSequencing (LockManager)
- ▶ WebSphere ESB Message Logger primitive

Depending on the database provider that you select, you have the option to have the database created automatically during profile creation. The option to create a new database is disabled for the following database types:

- ▶ DB2 for z/OS and OS/390® v7
- ▶ DB2 for z/OS v8
- ▶ DB2 for z/OS v9
- ▶ Oracle® 9i and 10g

If you elect to defer the database creation, scripts are created for your use in defining the database. The information that you enter for the existing database (whether or not it actually exists yet) must be correct, because it is used to create the data source and the scripts. The scripts are located in *profile_root/dbscripts/CommonDB/db Type/dbName*.

For more information, see:

► *Common database specifications*

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/doc/cins_common_db_specs.html

► *Common database specifications (z/OS)*

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.z.610.doc/doc/cins_common_db_specs.html

Table 1-1 shows the database provider options for the Common database.

Table 1-1 Options for common database on distributed systems

Database product	
Derby Embedded	<ul style="list-style-type: none">► Install (typical): Stand-alone profile (the default - no option)► PMT (typical): Stand-alone profile (the default - no option)► PMT (advanced): Stand-alone profile► Administrative console (deployment environment)
Derby Network	<ul style="list-style-type: none">► Install (typical): Deployment manager (the default - no option)► Install (Deployment Environment): Deployment manager► PMT (advanced): All profiles► Administrative console (deployment environment)
DB2 UDB	<ul style="list-style-type: none">► Install (Deployment Environment): Deployment manager► PMT (advanced): All profiles► Administrative console (deployment environment)
DB2 for i5/OS Native Driver	<ul style="list-style-type: none">► Install and PMT stand-alone profiles► Administrative console (deployment environment)
DB2 for i5/OS Toolbox Driver	<ul style="list-style-type: none">► PMT (advanced): All profiles► Administrative console (deployment environment)
DB2 for z/OS	<ul style="list-style-type: none">► PMT (advanced): All profiles► Administrative console (deployment environment)
DB2 Call-Level Interface Provider (CLI), DB2 Universal Runtime Client	<ul style="list-style-type: none">► PMT (advanced): All profiles► Administrative console (deployment environment)
Oracle 9i and 10g	<ul style="list-style-type: none">► Install (Deployment Environment): Deployment manager► PMT (advanced): All profiles► Administrative console (deployment environment)

Database product	
Informix® Dynamic Server	<ul style="list-style-type: none"> ▶ PMT (advanced): All profiles ▶ Administrative console (deployment environment)
Microsoft® SQLServer (Embedded Driver and DataDirect)	<ul style="list-style-type: none"> ▶ PMT (advanced): All profiles ▶ Administrative console (deployment environment)

If you elect to create the common database outside of the installation or profile creation, see the following article:

- ▶ *Creating the Common database and tables after profile creation or augmentation*

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/doc/tins_new_dbcreate.html

Business Process Choreographer databases

The Business Process Execution database (BPEDB) is used by the Business Flow Manager and the Human Task Manager.

In a non-production system, the BPEDB, OBSRVDB, and messaging engine databases can all use the common database (WPRCSDB). This is the configuration that you get as a result of using the installation wizard or PMT to create a deployment environment.

In a production system, you can use the common database for all three databases, but you might see better performance using separate databases for the functions that you plan to use.

For each deployment target where the Business Process Choreographer is used, create a BPEDB database that is separate from the common database. If you use the Business Process Choreographer Observer, this can use the same BPEDB or, for better performance, can be a separate database.

The Business Process Choreographer messaging engines can either share the database that used by the SCA messaging engines or can have their own BPEDB database.

For information about planning for the Business Process Choreographer databases, including BPEDB, OBSRVDB, and BPEMEDB databases, see:

- ▶ *Planning the databases for Business Process Choreographer*
http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.bpc.610.doc/doc/bpc/t1p1_database.html
- ▶ *Planning the databases for Business Process Choreographer (z/OS)*
http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.bpc.z.610.doc/doc/bpc/t1p1_database.html

Common event database

The event database is used by the Common Event Infrastructure to capture events as they occur, as follows:

- ▶ When you create a stand-alone profile, the event database is created for you.
- ▶ In distributed environments, the event database is not created when you create a deployment manager profile or managed profile. You must create and configure the database.
- ▶ When you create a deployment environment, you can specify a database to use for the event database. You have the option to have the tables created as part of the deployment environment creation process or you can choose to run the scripts later to create the tables.

For more information about creating and configuring an event database, see:

- ▶ *Common Event Infrastructure database specifications for distributed and i5/OS systems*
http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/doc/cins_cei_db_specs.html
- ▶ *Common Event Infrastructure database specifications for z/OS systems*
http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.z.610.doc/doc/cins_cei_db_specs.html

1.4.2 Security

Security for WebSphere Process Server is provided by the WebSphere Application Server infrastructure. Security is a topic that cannot be addressed adequately in a single section of this chapter. However, it is important that you understand the concepts of administrative security and application security to install and set up your installation adequately.

Administrative security

Administrative security requires a user ID and password to log in to the administrative console and to enter WebSphere administrative commands from a command line.

When you create a stand-alone application server or deployment manager, you are given the option to enable security. If you are creating a deployment manager in a deployment environment, administrative security is not optional. It is enabled when the profile is created.

If you intend to create a profile during installation and want to secure your administrative environment at the same time, you need to identify one user ID that is used for administration. The user ID and password that you specify during profile creation is created in the repository and assigned the Administrator role. This ID can be used to access the administration tools and to add additional user IDs for administration.

When you enable security during profile creation, LTPA is used as the authentication mechanism.

On distributed systems, a file-based user repository is created and populated with the administrator ID. This file-based system can be federated with other repository types to form an overall repository system. If you do not want to use the file-based repository, do not enable administrative security during profile creation. In WebSphere for z/OS, you can choose to use the file-based repository or use the z/OS system SAF-compliant security database.

Whether you choose to enable administration security during profile creation or after, it is important that you do it *before* going into production.

Application security

Application security enables security for the applications in the cell. This type of security is disabled by default in WebSphere Application Server profiles, but is enabled by default in WebSphere Process Server profiles. Before deploying applications that implement security, you must review and configure the application security settings for your environment.

For information about securing applications, see *IBM WebSphere Application Server V6.1 Security Handbook*, SG24-6316.

1.4.3 Ports

Each WebSphere process uses a set of TCP/IP ports. These must be unique to the system. Ports used for WebSphere Process Server and WebSphere ESB servers are the same as those used for WebSphere Application Server.

The default ports for i5/OS and distributed platforms are listed in the following information center article:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rmig_portnumber.html

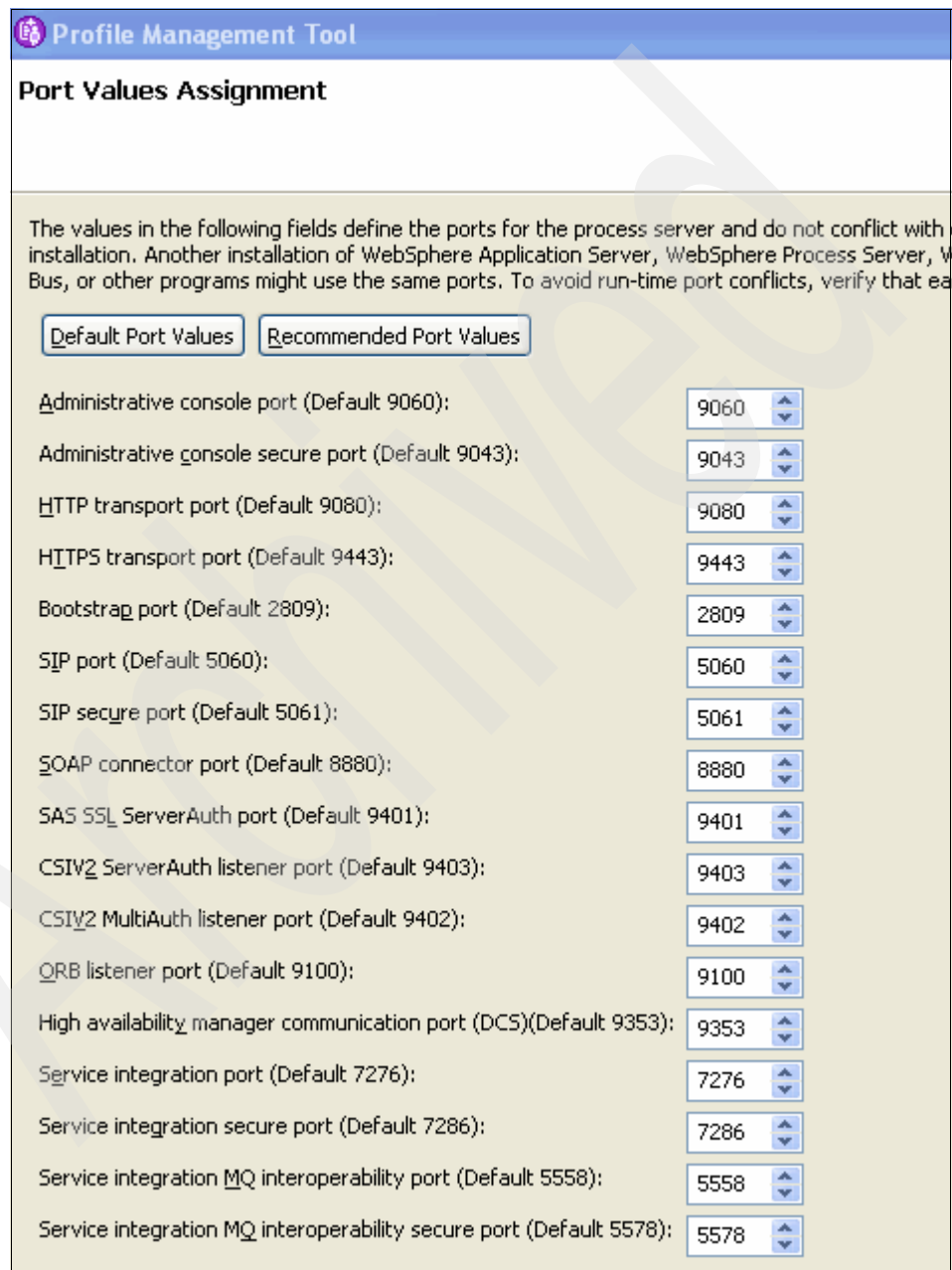
The default ports for z/OS are listed in the following article:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.zseries.doc/info/zseries/ae/rrun_ports.html

Selecting ports using the Profile Management Tool

When you create a profile using the Profile Management Tool, a set of ports is selected for you automatically. These ports are unique to the WebSphere installation on your system but is not necessarily unique to the system. Before creating a profile, review the ports in use on the system to ensure that there are no conflicts with the WebSphere process.

If you anticipate a port conflict with another process running on your system, be sure to take the Advanced profile creation configuration path. You are given the opportunity to review or change the ports that are selected (Figure 1-9)



Profile Management Tool

Port Values Assignment

The values in the following fields define the ports for the process server and do not conflict with installation. Another installation of WebSphere Application Server, WebSphere Process Server, WebSphere Business, or other programs might use the same ports. To avoid run-time port conflicts, verify that each port is unique.

Administrative console port (Default 9060):	9060
Administrative console secure port (Default 9043):	9043
HTTP transport port (Default 9080):	9080
HTTPS transport port (Default 9443):	9443
Bootstrap port (Default 2809):	2809
SIP port (Default 5060):	5060
SIP secure port (Default 5061):	5061
SOAP connector port (Default 8880):	8880
SAS SSL ServerAuth port (Default 9401):	9401
CSIV2 ServerAuth listener port (Default 9403):	9403
CSIV2 MultiAuth listener port (Default 9402):	9402
QRB listener port (Default 9100):	9100
High availability manager communication port (DCS)(Default 9353):	9353
Service integration port (Default 7276):	7276
Service integration secure port (Default 7286):	7286
Service integration MQ interoperability port (Default 5558):	5558
Service integration MQ interoperability secure port (Default 5578):	5578

1.5 Planning for the Profile Management Tool

This section discusses the decision points to consider when you create profiles using the Profile Management Tool.

Note: You cannot use the Profile Management Tool to create or augment profiles on 64-bit platforms (with the exception of i5/OS) or on the Linux on System z™ platform.

1.5.1 Profile environment and type

Profiles that are used for WebSphere Process Server and WebSphere ESB are Network Deployment profiles that are augmented to include features for these products. When using the Profile Management Tool to create a profile, you see the panel shown in Figure 1-9.

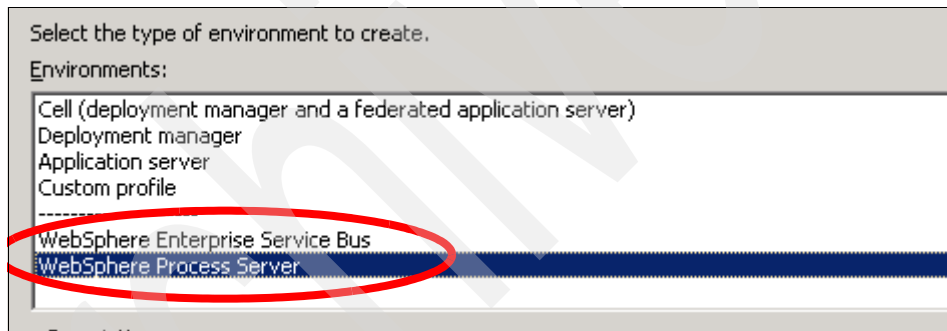


Figure 1-9 Select the profile environment

Selecting one of the first four options creates a Network Deployment profile. To create a profile for WebSphere Process Server or WebSphere ESB, select the appropriate option from the last two options (as shown in Figure 1-9):

- ▶ When you select WebSphere ESB, the profile includes the functions of WebSphere ESB and WebSphere Application Server. Select this environment if you have a licensed WebSphere ESB installation only or if you intentionally want to narrow the scope of the server capabilities.
- ▶ When you select WebSphere Process Server as the environment, the profile includes all the functions of WebSphere Process Server, WebSphere ESB, and WebSphere Application Server.

Next, you select the type of augmented profile that you want to create. The options are a deployment manager profile, a stand-alone application server

profile of the type you selected in the environment selection panel, or a custom node profile. The options that are presented are determined by your previous choice, as follows:

- If you select WebSphere ESB, your options are those shown in Figure 1-10.

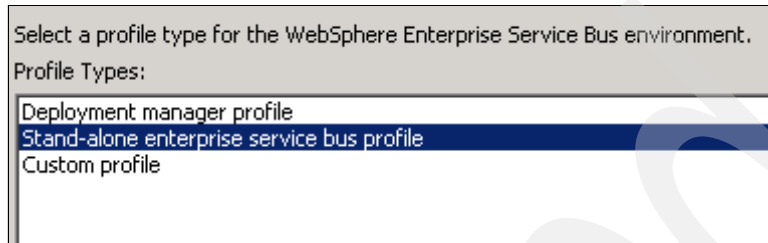


Figure 1-10 Profile Type Selection for WebSphere ESB

- If you select WebSphere Process Server, your options are those shown in Figure 1-11.

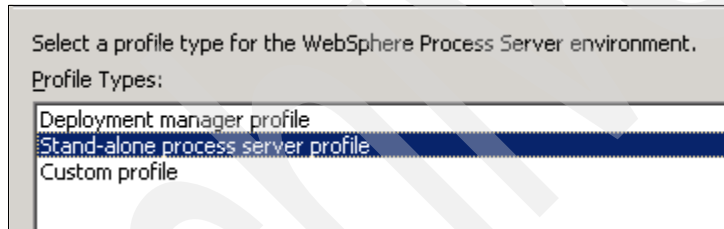


Figure 1-11 Profile Type Selection for WebSphere Process Server

Depending on your selection at this point, you have two or three options for the configuration path as shown in Figure 1-12.

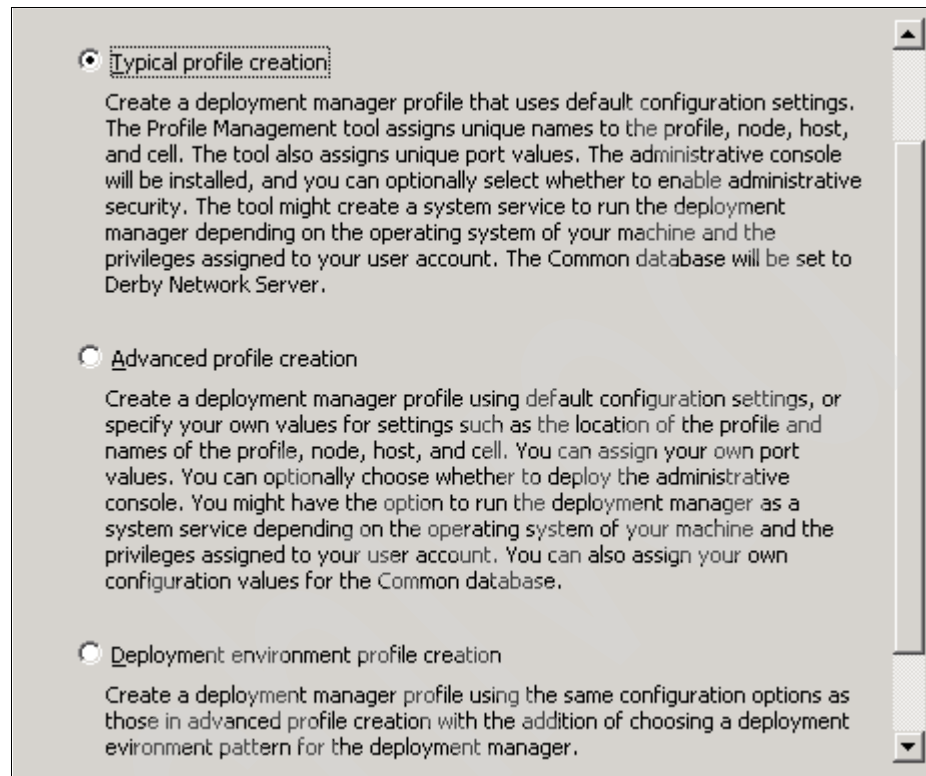


Figure 1-12 Profile creation options

The options are as follow:

- ▶ **Typical profile creation:** This option uses defaults to create the profile. The input requires is kept to a minimum, but you have few options.
- ▶ **Advanced profile creation:** This option gives you more control over settings that are used for profile creation. Default values are provided, as with the Typical option, but you have the option to change them.
- ▶ **Deployment environment profile creation:** This option (not applicable to stand-alone profiles) provides the option of creating a new deployment environment and deployment manager or adding a custom node to an existing deployment environment.

If you choose the deployment environment profile creation option and are creating a deployment manager, you can select from the list of patterns, as shown in Figure 1-13.

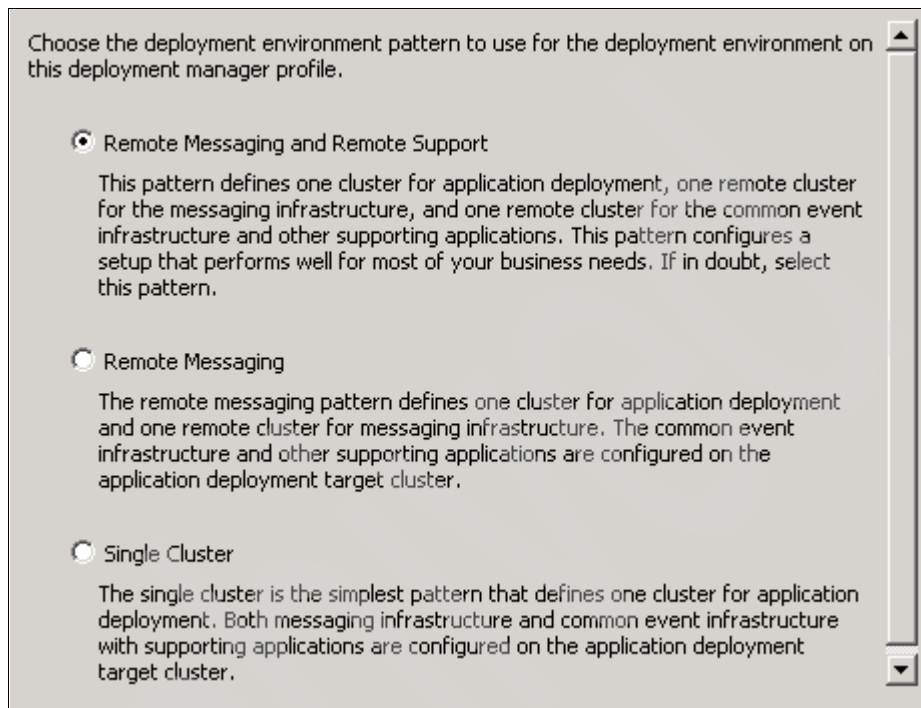


Figure 1-13 Deployment environment patterns

Installing on distributed and i5/OS systems

This chapter shows how to install WebSphere Process Server server on distributed and i5/OS platforms. There are a combination of options that are available to you when you install. This chapter illustrates the installation using one set of options. The intent is to help you to understand the installation process. You can find further examples in Chapter 3, “Creating profiles on distributed and i5/OS systems” on page 53.

For complete planning and installation information, see *Installing and configuring WebSphere Process Server*, which is available at:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/doc/welcome_wps_ins.html

This chapter includes the following topics:

- ▶ Overview of the installation
- ▶ Typical installation
- ▶ Installing on i5/OS
- ▶ Installing the IBM Update Installer
- ▶ Downloading and installing maintenance

2.1 Overview of the installation

You use the following steps to install and configure WebSphere Process Server:

1. Ensure that you have the proper prerequisites by checking the system requirements at:
<http://www-306.ibm.com/software/integration/wps/sysreqs/>
2. Install WebSphere Process Server on each system. The installation process installs WebSphere Application Server Network Deployment and WebSphere Process Server, which includes WebSphere Enterprise Service Bus (WebSphere ESB) functionality.
3. Install the recommended maintenance.
4. Build your topology by creating profiles.

2.2 Typical installation

This section explains the steps to install WebSphere Process Server for a typical installation. You must install WebSphere Process Server on every system that hosts a stand-alone server, deployment manager, or custom node profile.

Note: In the installation examples in this book, we do not create profiles during installation. Instead, we separated the tasks so that installation concepts are not confused with profile concepts. However, if you want to create your first profile during installation, the same concepts for creating profiles exist. For more information, see Chapter 3, “Creating profiles on distributed and i5/OS systems” on page 53.

To install WebSphere Process Server:

1. Start the launchpad, shown in Figure 2-1 on page 41.

The launchpad for WebSphere Process Server includes a menu with options for installing the products that are packaged with WebSphere Process Server. It is located on Disk1 of the installation materials or in the root directory of downloaded installation materials.

For example, we downloaded the product to C:\tmp\C177TML\launchpad.bat.

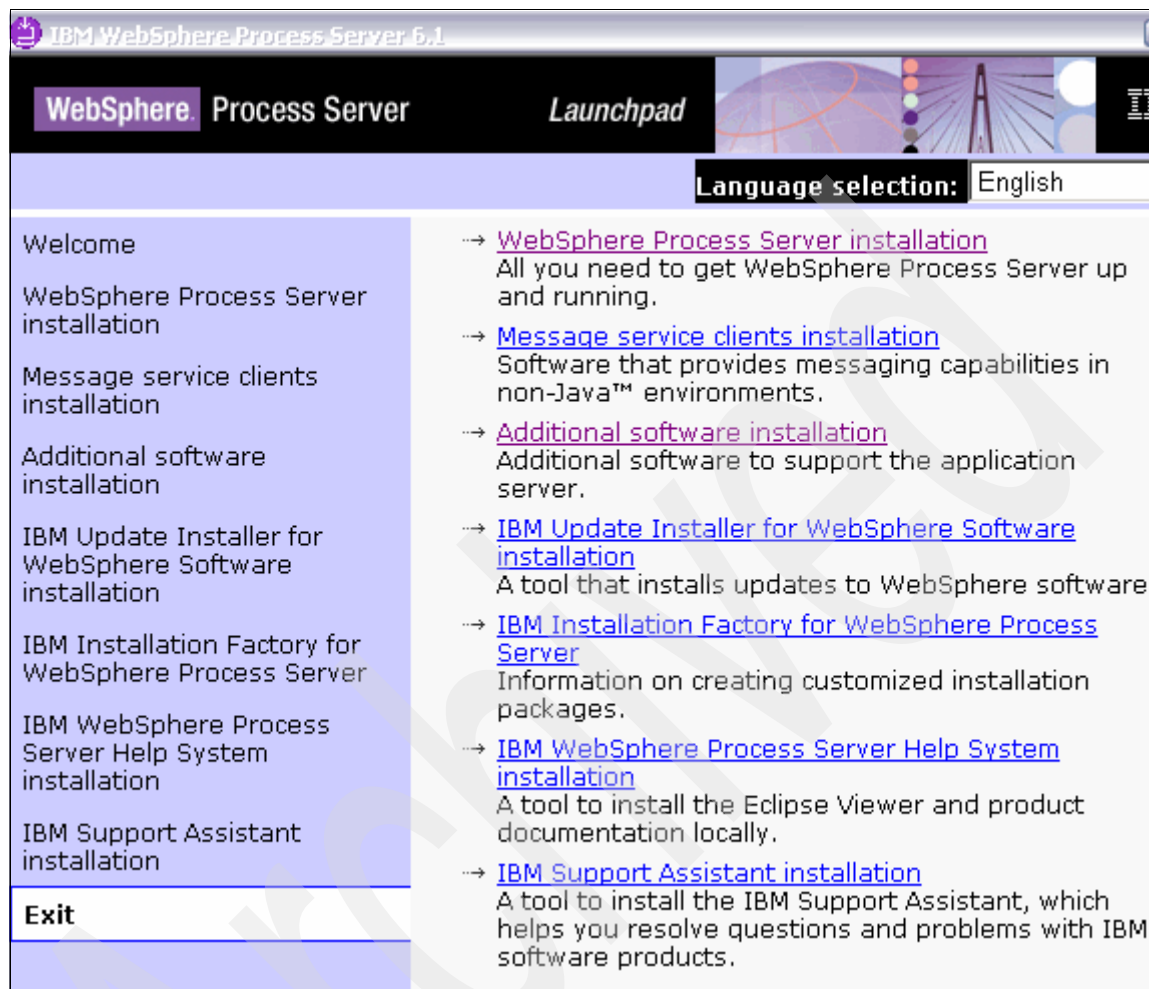


Figure 2-1 Launchpad

2. From the Launchpad, select **WebSphere Process Server installation**.
3. Select **Launch the installation wizard for WebSphere Process Server for Multiplatforms**.
4. On the Welcome page, click **Next**.
5. Read and accept the license agreement. Click **Next**.
6. The wizard checks your operating system to ensure that it meets the requirements for the installation. If your system passes the check, you can proceed. The wizard also checks for the existence of another WebSphere Application Server, WebSphere Process Server, or WebSphere ESB

installation. If one is found, you have the choice of adding features to it, installing a new copy, or creating a new profile. To proceed with the installation, click **Next**.

7. For a new installation, you are given the three options shown in Figure 2-2 for the installation type.

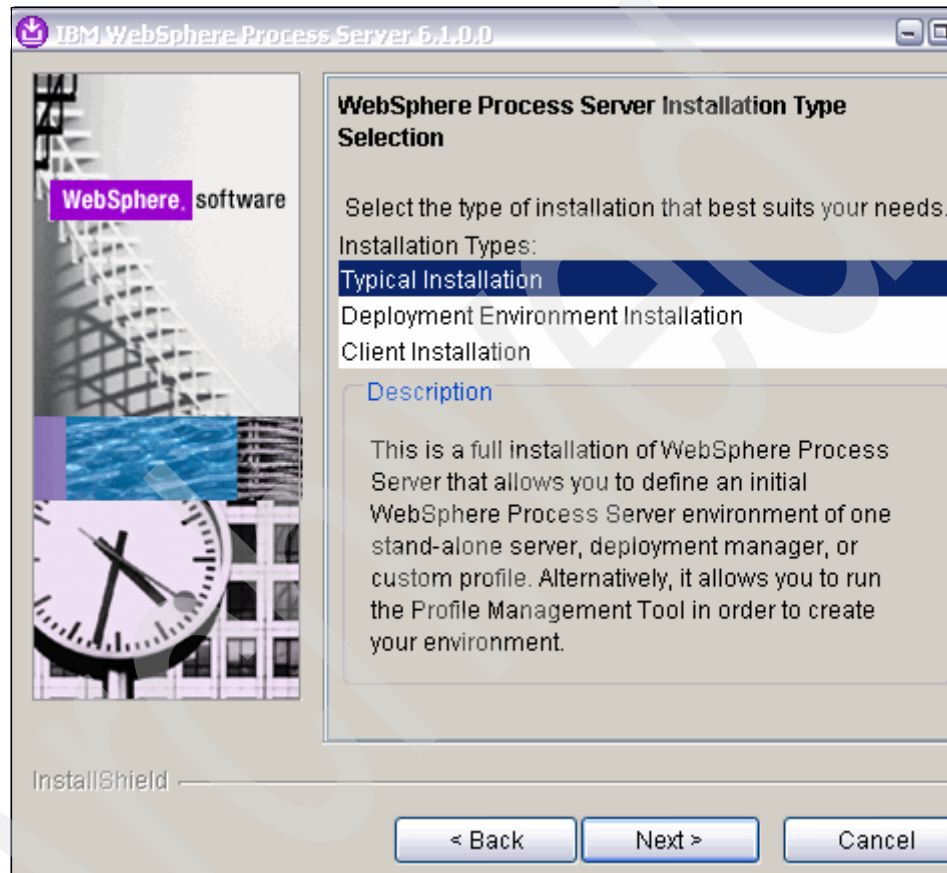


Figure 2-2 Installation type selection

The Deployment Environment Installation option creates a profile and performs the configuration that is required for a selected pattern. In this example, we will delay creating the profile until after installation, so we select **Typical Installation**, which allows you to create a profile or to skip that step.

For more information, see 1.3.4, “Installation type (distributed and i5/OS)” on page 22.

8. The next dialog box allows you to select from the following additional features:

- Business rule beans
- Extended messaging
- Samples

The first two features are deprecated, and you should not use them in new applications. The samples are appropriate for education purposes and are most appropriate in test environments.

For more information, see 1.3.5, “Selecting additional features” on page 22.

Click **Next**.

9. The next dialog box allows you to select the directory that is used as the installation root directory.

On Windows systems, the default directory is:

C:\Program Files\IBM\WebSphere\ProcServer

To avoid any potential issues with file names greater than 256 characters, we suggest that you select something shorter.

From this point forward in this chapter, we refer to the root directory as *WPS_install_root*.

10. You can create profiles as part of the installation process. (For more information about profile types, see “Profile types” on page 3.)

Note: In this example, we assume that the profile will be a WebSphere Process Server capable profile.

Select one of the options shown in Figure 2-3.

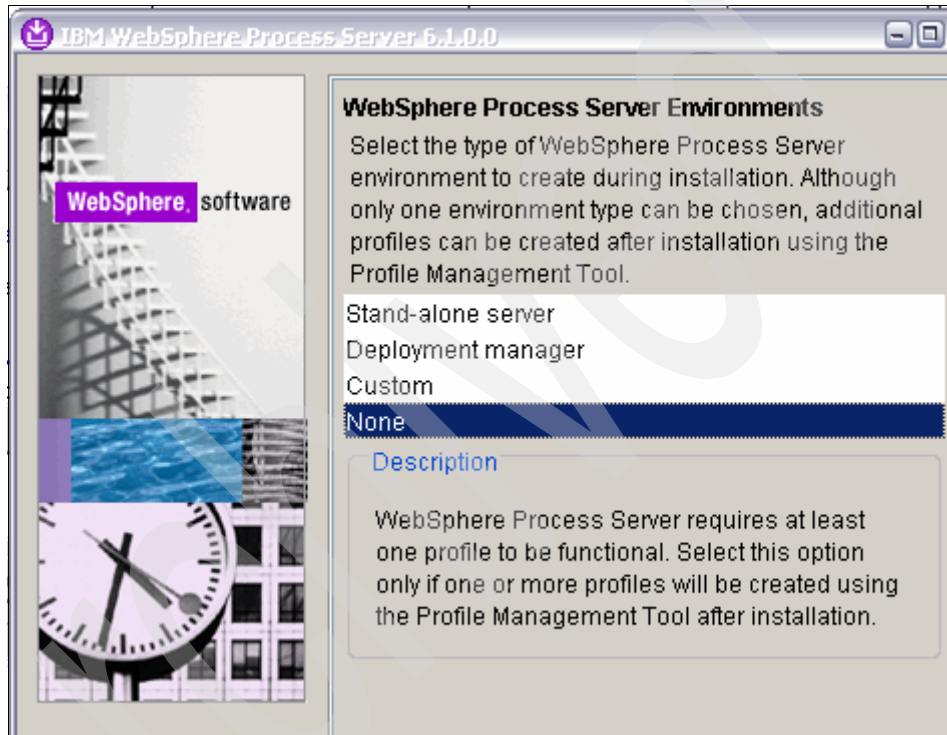


Figure 2-3 Select an environment

If you select a profile type to install, the next series of panels leads you through the options for the profile type.

In this example, we select **None** because we plan to create the profiles after installation.

11. The next dialog box shows a summary of your installation options. Click **Next** to start the installation.
12. When the installation is complete, make sure that it was successful and click **Finish**.

2.3 Installing on i5/OS

Beginning with V6.1, WebSphere Process Server and WebSphere ESB support the System i™ platform. This section illustrates two methods that you can use to install WebSphere Process Server and WebSphere ESB on System i:

- ▶ Installing WebSphere Process Server remotely from a Windows workstation
- ▶ Installing WebSphere Process Server on i5/OS silently

2.3.1 Installing WebSphere Process Server remotely from a Windows workstation

The GUI installation method is the easiest way to install WebSphere Process Server. We recommend that you use this method if you are going to install a simple environment. Do not use it, however, if you have a slow network connection between your server and workstation or if there is a firewall that does not permit access to the hosts servers' ports.

To install WebSphere Process Server V6.1 from a workstation using the graphical installation method, use the following procedure:

1. Start a 5250 emulator session and sign on.
2. Start hosts servers. From a i5/OS command line type:
`STRHOSTSVR SERVER(*ALL)`
Click Enter.
3. Insert the WebSphere Process Server for i5/OS CD into a CD or DVD drive on your Windows workstation. If your workstation has autostart configured, the launchpad starts. If the autostart program does not execute, locate the launchpad.exe file on the CD, and execute it.

Figure 2-4 shows the Launchpad.

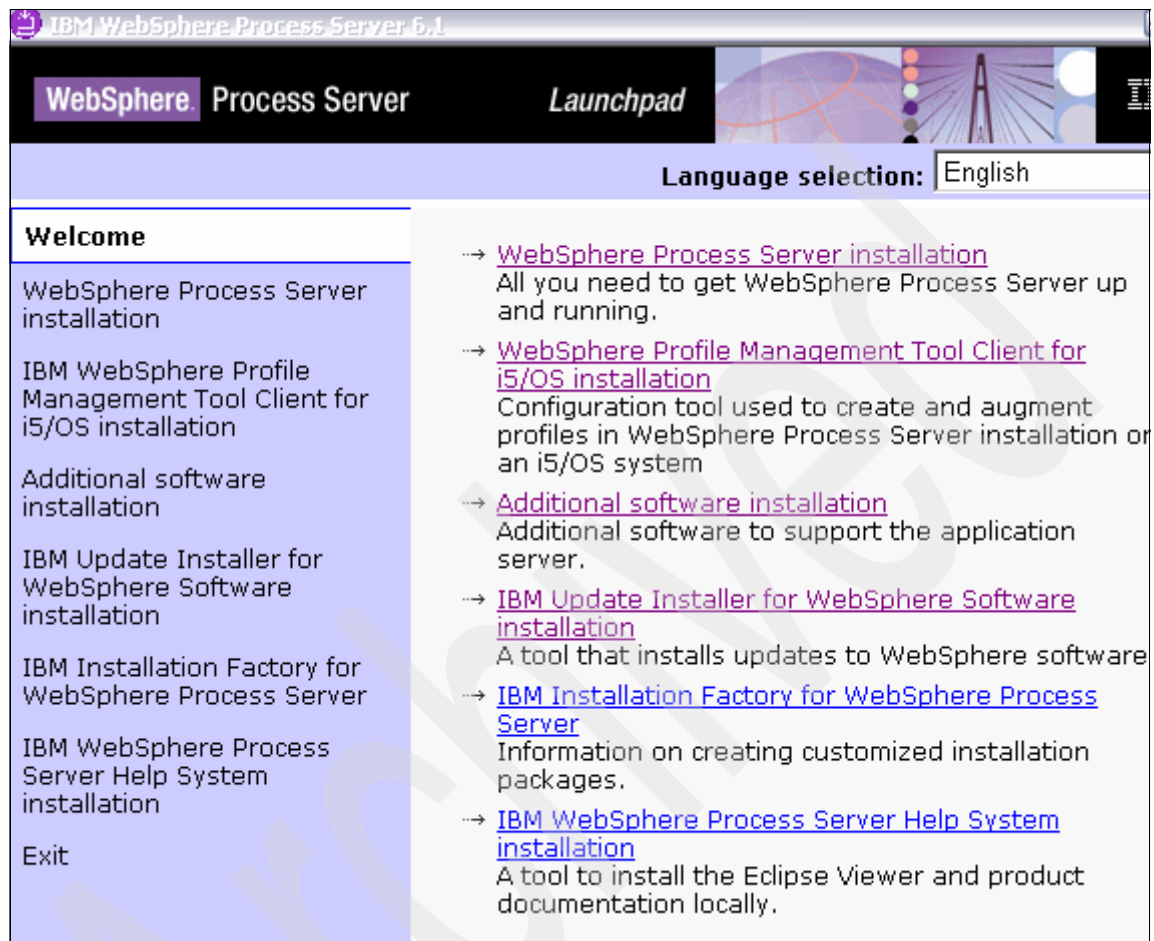
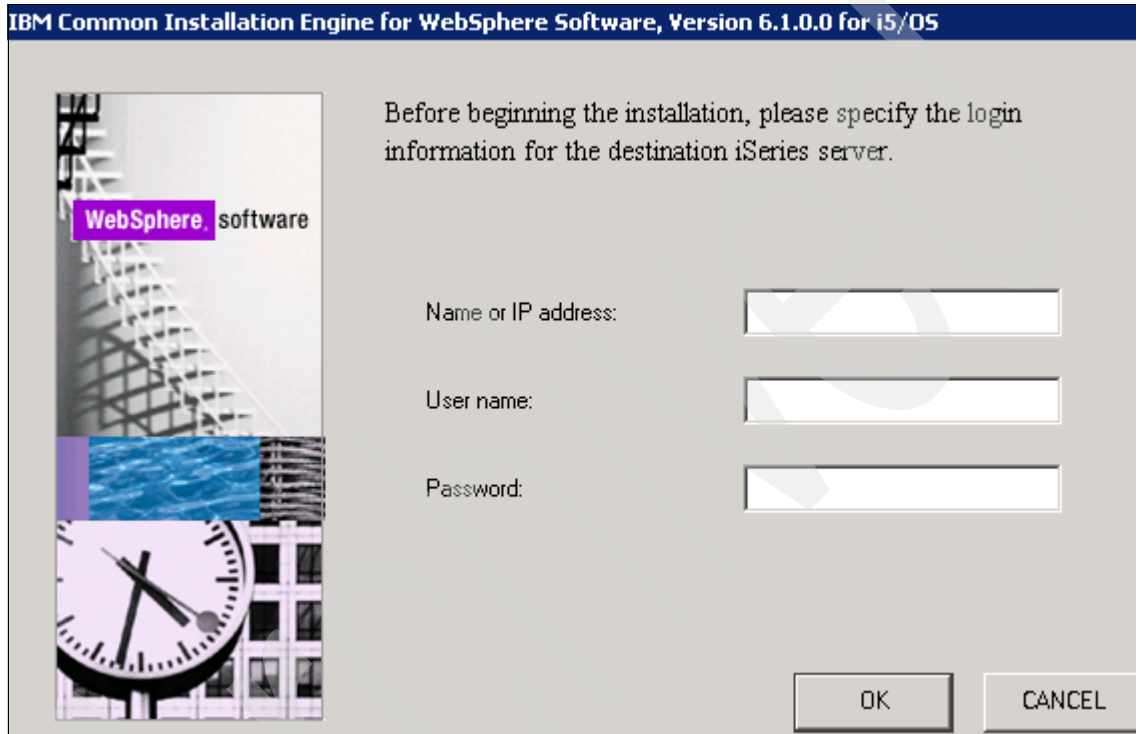


Figure 2-4 Launchpad for WebSphere Process Server for i5/OS platforms

4. Select **WebSphere Process Server Installation** to start the installation wizard for WebSphere Process Server.

5. You are prompted for the login information for the remote i5/OS system, as shown in Figure 2-5.

Note: Make sure that the user ID on the i5/OS system has the *ALLOBJ and *SECADM special authorities.



IBM Common Installation Engine for WebSphere Software, Version 6.1.0.0 for i5/OS

Before beginning the installation, please specify the login information for the destination iSeries server.

Name or IP address:

User name:

Password:

OK CANCEL

Figure 2-5 Log in to the i5/OS system

6. Select an installation type:
 - Typical Installation
 - Deployment Environment Installation
 - Client Installation

In this example, we plan to delay creating the profile until after installation, so select **Typical Installation**. This option allows you to create a profile or to skip that step.

For more information, see 1.3.4, “Installation type (distributed and i5/OS)” on page 22.

7. Specify the installation root directory for WebSphere Process Server as shown in Figure 2-6.

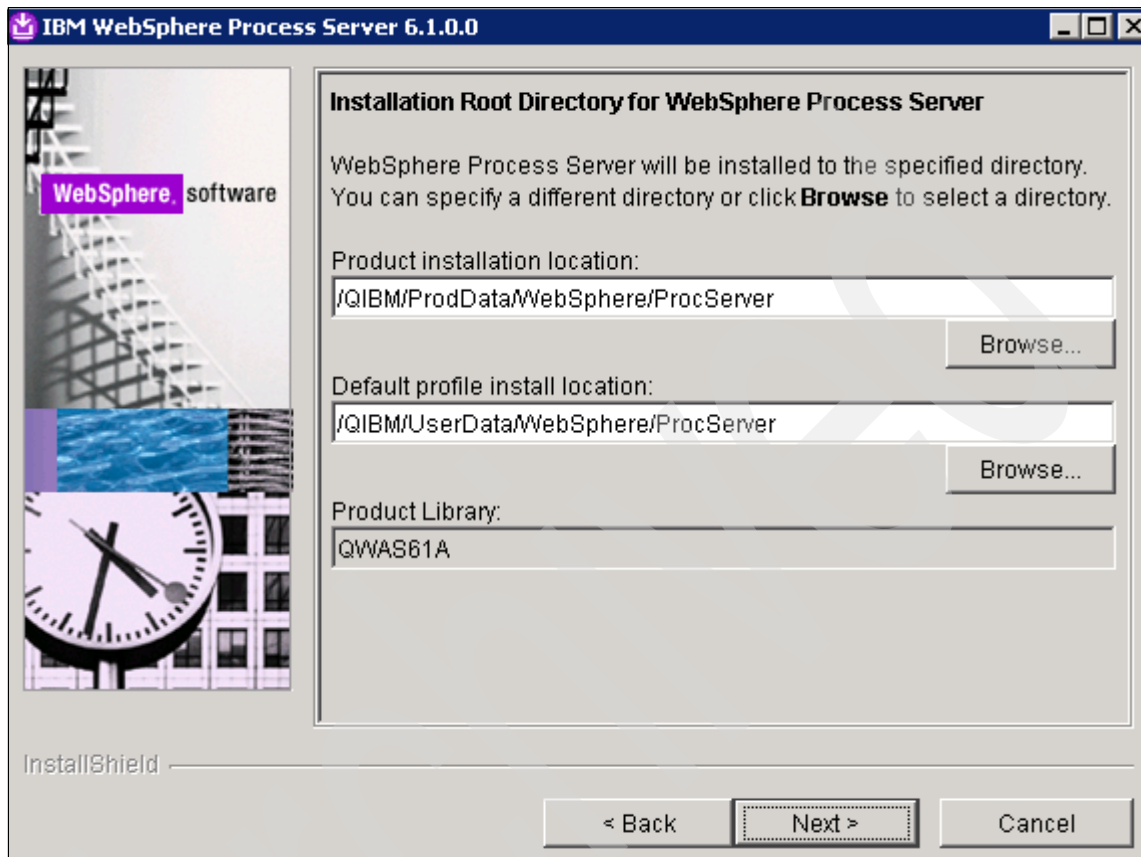


Figure 2-6 Select the installation root directory

8. You can create profiles as part of the installation process. (We discuss profile types in “Profile types” on page 3.) Select one of the following options:
 - Stand-alone server
 - Deployment manager
 - Custom
 - None

In this example, we select **None** because we will build the profiles after installation.

9. The next dialog box shows a summary of your installation options. Click **Next** to start the installation.

10. When the installation completes, make sure that it was successful and click **Finish**.

2.3.2 Installing WebSphere Process Server on i5/OS silently

The following process explains how to install WebSphere Process Server on an i5/OS system in silent mode:

1. Sign on to the i5/OS server with a user profile that has *ALLOBJ and *SECADM special authorities.
2. Edit the responsefile.wbis.txt file, which is located in `$BUILD_HOME/WBI`, where `$BUILD_HOME` is the directory where you decompressed the WebSphere Process Server installation image. Usually, you need to change the following values:
 - Change the value for `-OPT silentInstallLicenseAcceptance` from `false` to `true`. A value of `true` indicates that you have read and accept the terms of the license agreement. This change is required to run the installation.
 - Change the value `-OPT wpsInstallType` according to the installation type you want. Use one of the following values:
 - **Typical:** A full installation. It allows you to define an initial environment that consists of a stand-alone process server, a deployment manager, or a custom node.
 - **Client:** A partial installation. It allows you to run client applications that interact with a WebSphere Process Server within the same cell.
 - **ndGuided:** A full installation that guides you through setting up a Network Deployment topology. You can create a new topology based on a topology pattern or continue to define a topology that you have created already.

Note: You cannot select the `ndGuided` installation if you are installing using an existing base WebSphere Application Server V6.1.

In this example, we select **Typical**.

- Change the value for `-OPT useExistingWAS` from `true` to `false`, indicating that you are installing a new WebSphere Application server. If you intend to use an existing installation of WebSphere Application Server, then keep the value as `true`.
- Uncomment the value for `-OPT defaultProfileLocation`.

3. On the Command entry of System i, issue **qsh** to start the Qshell command shell.
4. Then issue the **INSTALL** command from the `$BUILD_HOME\WBI` directory to start the installation program as follows:

```
INSTALL -options $BUILD_HOME/WBI/responsefile.wbis.txt -silent
```

2.4 Installing the IBM Update Installer

You install maintenance for products based on WebSphere Application Server using the IBM Update Installer for WebSphere Software. Follow these steps to install the IBM Update Installer:

1. From the Launchpad, select **IBM Update Installer for WebSphere Software installation**.
2. Select **Launch the installation wizard for IBM Update Installer**.
3. Click **Next** on the Welcome page.
4. Read and accept the software license agreement. Click **Next**.
5. When the prerequisite check completes successfully, click **Next**.
6. Select the root directory for the installation. For example, on our system, we installed WebSphere Process Server in the following directory:

C:\WebSphere\ProcServer

So we selected the following directory for the Update Installer installation root directory:

C:\WebSphere\UpdateInstaller

Click **Next**.

7. Review your settings, and click **Next**.
8. When the installation completes successfully, click **Finish**.

2.5 Downloading and installing maintenance

The next step is to install any maintenance that is required. For the latest information, see Recommended Fixes for WebSphere Process Server, which is available at:

<http://www-1.ibm.com/support/docview.wss?rs=2307&uid=swg27006649>

To install any maintenance follow these steps:

1. Download the maintenance to a temporary location.
2. Decompress the maintenance to the *updater_root*/maintenance directory, for example:

C:\WebSphere\UpdateInstaller\maintenance

3. Start the Update Installer using the ***updater_root/update*** command.
4. Follow the instructions in the Update Installer wizard to select and install the maintenance.

For more information about using the Update Installer, see *Installing fix packs and refresh packs with the Update Installer* at:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/doc/tins_apply_service.html

Creating profiles on distributed and i5/OS systems

This chapter includes examples of how to create a variety of topologies, including those defined by the deployment environment patterns. While all the topologies that we show in this chapter are applicable to distributed, i5/OS, and z/OS systems, the methods that we use to create them are specific to distributed and i5/OS. For an example of building a z/OS topology, see Chapter 4, “Creating and managing profiles on z/OS” on page 97.

This chapter includes the following topics:

- ▶ Methods used to build the profiles
- ▶ Stand-alone WebSphere Process Server
- ▶ Building a deployment environment using a pattern
- ▶ Augmenting a WebSphere ESB profile to WebSphere Process Server

3.1 Methods used to build the profiles

The two methods that we use to build profiles in the topology examples in this chapter are the Profile Management Tool and the profile creation options during installation. A third option, using the **manageprofiles** command, exists for creating profiles on distributed and i5/OS systems. We show an example of using this command but do not use it to build the example topologies.

On z/OS systems, you create profiles by generating a series of jobs using the z/OS Profile Management Tool (zPMT) or ISPF customization dialog box and by running each job. You can find an example of creating profiles in z/OS in Chapter 4, “Creating and managing profiles on z/OS” on page 97.

3.1.1 Profile Management Tool

The Profile Management Tool can run on i5/OS and distributed platforms.

Note: You cannot use the Profile Management Tool to create or to augment profiles on 64-bit platforms (except for i5/OS) or on the Linux on System z platform. To create profiles on these platforms, you must use the **manageprofiles** command.

Start the Profile Management Tool using one of the following methods:

- ▶ From the *First steps* console. This console starts at the end of an installation if you select the option to start the Profile Management Tool.
- ▶ (Windows) From the Start menu, select **Start → All Programs → IBM WebSphere → Process Server 6.1 → Profile Management Tool**.
- ▶ (UNIX, Linux) Run `WPS_install_root/bin/ProfileManagement/pmt.sh`.
- ▶ (Windows) Run `WPS_install_root/bin/ProfileManagement/pmt.bat`.
- ▶ (i5/OS) Start the Profile Management Tool client on a Windows system.

Profile Management Tool Client for i5/OS

The Profile Management Tool client for i5/OS is installed from the WebSphere Process Server V6.1 installation launchpad. You need to move the installation files to the Windows server, start the Launchpad, and then select the WebSphere Profile Management Tool Client for i5 installation. Follow these steps:

1. Start the Profile Management Tool client on a Windows system using one of the following methods:
 - From the Start menu, select **Start → All Programs → IBM WebSphere → Profile Management Tool Client for i5/OS**.
 - Run `PMT_client_install\PMTP\pmt.bat`.
2. Enter the host name, user ID, and password that are required to log in to the i5/OS system (Figure 3-1).

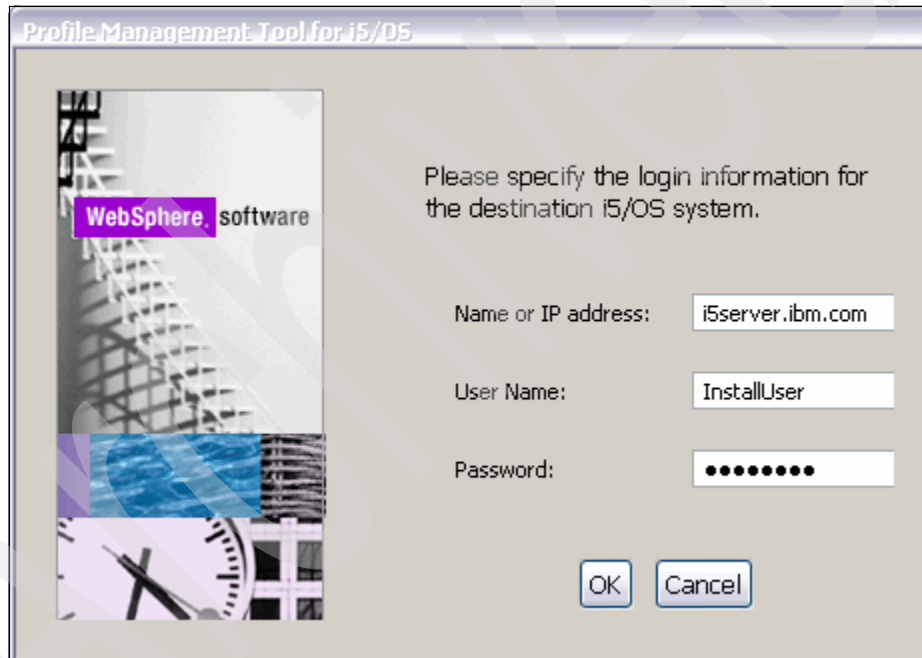


Figure 3-1 Log in to the i5/OS system

3. Select the WebSphere Process Server installation location with which you want to work, and click **Launch Profile Management Tool** (see Figure 3-2 on page 56).

The default port number is 1099. If that port number is busy, you will receive an error message, in which case, you can select a different port number to continue.

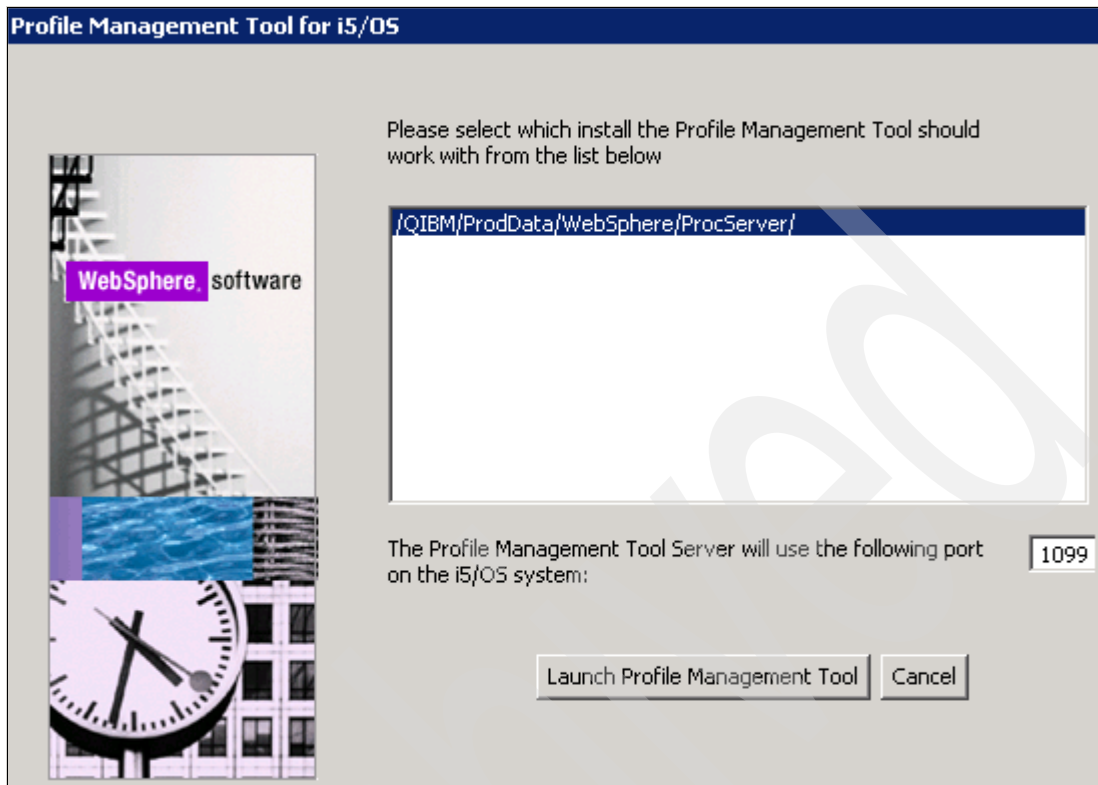


Figure 3-2 Select the installation

3.1.2 The manageprofiles command

You can create profiles on i5/OS and distributed platforms using the **manageprofiles** command, which is located in the *WPS_install_root/bin* directory. The Profile Management Tool is the GUI interface to this command for creating and augmenting profiles, but the **manageprofiles** command has additional function that is not exposed by the Profile Management Tool (for example, listing and deleting profiles).

For information about using the **manageprofiles** command and its parameters, see:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/doc/rins_manageprofiles.html

The type of profile that you create depends on the template that you select with the **-templatePath** parameter, as follows:

- ▶ **default.wbiserver**: WebSphere Process Server stand-alone server profile
- ▶ **dmgr.wbiserver**: WebSphere Process Server deployment manager profile
- ▶ **managed.wbiserver**: WebSphere Process Server custom profile
- ▶ **default.esbserver**: WebSphere Enterprise Service Bus (WebSphere ESB) stand-alone server profile
- ▶ **dmgr.esbserver**: WebSphere ESB deployment manager profile
- ▶ **managed.esbserver**: WebSphere ESB custom profile

Example 3-1 provides an example of using the **manageprofiles** command on an i5/OS system to create a stand-alone WebSphere Process Server profile. You issue the command from Qshell command shell.

Note: We formatted the example so that each parameter is on a separate line for readability. When you enter the command, it is entered as a continuous line.

Example 3-1 Example of using manageprofiles

```
/qibm/proddata/websphere/appserver/v61/nd/manageprofiles -create
-templatePath
/QIBM/ProdData/WebSphere/AppServer/V61/ND/profileTemplates/default.wbis
erver -ceiOutputScriptDir
/QIBM/UserData/WebSphere/AppServer/V61/ND/profiles/ProcSrv01/dbscripts/
CEI_LOCAL
-dbPassword ${password} -dbHostName ${hostname}
-profilePath /QIBM/UserData/WebSphere/AppServer/V61/ND/profiles/pat
-nodeName ${nodename}
-enableAdminSecurity false
-dbSchemaName ${commondb}
-dbName *SYSBAS
-dbCreateNew true
-ndtopology false
-dbUserId ${username}
-dbJDBCClasspath /QIBM/ProdData/HTTP/Public/jt400/lib
-ceiNativeJdbcClassPath /QIBM/ProdData/Java400/ext
-ceiDbPassword ${password} -omitAction samplesInstallAndConfig
-ceiDbExecuteScripts true
-dbCommonForME false
-dbType DB2UDBISERIES_TOOLBOX
-configureBRM false
```

```
-profileName    pat
-hostName ${hostname}
-ceiDbName     *LOCAL
-ceiDbUser    ${username}
-configureBPC  false
-cellName     WBIRTPCell
-ceiDbProduct  CEI_DB_ISERIES_NATIVE
-dbDelayConfig false -ceiCollection ${ceidb}
-bpcmqPassword ${password} -bpcmqUser ${username}
-configureBPC true
-fileStoreForME true
```

3.2 Stand-alone WebSphere Process Server

To create a WebSphere Process Server stand-alone server, follow these steps:

1. Start the Profile Management Tool.
2. Select **Create**.
3. Click **Next** on the Welcome page.
4. Select **WebSphere Process Server** as the environment selection, and click **Next** as shown in Figure 3-3.

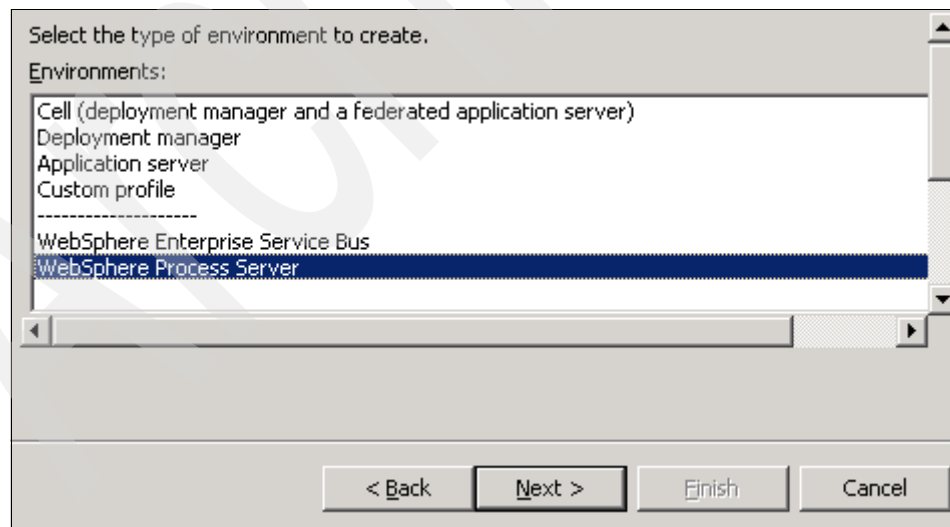


Figure 3-3 Profile environment selection

5. Select **Stand-alone process server profile** and click **Next** as shown in Figure 3-4.

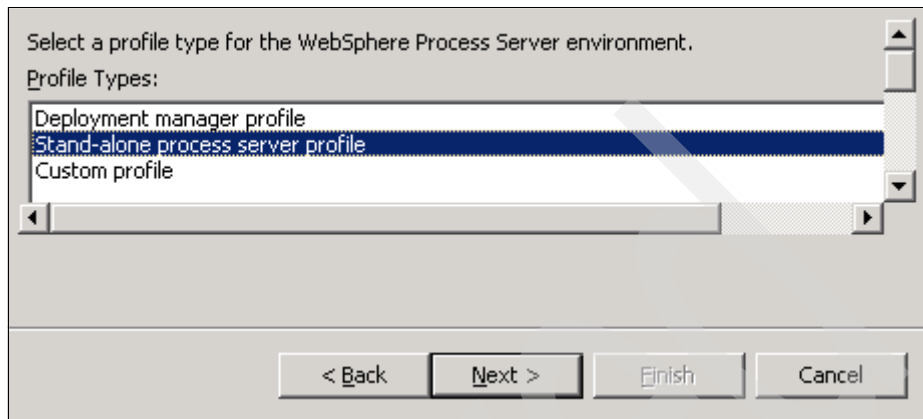


Figure 3-4 Profile type selection

6. The next dialog box allows you to select one of two paths for the remainder of the profile creation process: Typical or Advanced.
Take the **Advanced** path and click **Next**.

7. Select **Deploy the administrative console** and **Deploy the default application**, and click **Next**.

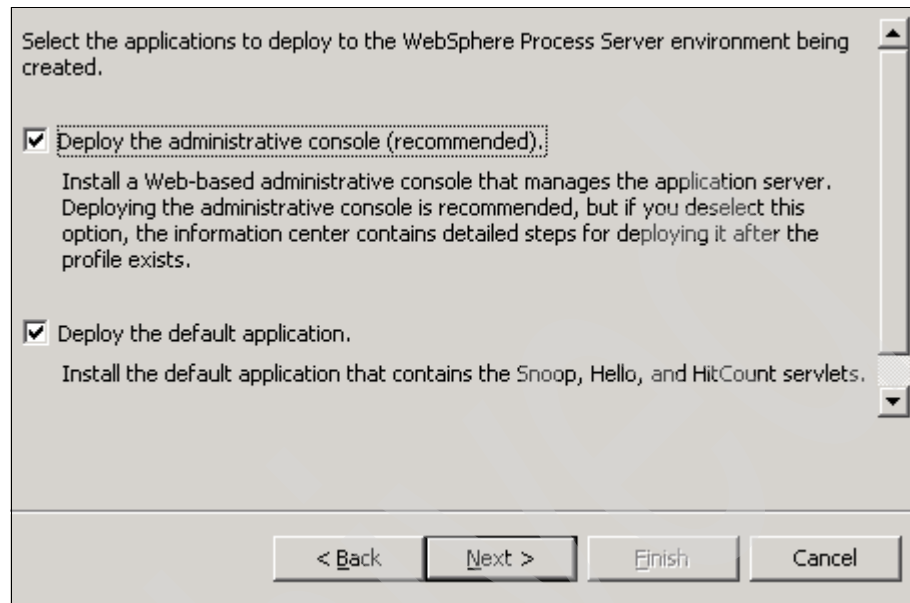


Figure 3-5 Selection optional applications

8. Enter a unique name for the profile or accept the default. It is helpful to use the same name for the profile and the directory location. From this point forward in this chapter, we refer to this directory as *profile_root*. There are two options (shown in Figure 3-6 on page 61):
- **Make this profile the default:** Select this option to make this profile the new default profile. Commands that run without referring to a specific profile (**startServer**, **stopServer**, and so forth) are run against the default profile.
 - **Create the server using the development template:** If this is a test environment, consider using this option. Not only does it reduce startup time, it also populates the user repository with the same sample organization users and groups that are defined automatically in the integrated test environment.

Click **Next**.

Specify a profile name and directory path to contain the files for the run-time environment, such as commands, configuration files, and log files. Click **Browse** to select a different directory.

Profile name:

Profile directory:

☐ **Create the server using the development template.**
Select this option to create a server using configuration settings optimized for development. The development template reduces startup time and allows the server to run on less powerful hardware. Do not use this option for production servers.

☐ **Make this profile the default.**
Each installation of WebSphere Process Server always has one default profile. Commands that run without referring to a specific profile use the default profile. Select this option to make this profile the new default.

Important: Deleting the directory a profile is in does not completely delete the profile. Use the **manageprofiles** command to completely delete a profile.

Figure 3-6 Profile options

9. Enter a node name, the system host name, and a cell name (see Figure 3-7). The node name and cell name defaults based on the host name of your system.

The names that you select for the node and cell should be meaningful and unique within the installation.

Click **Next**.

Specify a node name, a host name, and a cell name for this profile.

Node name:
WPSNode1

Host name:
WPS1.ibm.com

Cell name:
WPSCell1

Node name: A node name is used for administration. If the node is federated, the name must be unique within the cell.

Host name: A host name is the domain name system (DNS) name (short or long) or the IP address of this computer.

Cell name: A cell name is a logical name for the group of nodes administered by this deployment manager.

For more information about profile naming and augmentation considerations, see the online information center.

[Online information center link](#)

< Back Next > Finish Cancel

Figure 3-7 Select node, cell, and host

10. Choose whether to enable administrative security. If you enable security here, you are asked for a user ID and password that is added to a file-base user registry with the Administrative role. We discuss administrative security in “Administrative security” on page 31.

In this example, we enable administrative security.

Click **Next**.

11. The wizard presents a list of TCP/IP ports for use by the application server. If you already have existing profiles on the system (within this installation), the installation program takes this into account when the wizard selects the port assignments. However, you need to verify that these ports are unique on the system.

Click **Next**.

12. *Windows only*: On Windows systems, you have the option of running the application server as a service, which provides a simple way of starting the application server automatically when the system starts. If you are installing on a Windows system, the next dialog box provides the option of running the server process as a Windows service. If you elect to do this, enter the values for the logon and startup type.

In this example, we specify that the server runs as a Windows service and starts automatically.

Click **Next**.

13. The wizard allows you to create an optional Web server definition. Web server definitions define an external Web server to the cell configuration, which allows you to manage Web server plug-in configuration files for the Web server and, in some cases, to manage the Web server. If you have not installed a Web server or want to do this later, you can do this easily from the administrative console.

In this example, we do not define an external Web server.

Click **Next**.

14. Select a database product and a name for the Common Event Infrastructure database. The database you choose here is created for you.

Alternatively, you can elect to generate the database scripts and execute them manually later.

In this example, we are creating a test server on a Windows system and elect to use Derby, as shown in Figure 3-8 on page 64.

Click **Next**.

Configure the Common Event Infrastructure database.

Choose a database product:

Derby Embedded

☐ Override the destination directory for generated scripts.

Database script output directory:

C:\WebSphere\ProcServer\profiles\SAPProcSrv\dbscripts\CEI_event

Browse...

Database name:

event

☐ Delay execution of database scripts.

☐ Override data source.

< Back Next > Finish Cancel

Figure 3-8 Select a database provider for the Event database

15. Depending on your selection, the next dialog box might ask for information that is required to access the database, such as the user ID with authority to operate database. Enter the required information, and click **Next**.
16. The next step gives you the option to create a sample Business Process Choreographer. Because this is not a production server, we select this option, as shown in Figure 3-9 on page 65.
Click **Next**.

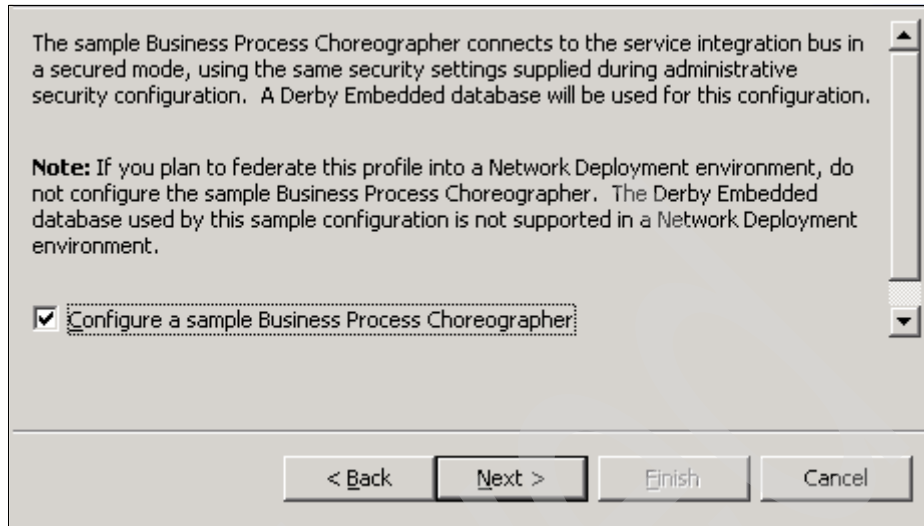


Figure 3-9 Configure a sample Business Process Choreographer

17. The next step allows you to have the business rules manager configured. Our application uses rules, so we select this option (Figure 3-10).

Click **Next**.

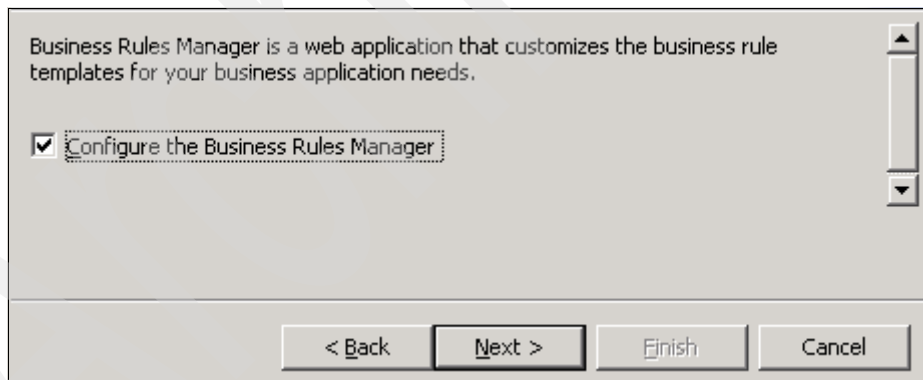


Figure 3-10 Configure the business rules manager

18. The next step is to configure a common database for WebSphere Process Server. Again, we take the Derby Embedded database (as shown in Figure 3-11) because this is a stand-alone server used for testing. Click **Next**.

Various components use WebSphere Process Server common database. Choose a database and enter the information based on that type.

Choose a database product:

Derby Embedded

☐ Override the destination directory for generated scripts.

Database script output directory:

C:\WebSphere\ProcServer\profiles\SAProcSrv\dbscripts\CommonDB\Derby\WP

☒ Create a new local database.

☐ Use an existing database.

Database name:

WPRCSDB

☐ Delay execution of database scripts for new or existing database.

☐ Use a file store for Messaging Engines (MEs).

☐ Use this database for Messaging Engines (MEs).

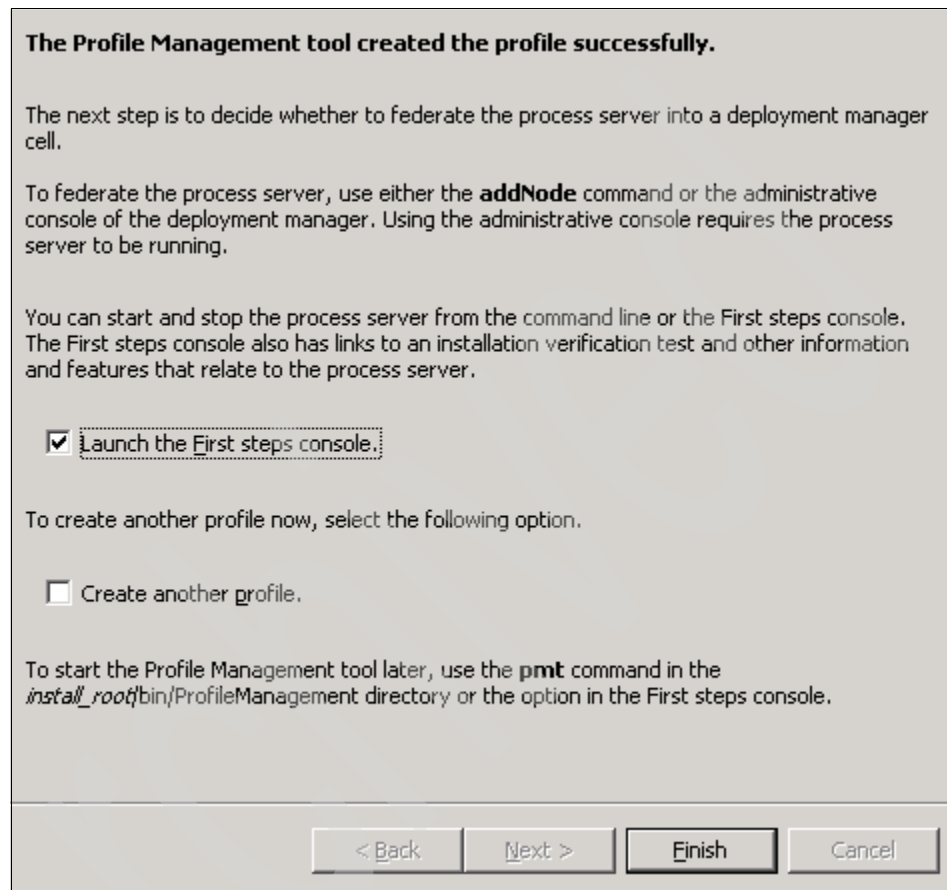
< Back Next > Finish Cancel

Figure 3-11 Common database

Messaging engines can use a data store or file store for persistent messages. By default, data stores are used; however, using the file store might improve performance. This option is only valid in a stand-alone server configuration.

19. Review the options that you selected. Click **Create** to create the new stand-alone server.

20. The last panel shows the status of the profile creation. Ensure that the profile is created successfully. Then, select the **Launch the First Steps console** option (as shown in Figure 3-12) and click **Finish**.



The Profile Management tool created the profile successfully.

The next step is to decide whether to federate the process server into a deployment manager cell.

To federate the process server, use either the **addNode** command or the administrative console of the deployment manager. Using the administrative console requires the process server to be running.

You can start and stop the process server from the command line or the First steps console. The First steps console also has links to an installation verification test and other information and features that relate to the process server.

☒ Launch the First steps console

To create another profile now, select the following option.

☐ Create another profile.

To start the Profile Management tool later, use the **pmt** command in the *install_root/bin/ProfileManagement* directory or the option in the First steps console.

< Back Next > **Finish** Cancel

Figure 3-12 Success

21. The First Steps console provides options that you can use to verify the installation, start and stop the server, and access the administrative console for the server. See Figure 3-13.

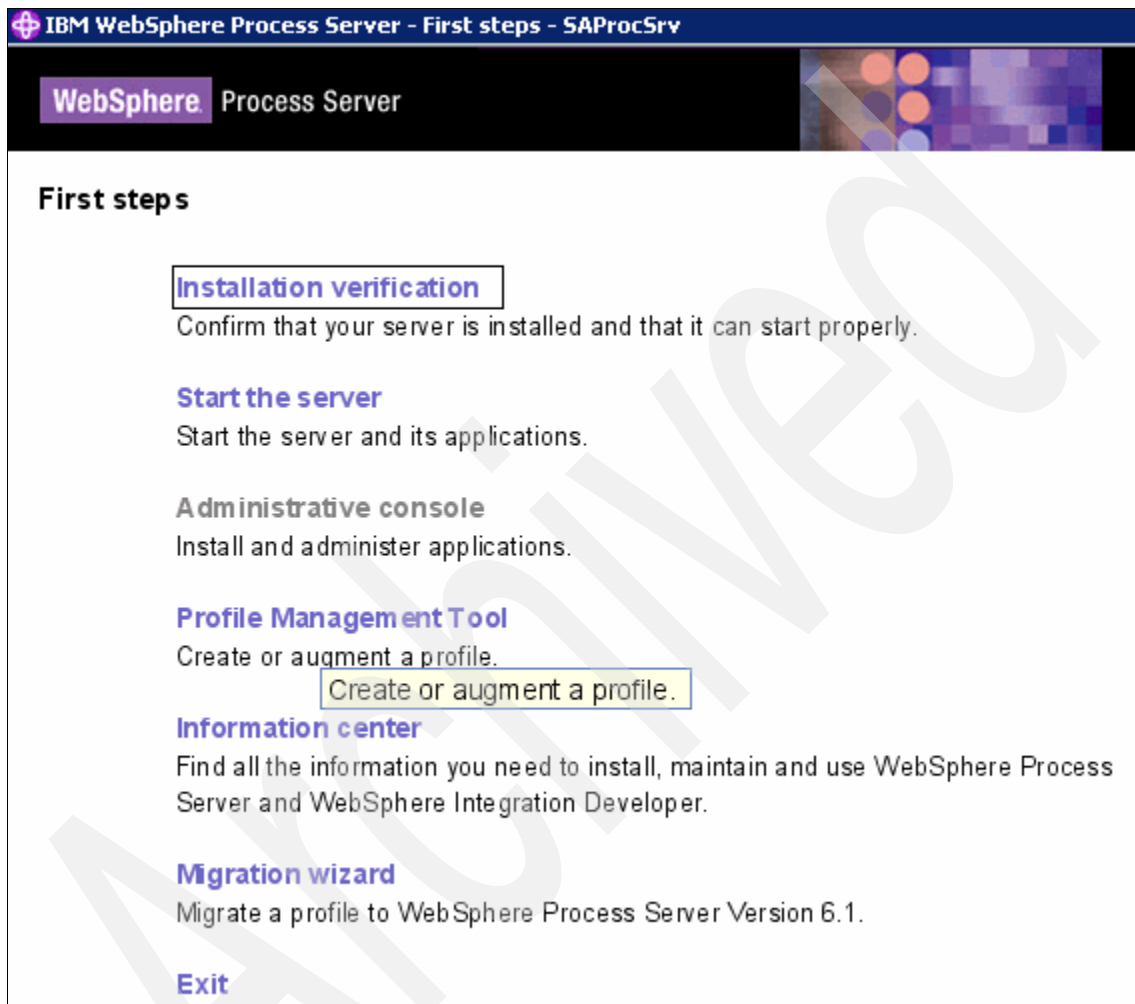


Figure 3-13 First Steps console for a stand-alone server

Creating a WebSphere ESB stand-alone server: If you select WebSphere ESB as the environment (see Figure 3-3 on page 58), the process is basically the same, except that you do not see the options to create the sample Business Process Choreographer and to configure the business rules manager.

3.3 Building a deployment environment using a pattern

This example shows how to create a deployment environment that consists of two nodes. The process is the same, regardless of the pattern that you choose. You can take this process and extend it easily to include additional nodes.

The overall topology that we build in this section has the following characteristics:

- ▶ A Web server performs load balancing among the clusters that host the applications.
- ▶ Each deployment environment has one cluster that hosts the applications. Depending on the pattern that you choose, additional clusters exist to host the messaging and support functions.
- ▶ A single common DB2 database, WPRCSDB, is used to host all the required tables for the WebSphere Process Server and WebSphere ESB functions.
- ▶ The cluster that hosts the business integration modules consists of WebSphere Process Server capable nodes and servers.
- ▶ The cluster that hosts the mediation modules consists of WebSphere Process Server capable nodes and WebSphere ESB capable servers.

Figure 3-14 shows the topology that uses the single cluster pattern to build both the WebSphere Process Server and WebSphere ESB clusters.

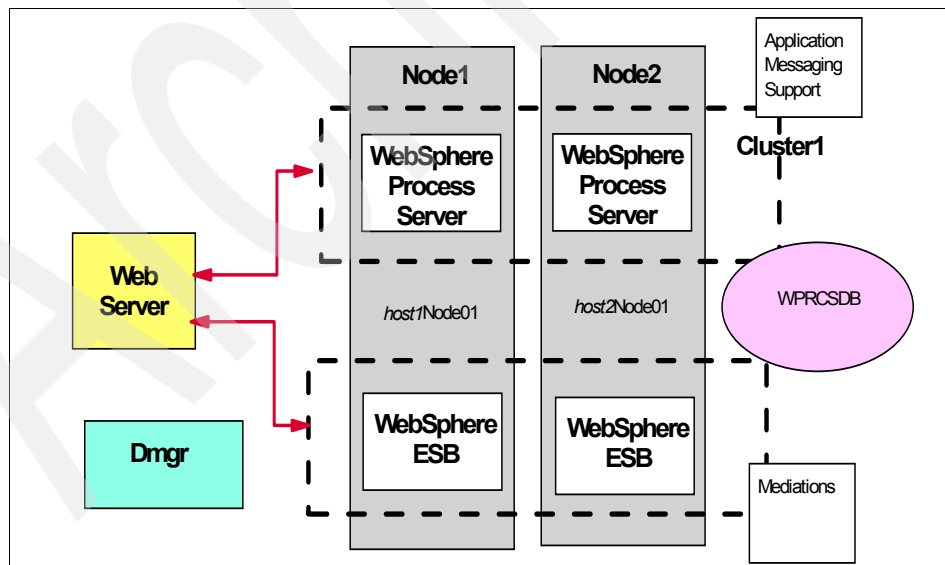


Figure 3-14 Single cluster for processes and single cluster for mediations

Figure 3-14 shows the topology that uses the remote messaging pattern to build the WebSphere Process Server clusters and the single cluster pattern to build the WebSphere ESB cluster.

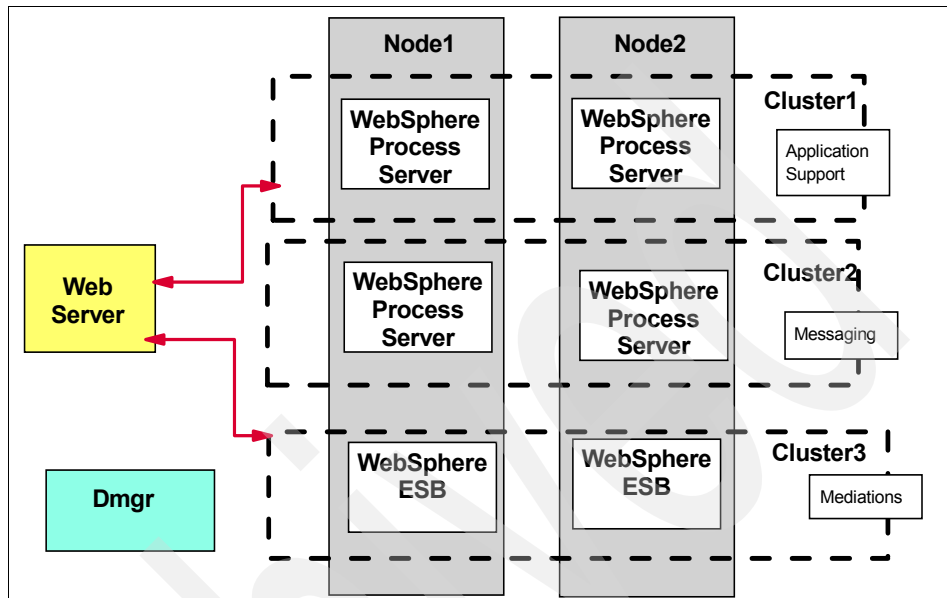


Figure 3-15 Remote messaging pattern

Figure 3-16 shows the topology that uses the remote messaging and remote support pattern to build the WebSphere Process Server clusters and the single cluster pattern to build the WebSphere ESB cluster.

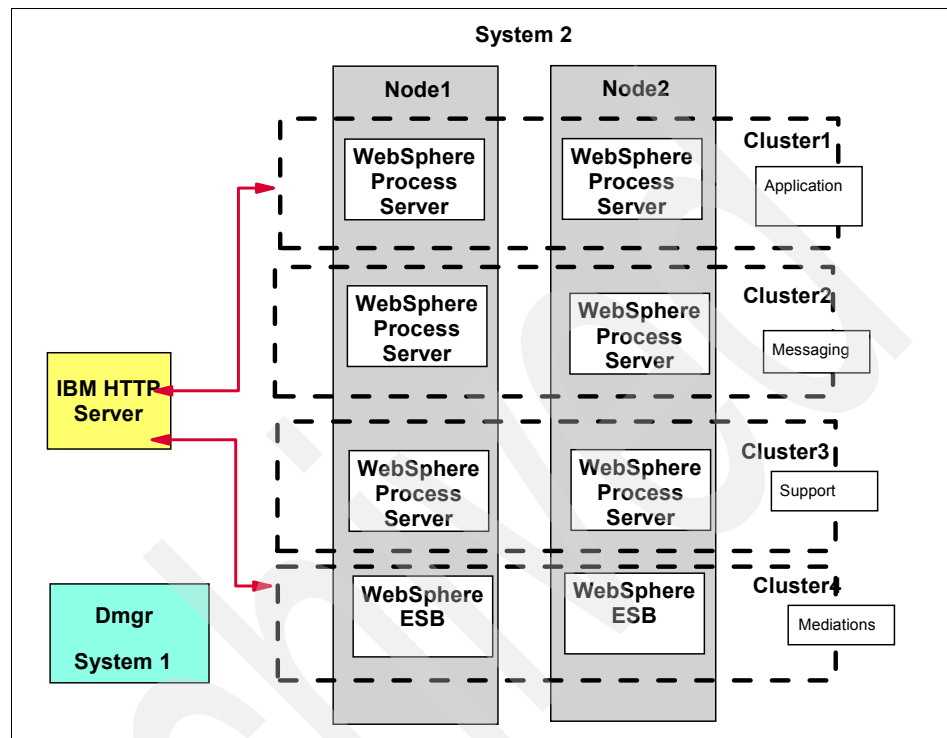


Figure 3-16 Remote messaging and remote support

3.3.1 Summary of steps

Note: The following steps create a deployment environment with one database. For a production environment, it is recommended that you use multiple databases. See 1.4.1, “Planning for databases” on page 25.

To create a test environment for the business integration and mediation modules, follow these steps:

1. Install a database server product on one server and database clients on any additional servers that you use.
2. Install and configure the IBM HTTP Server.

3. Install WebSphere Process Server on System 1. Use the Deployment Environment installation option to create the deployment manager during the install process. This option also creates the common database.
4. Install the Update Installer and recommended maintenance on System 1.
5. Verify and start the deployment manager.
6. Ensure that each additional node that you add to the environment can access the new common database.
7. Install WebSphere Process Server on System 2. Use the Deployment Environment installation option to create a custom node (*Node1*) and add it to the deployment environment during the installation process.

This step creates the clusters that are defined by the pattern that you select and the deployment targets that you choose. Each cluster that is created has one server on Node1. The process installs the applications that are necessary for WebSphere Process Server and defines the system integration buses.
8. Use the Profile Management Tool to create a second custom node on System 2. Use the Deployment environment profile creation option to create the node and to add it to the deployment environment.

This step creates a new node and creates an additional server in each cluster you select.
9. Use the Profile Management Tool to create additional nodes if required.
10. Use the administrative console to create a new deployment environment and cluster for the WebSphere ESB servers. Add two servers to the cluster, one on Node1 and one on Node2.
11. Define the Web server to the deployment manager.

3.3.2 Building the Web server

The IBM HTTP Server ships with WebSphere Process Server and is the Web server that we use in this example. To install the IBM HTTP Server and the Web server plug-in, follow these steps:

1. Execute launchpad.exe.
2. Select **HTTP Server Installation** under **Additional Software** on the left, and then select **Launch the installation wizard for IBM HTTP Server**.
3. Click **Next** on the Welcome screen.
4. Read and accept the license agreement. Click **Next**.

5. The wizard checks the system for prerequisites. If the system passes the check, continue by clicking **Next**.
6. Select an install location. For example:
`C:\HTTPServer`
Click **Next**.
7. Select the HTTP ports for the Web server and for the administration server. The defaults are:
 - HTTP Port: 80
 - HTTP Administration port: 8008Click **Next**.
8. On Windows systems, you have the option to run the server as a Windows server. Select the appropriate options for your system, and click **Next**.
9. The next dialog box allows you to create the user ID and password for HTTP administration authentication. The user ID and password that you enter is encrypted in the *HTTP_install_root/conf/admin.passwd* file.
Click **Next**.
10. The next dialog box provides the option to install the Web server plug-in for the IBM HTTP Server. Select the option to install the plug-in and provide a name for the Web server definition. The default name is *webserver1*.
You also need to provide the host name or IP address of the application server to which you route requests. If you have more than one application server, select only one.
Click **Next**.
11. Review the summary of the installation options that you selected, and click **Next** to start the installation.
12. When the installation completes, click **Finish**.
13. Start the IBM HTTP Server and the HTTP Administration server. On Windows, if you elected to run these processes as Windows services, start the services. On other systems, issue the **apachectl start** command as follows:
`HTTPServer_root/bin/apachectl start`
14. Verify that the HTTP Server is running by opening a Web browser and going to the URL for the server:
`http://localhost`
Note that port 80 is the default port for HTTP. If you used a port other than 80, you need to specify that on the URL.

3.3.3 Building the deployment manager

Preparation: You need to select a database server during this process to host the common database. In our environment, we installed DB2 UDB V9.1 on the deployment manager system before starting the WebSphere Process Server installation.

On the system where the deployment manager is hosted, follow these steps:

1. Execute launchpad.exe.
2. Select **WebSphere Process Server installation** in the left panel, and select **Launch the installation wizard for WebSphere Process Server for Multiplatforms** on the right.
3. Click **Next** at the Welcome screen.
4. Read and accept the license agreement. Click **Next**.
5. The wizard performs a prerequisite check. If the check is successful, click **Next**.
6. Select **Deployment Environment Installation** as the installation type (as shown in Figure 3-17), and click **Next**.

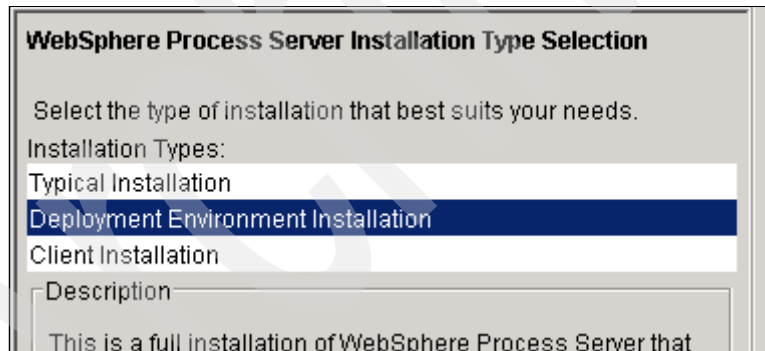


Figure 3-17 Installation type

7. Leave the optional features unselected, and click **Next**.
8. Select an installation location. For example:
C:\WebSphere\ProcServer
Click **Next**.

9. Select **Create a deployment manager and choose a deployment environment pattern**, as shown in Figure 3-18. Click **Next**.



Figure 3-18 Deployment environment installation

10. Enter a user ID and password to have administrative authority. Click **Next**.
11. Select the deployment environment pattern (see Figure 3-19). The response to this step varies depending on the pattern that you select. Click **Next**.

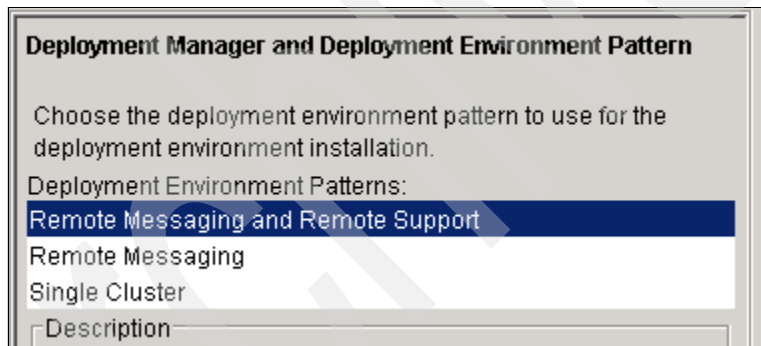
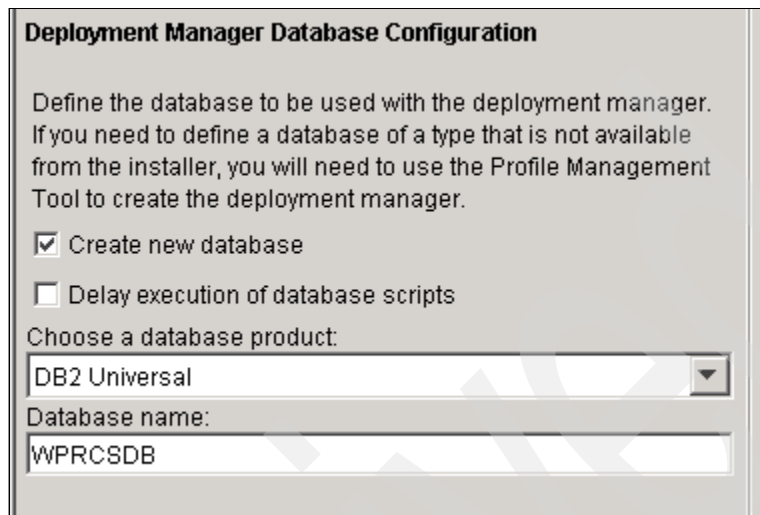


Figure 3-19 Select the deployment pattern

12. Select a database product and database name for the common database (WPRCSDB), as shown in Figure 3-20. The database is created automatically in this case. Click **Next**.



The image shows a dialog box titled "Deployment Manager Database Configuration". It contains the following elements:

- Text: "Define the database to be used with the deployment manager. If you need to define a database of a type that is not available from the installer, you will need to use the Profile Management Tool to create the deployment manager."
- Checkboxes:
 - ☒ Create new database
 - ☐ Delay execution of database scripts
- Text: "Choose a database product:"
- Dropdown menu: Shows "DB2 Universal" with a downward arrow button.
- Text: "Database name:"
- Text input field: Contains "WPRCSDB".

Figure 3-20 Select a database product

13. Depending on your selection for database product, the next dialog box might ask for additional information to define the location of the database server, as shown in Figure 3-21. Click **Next**.

Additional Database Configuration

Additional information is required to complete database configuration for the DB2 Universal database.

User name to authenticate with the database:
db2admin

Password for database authentication:

Confirm password:

Location (directory) of the JDBC driver classpath:
C:\WebSphere\ProcServer\universalDriver_wbi\lib
Browse...

JDBC driver type:
☐ 2
☒ 4

Database server host name (for example, IP address):
kpmgyw0.itso.ral.ibm.com

Server port:
50000

Figure 3-21 Additional database information

14. Review the installation selections that you made, and click **Next** to begin the installation and profile creation.
15. When the installation completes, select **Launch the First steps console**, and click **Finish**.
16. In the First Steps console, select **Installation verification** to start the deployment manager and then verify that it was installed correctly.
17. Stop the deployment manager from the First Steps window.
18. Install the Update Installer from the Launchpad.
19. Download the recommended maintenance install it.
20. Start the deployment manager.

21. Log in to the administrative console and determine the SOAP port used by the deployment manager. You need this information when you create the remaining nodes.

In the console, select **System administration** → **Deployment manager**. The SOAP port is the value listed in the Ports section for SOAP_CONNECTOR_ADDRESS.

3.3.4 Building Node1 and the first process server of the cluster

Preparation: The profile creation process needs to access the common database that was created during the deployment manager creation. In this example, we installed the DB2 UDB client on this node and defined the DB2 server on the deployment manager system to it. Then, we defined the WPRCSDB database to the client and tested the connection.

On the system where the first node of your cluster is hosted, follow these steps:

1. Execute launchpad.exe.
2. Select **WebSphere Process Server installation** in the left panel, and select **Launch the installation wizard for WebSphere Process Server for Multiplatforms** on the right.
3. Click **Next** at the Welcome screen.
4. Read and accept the license agreement. Click **Next**.
5. The wizard performs a prerequisite check. If the check is successful, click **Next**.
6. Select **Deployment Environment Installation** as the installation type.
7. Leave the feature selections unselected, and click **Next**.
8. Select an installation location. For example:

C:\WebSphere\ProcServer

Click **Next**.

9. Select **Create a custom profile and choose the cluster members within the deployment environment** as shown in Figure 3-22. Click **Next**.

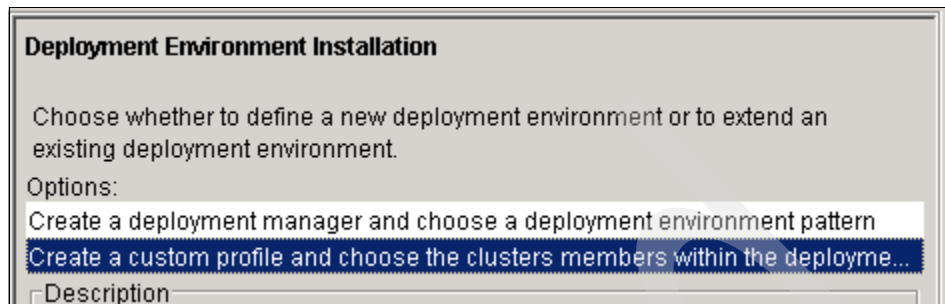


Figure 3-22 Select the custom profile option for the deployment environment

10. The custom node that you create is federated to the deployment manager, requiring that the deployment manager is running and that you have network connectivity between the two nodes.
- Enter the host name and SOAP port for the deployment manager.
 - Enter the user ID and password you specified as having administrative authority when you created the deployment manager.
 - Click **Next**.
11. Select the clusters that are defined by the deployment environment pattern to which to assign this node. This step varies depending on the pattern that you select:
- **Single cluster pattern:** You only have one choice, Application Deployment Target. Select this option.
 - **Remote messaging pattern:** You can select Application deployment target, messaging infrastructure, or both.
 - **Remote messaging and remote support pattern:** You can select Application deployment target, Messaging infrastructure, Support infrastructure, or any combination of these.

To complete the deployment environment, you need at least one node with a server in each cluster. In our examples, we have two nodes, each with a server in each cluster. Figure 3-23 on page 80 shows the selection for the remote messaging and remote support pattern, which matches the configuration in Figure 3-16 on page 71. We select all three cluster types, meaning that this node has three servers, each server in one of the selected clusters. Later, you add a second node and make the same selection.

Cluster and Database Configuration

Choose at least one cluster to assign this node to on the deployment environment topology:

☒ Application deployment target

Consists of a cluster to which user applications need to be deployed. Depending on the chosen deployment environment pattern, the application deployment target cluster may also assume the functionality of the messaging and the supporting infrastructure clusters.

☒ Messaging Infrastructure

Consists of a cluster where the bus members are located.

☒ Support Infrastructure

Consists of a cluster that hosts the Common Event Infrastructure server and other infrastructure services that are used to manage your system.

Figure 3-23 Cluster and database configuration

Click **Next**.

12. Select the database product that is used for the common database and provide the location of the JDBC™ driver classpath on the system that hosts the new node (shown in Figure 3-24). Click **Next**.

Cluster and Database Configuration

Database product used on the deployment manager:

DB2_Universal

Location (directory) of the JDBC driver classpath:

C:\WebSphere\ProcServer\universalDriver_wbi\lib

Browse...

Figure 3-24 Select the database product and driver

13. Review the installation selections that you made, and click **Next** to begin the installation and profile creation.
14. When the installation completes, clear the **Launch the First steps console** check box, and click **Finish**.
15. Stop the node from the First Steps menu.

16. Install the Update Installer from the Launchpad.
17. Install the recommended maintenance.
18. Start the node from the First Steps menu, or use the following command:

```
profile_root/bin/startNode(.bat/.sh)
```
19. Log in to the administrative console to see the new cluster and servers.
 To see the cluster, select **Servers** → **Clusters**. Figure 3-25 shows the clusters created for the remote messaging and remote support pattern.

<div> <div>New</div> <div>Delete</div> <div>Start</div> <div>Stop</div> <div>Ripplestart</div> <div>ImmediateStop</div> </div>		
<div> <div></div> <div></div> <div></div> <div></div> </div>		
Select	Name	Status
<input type="checkbox"/>	default.AppTarget	
<input type="checkbox"/>	default.Messaging	
<input type="checkbox"/>	default.Support	
Total 3		

Figure 3-25 New clusters

To see the new application servers, select **Servers** → **Application servers**. Figure 3-26 shows the new servers created on Node1 and the clusters to which they are assigned.

Name	Node	Version	Cluster Name	Status
default.AppTarget.kpmgyk3Node01.0	kpmgyk3Node01	Business Process Choreographer 6.1.0.0 ND 6.1.0.13	default.AppTarget	
default.Messaging.kpmgyk3Node01.0	kpmgyk3Node01	Business Process Choreographer 6.1.0.0 ND 6.1.0.13	default.Messaging	
default.Support.kpmgyk3Node01.0	kpmgyk3Node01	Business Process Choreographer 6.1.0.0 ND 6.1.0.13	default.Support	

Figure 3-26 New servers on Node1

3.3.5 Building Node2 and the second process server of the cluster

The code was installed on the system for Node2, so we use the Profile Management Tool to create the profile as follows:

1. Start the Profile Management Tool from the First Steps console.
2. Click **Create**.
3. Click **Next** on the Welcome screen.
4. Select **WebSphere Process Server** as the environment type. Click **Next**.
5. Select **Custom profile** as the profile type. Click **Next**.
6. Select **Deployment environment profile creation** as the profile creation option. Click **Next**.
7. Provide a name for the profile and a directory name to use to create the profile in. Click **Next**.
8. Provide a node name and the host name for the node. Click **Next**.
9. The custom node that you create is federated to the deployment manager requiring that the deployment manager is running and that you have network connectivity between the two nodes.
 - a. Enter the host name and SOAP port for the deployment manager.
 - b. Enter the user ID and password that you specified as having administrative authority when you created the deployment manager.
 - c. Click **Next**.
10. Review the port selection and update them if necessary. Click **Next**.
11. Select the clusters that are defined by the deployment environment pattern to which to assign this node. This step varies depending on the pattern that you select:
 - **Single cluster pattern**: You only have one choice, Application Deployment Target. Select this option.
 - **Remote messaging pattern**: You can select Application deployment target, messaging infrastructure, or both.
 - **Remote messaging and remote support pattern**: You can select Application deployment target, Messaging infrastructure, Support infrastructure, or any combination of these.

To complete the deployment environment, you need at least one node with a server in each cluster. In our examples, we have two nodes, each with a server in each cluster. In this example, we select all three cluster types, meaning that this node has three servers, each server in one of the selected clusters.

Click **Next**.

12. Select the database product that is used on the deployment manager and provide the location of the JDBC driver classpath on the system that hosts the new node. Click **Next**.
13. Review the installation selections that you made, and click **Create** to begin the installation and profile creation.
14. When the installation completes, clear the **Launch the First steps console** check box, and click **Finish**.
15. Log in to the administrative console to see the new servers.

Select **Servers** → **Application servers**. Figure 3-27 shows the new servers and their clusters. Each cluster now has two servers, one on each node.

Name ▾	Node ▾	Version ▾	Cluster Name ▾	Status
default.AppTarget.kpmgyk3Node01.0	kpmgyk3Node01	Business Process Choreographer 6.1.0.0 ND 6.1.0.13	default.AppTarget	➡
default.AppTarget.kpmgyw0Node02.0	kpmgyw0Node02	Business Process Choreographer 6.1.0.0 ND 6.1.0.13	default.AppTarget	✖
default.Messaging.kpmgyk3Node01.0	kpmgyk3Node01	Business Process Choreographer 6.1.0.0 ND 6.1.0.13	default.Messaging	➡
default.Messaging.kpmgyw0Node02.0	kpmgyw0Node02	Business Process Choreographer 6.1.0.0 ND 6.1.0.13	default.Messaging	✖
default.Support.kpmgyk3Node01.0	kpmgyk3Node01	Business Process Choreographer 6.1.0.0 ND 6.1.0.13	default.Support	➡
default.Support.kpmgyw0Node02.0	kpmgyw0Node02	Business Process Choreographer 6.1.0.0 ND 6.1.0.13	default.Support	✖

Figure 3-27 New servers on Node2

16. Start the new servers by selecting **Deployment environments**. Select the new deployment environment, and click **Start**.

3.3.6 Creating the cluster for mediation modules

The mediation modules are deployed to a cluster that consists of WebSphere ESB capable servers. Note that you can also deploy the mediation modules to WebSphere Process Server capable servers.

The nodes to be included in the new environment must already exist.

To create a deployment environment from the administrative console:

1. Determine the number of nodes that you need. If the nodes that you include do not exist, create a custom profile for each node using the Profile Management Tool or through the WebSphere Process Server installation process with the following options:
 - WebSphere ESB or WebSphere Process Server profile environment
 - Custom profile type
 - Advanced profile creation
 - Federate the profileComplete the profile creation and start the new node.
2. In the administrative console, navigate to **Servers** → **Deployment environments**. Click **New**.

3. Select the option to create a new deployment environment. Enter a name for the deployment environment, and select **WPS** or **WESB** as the runtime capability as shown in Figure 3-28.
Click **Next**.

Create Deployment Environment

☒ Create a new deployment environment

☐ Load an external deployment environment definition

File path

* Deployment environment name

Runtime capability

▼

☐ Show only steps that need my attention

Figure 3-28 Create a new deployment environment

4. Select the deployment environment type. When you create a deployment environment using the administrative console, you have an additional option that allows you to build a custom deployment environment. See Figure 3-29.

Deployment Environment Patterns

☐ Remote Messaging and Remote Support

This pattern defines one cluster for application deployment, one remote cluster for the messaging infrastructure, and one remote cluster for the Common Event Infrastructure and other supporting applications. This pattern configures a setup that performs well for most of your business integration needs. If in doubt, select this pattern.

☐ Remote Messaging

The remote messaging pattern defines one cluster for application deployment and one remote cluster for the messaging infrastructure. The Common Event Infrastructure and other supporting applications are configured on the application deployment target cluster.

☒ Single Cluster

The single cluster is the simplest pattern that defines one cluster for application deployment. Both messaging infrastructure and Common Event Infrastructure with supporting applications are configured on the application deployment target cluster.

☐ Custom

If none of the given patterns suit your needs, you can customize the deployment environment by configuring your own clusters and servers. This option is intended for advanced users who know how to tune their deployment environment.


Previous Next Cancel



Figure 3-29 Select the environment type

5. Select the nodes to include in the new deployment environment (shown in Figure 3-30). Note that the nodes that exist are WebSphere Process Server capable. WebSphere Process Server capable nodes have the highest capability and can host application servers with WebSphere ESB or WebSphere Application Server capability.

Click **Next**.

Select Nodes

 The *Single Cluster* deployment environment *WESB* needs at least **1 node**. 2 nodes are needed for high availability and failover. Select the nodes you would like to use for the deployment environment.



Select	Node 	Capability 	Version	Base Version	Host
<input type="checkbox"/>	HTTPServerNode	WAS	6.1.0.13	6.1.0.13	xpxv812.itso.ral.ibm.co
<input checked="" type="checkbox"/>	kqfrz2rNode01	WPS	6.1.0.0	6.1.0.13	kqfrz2r.itso.ral.ibm.com
<input checked="" type="checkbox"/>	xpxv812Node01	WPS	6.1.0.0	6.1.0.13	xpxv812.itso.ral.ibm.co
Number of nodes needed		1			
Number of selected nodes		2			

Figure 3-30 Select the nodes

6. Map the nodes to the target cluster type, as shown in Figure 3-31. In a single cluster environment, the only option is the application deployment target.

Node	Capability	Version	Base Version	Application Deployment Target
kqfrz2rNode01	WPS	6.1.0.0	6.1.0.13	1
xpxv812Node01	WPS	6.1.0.0	6.1.0.13	1

Figure 3-31 Map the nodes to the cluster

7. The next step allows you to manage the database table configuration that is required for the new environment.

To support several CEI server configurations, you need to configure a different event database instance for each additional CEI server (shown in Figure 3-32 on page 89) as follows:

- a. Using the database product, create a new database for the event database. In this example, we created a database called WESBEVT.
- b. Edit the WBI_CEI_EVENT configuration by selecting the check box to the left of it and clicking **Edit**. Change the database name, and update the user ID and password that are required to access the new database.
Alternatively, you can set up different event database users for each CEI server configuration for databases that map user IDs to schemas.
- c. Click **Next**.

<input type="button" value="Edit..."/> <input type="button" value="Reset"/> <input type="button" value="Test Connection"/> <input type="button" value="Edit Provider..."/>							
Select	Component	Database Instance	Schema	Create Tables	User Name	Password	Server
<input type="checkbox"/>	WBI_CEI_EVENT	WESBEVT		<input checked="" type="checkbox"/>	db2admin	*****	kpm
<input type="checkbox"/>	WBI_CEI_ME	WPRCSDB	WPRCM01	<input checked="" type="checkbox"/>	db2admin	*****	kpm
<input type="checkbox"/>	WBI_SCA_SYS_ME	WPRCSDB	WPRSS01	<input checked="" type="checkbox"/>	db2admin	*****	kpm
<input type="checkbox"/>	WBI_SCA_APP_ME	WPRCSDB	WPRSA01	<input checked="" type="checkbox"/>	db2admin	*****	kpm

Figure 3-32 Specify the databases and schema

- Enter the user ID and password that are used to create the authentication alias for the event database (Figure 3-33).

Component	User name	Password	Confirm Password	Description
WBI_CEI_EVENT	Administrat	*****	*****	CEI JMS authentication alias

Figure 3-33 Enter the user name and password that can access the event database

- Review the selections, and click **Finish and Generate Environment**.
- If there are no errors, save click **Save Changes**.

3.3.7 Defining the Web server to the cell

The last step in setting up the deployment environment is to define the Web server to the cell. In this scenario, the Web server is on a system that does not have a WebSphere node, and so it is an *unmanaged* server. In this example, we use the administrative console to create the definitions, versus using the scripts generated when the plug-in was installed.

Adding an unmanaged node to the cell

To add an unmanaged node using the administrative console:

1. Select **System Administration** → **Nodes** in the console navigation tree.
2. Click **Add Node**.
3. Select **Unmanaged node**.
4. Click **Next**.
5. Enter the following values in the General Properties page (Figure 3-34):
 - a. Name
Enter a logical name for the node. The name must be unique within the cell. A node name usually is identical to the host name for the computer. However, you can make the node name different than the host name.
 - b. Host name
Enter the host name of the unmanaged node that is added to the configuration.
 - c. Platform Type
Select the appropriate platform type.

General Properties

* Name
HTTPServerNode

* Host Name
webserver1.itso.ral.ibm.com

* Platform Type
Windows ▼

The additional properties will not be available until the general properties for this item are applied or saved.

Additional Properties

■ Custom Properties

Apply OK Reset Cancel

Figure 3-34 General properties for an unmanaged node

6. Click **OK**. The node is added and the name is displayed in the collection on the Nodes page.

Adding a Web server

After you have defined the node for the Web server, you can add the Web server definition. To add a Web server definition:

1. Select **Servers** → **Web servers**.
2. Click **New**. See Figure 3-35.
3. Select the node, and enter the server name and its type. Click **Next**.

→ **Step 1: Select a node for the Web server and select the Web server type**

Step 2: Select a Web server template

Step 3: Enter the properties for the new Web server

Step 4: Confirm new Web server

Select a node for the Web server and select the Web server type

Select a node that corresponds to the Web server you want to add.

Select node
HTTPServerNode ▼

* Server name
webserver1

* Type
IBM HTTP Server ▼

Next **Cancel**

Figure 3-35 Defining a Web server: Step 1

4. Select the template for Web server specification. Initially, this template is supplied with WebSphere and is specific to the Web server type. After you have defined a Web server, you can make it a template for use the next time. See Figure 3-36. Click **Next**.

Step 1: Select a node for the Web server and select the Web server type

→ **Step 2: Select a Web server template**

Step 3: Enter the properties for the new Web server

Step 4: Confirm new Web server

Select a Web server template

Select the template that corresponds to the server that you want to create.

Select	Template Name	Type	Description
<input checked="" type="radio"/>	IHS	System	The IHS Web Server Template

Previous Next Cancel

Figure 3-36 Defining a Web server: Step 2

5. Enter the properties for the Web server as shown in Figure 3-37.

Step 1: Select a node for the Web server and select the Web server type

Step 2: Select a Web server template

→ **Step 3: Enter the properties for the new Web server**

Step 4: Confirm new Web server

Enter the properties for the new Web server

Enter the Web server properties.

- * Port: 80
- Web server installation location: C:\HTTPServer
- * Service name: IBMHTTPServer6.1
- * Plug-in installation location: c:\HTTPServer\Plugins
- Application mapping to the Web server: All

Enter the IBM Administration Server properties.


- * Port: 8008
- * Username: Administrator
- * Password:
- * Confirm password:
- ☐ Use SSL

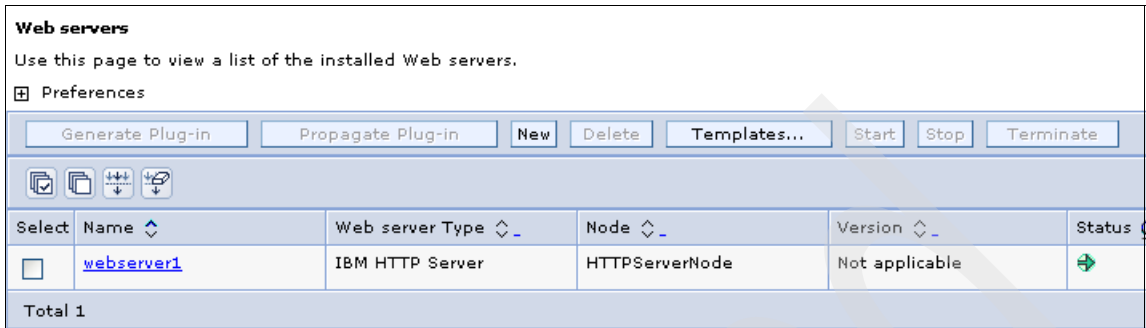
Previous Next Cancel

Figure 3-37 Defining a Web server: Step 3


When defining a Web server hosted on a Windows operating system, use the real service name instead of the display name. The service name does not contain spaces. If you do not use the service name, you might have problems starting and stopping the service.

6. Review the options, and click **Finish**.
7. Save your changes.
8. View the status of the new server by selecting **Servers** → **Web servers**. If a Web server is started or stopped using a native command, you might need to

refresh the view by clicking the  icon to see the new status. See Figure 3-38.



The screenshot shows a 'Web servers' window with a title bar. Below the title bar is a subtitle 'Use this page to view a list of the installed Web servers.' and a 'Preferences' button. A toolbar contains buttons: 'Generate Plug-in', 'Propagate Plug-in', 'New', 'Delete', 'Templates...', 'Start', 'Stop', and 'Terminate'. Below the toolbar is a row of icons: a checkmark, a document, a double-headed arrow, and a refresh icon. The main area is a table with columns: 'Select', 'Name', 'Web server Type', 'Node', 'Version', and 'Status'. There is one row with the following data: an unchecked checkbox, 'webserver1', 'IBM HTTP Server', 'HTTPServerNode', 'Not applicable', and a green arrow icon. At the bottom, it says 'Total 1'.

Select	Name	Web server Type	Node	Version	Status
<input type="checkbox"/>	webserver1	IBM HTTP Server	HTTPServerNode	Not applicable	

Total 1

Figure 3-38 Web server status

3.4 Augmenting a WebSphere ESB profile to WebSphere Process Server

In the event that you create a WebSphere Application Server or WebSphere ESB profile and discover that you need that profile to have additional function, you can *augment* the profile.

To augment a WebSphere ESB stand-alone profile to a WebSphere Process Server stand-alone profile, follow these steps:

1. Start the Profile Management Tool, and click **Augment**.
2. The profiles that exist in your installation are listed. Select the WebSphere ESB stand-alone profile that you want to augment, as shown in Figure 3-39.

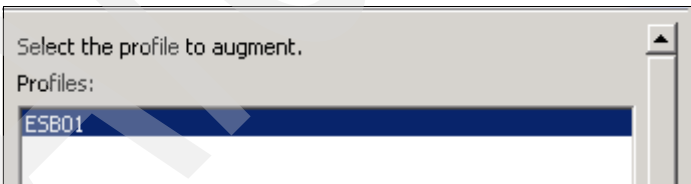


Figure 3-39 Select the profile to augment

3. Select **WebSphere Process Server** as the profile type to which to augment the existing profile, as shown in Figure 3-40.

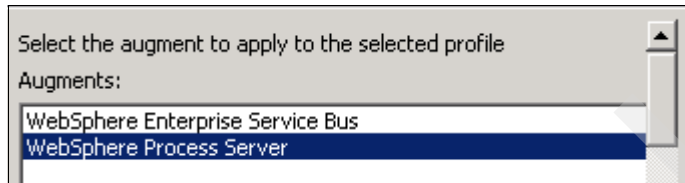


Figure 3-40 Select the augmentation type

4. Select whether to take the typical or advanced path through the augmentation process (see Figure 3-41 on page 96). Table 3-1 shows the options and results that you see, depending on which path you take.

Table 3-1 Profile augmentation options

Typical	Advanced
Does not Configure sample Business Process Choreographer	Allow to choose whether to configure sample Business Process Choreographer
Does not Configure Business Rule Manager	Allow to choose whether to configure Business Rule Manager

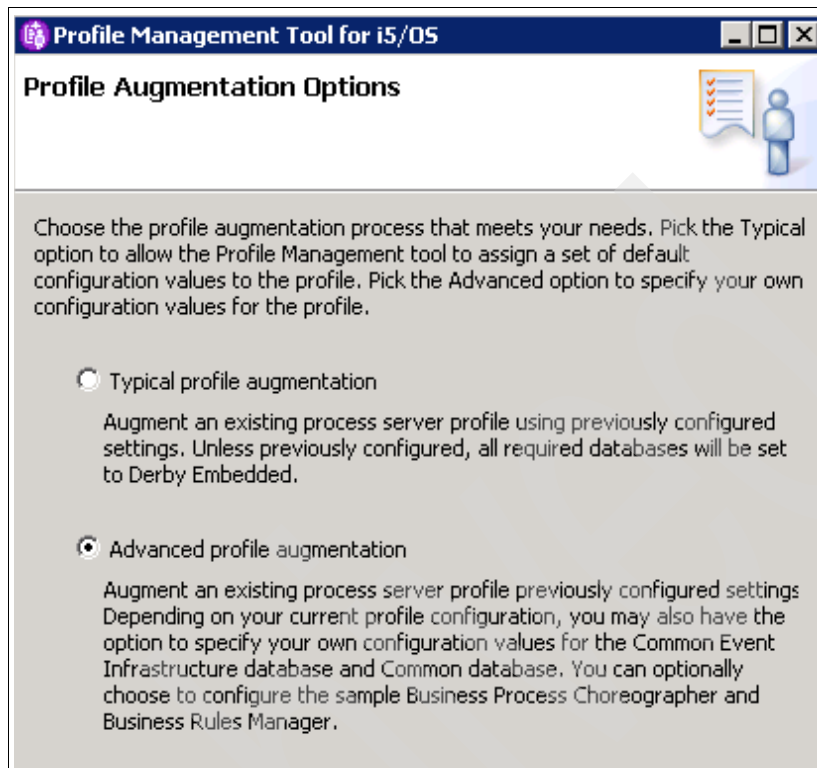


Figure 3-41 Profile Management Tool

5. Review your selections, and click **Augment** to complete the process.

Creating and managing profiles on z/OS

The process for installing and creating profiles on z/OS platforms differs from those actions on distributed and i5/OS platforms. WebSphere Application Server for z/OS serves as the base for WebSphere Process Server, and you install it first. Next, you create the Network Deployment profiles. Then, you install WebSphere Process Server and augment the existing profiles for WebSphere Process Server.

In this chapter, we explain how to create a single cluster WebSphere Process Server configuration on z/OS. This chapter assumes that WebSphere Process Server is installed using the instructions in the WebSphere Information Center:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.z.610.doc/doc/iins_zos_welcome.html

This chapter includes the following topics:

- ▶ Methods of creating profiles
- ▶ Topology built in this chapter
- ▶ Creating a deployment manager node
- ▶ Creating a custom node
- ▶ Configuring the deployment manager node
- ▶ Configuring a custom node
- ▶ Creating a WebSphere Process Server cluster

4.1 Methods of creating profiles

There are three main ways to create profiles on z/OS:

- ▶ Using the WebSphere Application Server for z/OS Profile Management Tool (zPMT) available with the WebSphere Application Server Toolkit (new with V6.1)

zPMT is a dialog tool that runs in the WebSphere Application Server Toolkit. It is an Eclipse plug-in that allows you to do the initial setup of WebSphere Application Server for z/OS cells and nodes. It provides the same functionality as the ISPF dialogs plus additional features to help you.

The zPMT itself does not create the cells and nodes; however, it creates batch jobs, scripts, and data files that you can use to perform WebSphere Application Server for z/OS customization tasks. These jobs, scripts, and data files form a customization definition on your workstation, which is then uploaded to z/OS where you submit the jobs. You use the zPMT to create profiles. It cannot be used to perform functions such as delete or list profiles. Only the **manageprofiles** command-line script interface can perform these functions.

- ▶ Using the ISPF dialogs

Although you can still set up your profile through the ISPF dialog boxes in V6.1, the ISPF customization dialog boxes are deprecated and are removed in a future release.

- ▶ Using the **manageprofiles** script interface.

In the examples in this chapter, we use the zPMT to create profiles.

Using zPMT requires that you have the WebSphere Application Server Toolkit for V6.1 installed. For more information about how to download and install the WebSphere Application Server Toolkit, see:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.zseries.doc/info/zseries/ae/tins_astinstall.html

With zPMT, you have the following options:

- ▶ Create an application server profile
Generates the customization jobs to create a z/OS stand alone application server.
- ▶ Create a cell profile (deployment manager profile and federated application server)
Generates the customization jobs to create a z/OS deployment manager and a federated node that contains an application server. This option only exists in the zPMT.
- ▶ Create a deployment manager profile
Generates the customization jobs to create a z/OS deployment manager cell without a federated application server.
- ▶ Federate an application server to a cell
Generates the customization jobs to federate an existing stand alone z/OS application server into an existing network deployment cell.
- ▶ Create a managed (custom) node
Generates the customization jobs to create a z/OS managed node and federate it into an existing network deployment cell.

All these options are also available through the ISPF Dialogs except the z/OS cell (deploy manager and application server) that is unique to zPMT.

4.2 Topology built in this chapter

This chapter builds the topology shown in Figure 4-1.

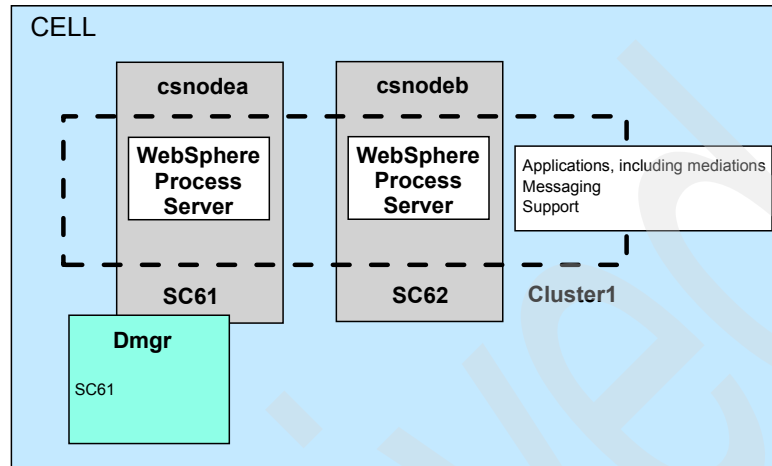


Figure 4-1 WebSphere Process Server topology on z/OS

The topology consists of a single cluster spread over two nodes for WebSphere Process Server. You can configure the cluster for both business integration modules and mediations.

4.3 Creating a deployment manager node

In this section, we show how to use zPMT to create a deployment manager. To create the profile:

1. Open the Application Server Toolkit and select **Window** → **Preferences**.
2. From the Preferences window, expand **Server** category, and select **WebSphere for z/OS Customization**.
3. Click **Create** (on the top right corner).

The Welcome window for the Profile Management Tool opens. Note that this window contains a link to the WebSphere Application Server V6.1 Information Center.

Click **Next**.

4. Select the appropriate configuration as shown in Figure 4-2, in this case **z/OS deployment manager**, and click **Next**.

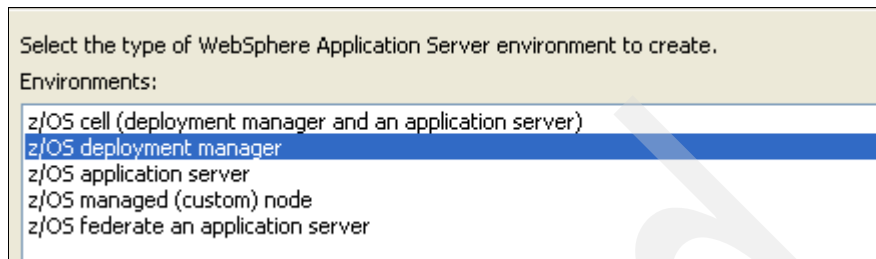


Figure 4-2 Creating a profile: Environment selection window

5. The next window, complete the following fields (as shown in Figure 4-3 on page 102):
 - **Customization definition name:** Used to specify the customization profile you are about to create. This name is not transported to your host system.
 - **Customization definition directory:** Sets the location on the workstation where the CNTL and DATA files are stored and from which they are uploaded to your host system.
 - **Response file path name:** Allows you to specify a saved file with values from a previously created configuration, which populates the fields throughout the windows with the values that are contained in the response file. This field is optional. Because it is the first time that you create a profile, you probably do not have this file. A response file is written each time a z/OS customization definition is created and its name is the *customization definition name.responseFile*. It is created under the root directory for the customization definition. Normally, you specify a response file from a customization definition of the same type as you are about to define. However, you can use a response file from a similar customization type to pre-load most of the default values.

Specify a customization definition name and directory path to contain the batch jobs and instructions that will be used to define the selected z/OS run-time environment.

Customization definition name:

Customization definition directory:

Response file pathname (optional)

Specify the full pathname of the response file that contains the default values. When this value is specified, the input fields in the tool will be pre-loaded with the values in the response file.

Figure 4-3 Customization name and location

After completing the required fields, click **Next**.

- The next window asks you to specify a high-level qualifier for the target z/OS data sets that contain the generated jobs and instructions (Figure 4-4).

Specify a high level qualifier for the target z/OS data sets that will contain the generated jobs and instructions.

High level qualifier (HLQ):

Figure 4-4 Define target data sets

The high-level qualifier is composed of multiple qualifiers up to 39 characters. When you upload a customization profile on the target z/OS system, the generated jobs and files are written on a pair of data sets. You can reuse the same data sets for a future installation; however, we strongly recommend that you create a new pair of data sets for every new profile installation. A good planning and naming convention is crucial when defining this type of information. For example, set the high-level qualifier according to the version and release of WebSphere Application Server for z/OS, the task that you are performing, and the cell (and, in some cases, the node name) that you are configuring.

The CNTL data set is a partitioned data set with a fixed block 80-byte records that keeps the customization jobs. The DATA data set is a partitioned data set as well but with variable length data to contain the other customization data.

Note: After you create the customization profile, you cannot change the data set names, because all jobs are based on these data set names.

After completing the HLQ field, click **Next**.

7. The next window contains the fields to configure common groups and users, as shown in Figure 4-5.

The screenshot shows a configuration window titled "WebSphere Application Server configuration group information". It contains several sections with input fields:

- WebSphere Application Server configuration group information:**
 - Group:
 - GID:
- WebSphere Application Server file system owner information:**
 - User ID:
 - UID:
- WebSphere Application Server servant group information:**
 - Group:
 - GID:
- WebSphere Application Server local user group information:**
 - Group:
 - GID:
- WebSphere Application Server user ID home directory:**
 -

Figure 4-5 Configure common groups and users

You need to complete the following fields:

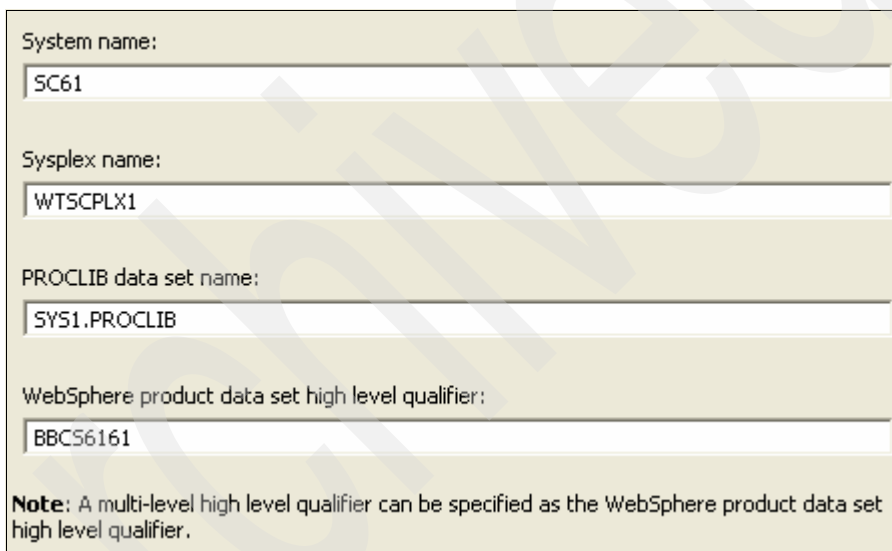
- **WebSphere Application Server configuration group information:** Used to specify the group name for the WebSphere Application Server administrator user ID and all server user IDs.
- **WebSphere Application Server file system owner information:** Specify the user ID that owns the file system.
- **WebSphere Application Server servant group information:** Used to connect all servant user IDs to this group. You can use it to assign

subsystem permissions, such as DB2 authorizations, to all servants in the security domain.

- **WebSphere Application Server local user group information:** Specify the local client group and unauthorized user IDs. This group provides minimal access to the cell.
- **WebSphere Application Server user ID home directory:** Specify a new or existing z/OS file system directory in which home directories for WebSphere Application Server for z/OS user IDs are created by the customization process. Note that this directory does not need to be shared among z/OS systems in a WebSphere Application Server cell.

After completing the required fields, click **Next**.

8. The next window asks for information about the z/OS system (Figure 4-6).



System name:
SC61

Sysplex name:
WTSCPLX1

PROCLIB data set name:
SYS1.PROCLIB

WebSphere product data set high level qualifier:
BBCS6161

Note: A multi-level high level qualifier can be specified as the WebSphere product data set high level qualifier.

Figure 4-6 System Locations: Names and data set qualifier

In this window, complete the following fields:

- **System name:** The system name of the target z/OS system.
- **Sysplex name:** The sysplex name of the target z/OS system. If you are not sure of the System and Sysplex names for your target z/OS system, you can use the console command D SYMBOLS on the target z/OS system to display them.

- **PROCLIB data set name:** The PROCLIB data set where the WebSphere Application Server for z/OS cataloged procedures are added.
- **WebSphere product data set high level qualifier:** This name prepends your system libraries, A multi-level high level qualifier can be specified as the WebSphere product data set high level qualifier.

After completing the required fields, click **Next**.

9. The next window asks for the product data set location information (Figure 4-7). In this window, you specify the product file system directory and the data sets.

In V6.1, most LPA-resident modules are merged into a single member in SBBLOAD, which is loaded into common storage for each node if SBBLOAD is not in LPA. The remaining module in SBBOLPA is BBORTS61, the CTRACE support module for WebSphere Application Server V6.1, which should always be loaded into LPA.

WebSphere Application Server product file system directory:
/usr/lpp/zWebSphereCS/V6R1

SBBOLPA data set name:
BBCS6161.SBBOLPA

SBBOEXEC data set name:
BBCS6161.SBBOEXEC

SBBOMSG data set name:
BBCS6161.SBBOMSG

SBBLOAD data set name:
BBCS6161.SBBLOAD

SBBGLOAD data set name:
BBCS6161.SBBGLOAD

SBBOLD2 data set name:
BBCS6161.SBBOLD2

☒ Run WebSphere Application Server from STEPLIB

Figure 4-7 System Locations: Product data sets

In this window, enter the following information:

- The SBBOLD2 data set is the WebSphere Application Server for z/OS load module library that you installed through SMP/E. It has members that should go into the link list, or into STEPLIB. Do *not* place them in LPA.
- The SBBOEXEC data set contains the WebSphere Application Server for z/OS CLIST library.
- The SBBOMSG data set contains the WebSphere Application Server for z/OS message skeletons for language translation.
- The WebSphere Application Server product file system directory is the name of the directory where WebSphere Application Server for z/OS product files reside after installation.

- The “Run WebSphere Application Server from STEPLIB” option specifies whether to load WebSphere Application Server for z/OS load modules from STEPLIB or from the link pack area and link list. The load modules must be loaded from STEPLIB if you have another instance of WebSphere Application Server for z/OS (Version 4 or later) in the system link pack area or link list.

After completing the required fields, click **Next**.

10. The next window allows you to specify the long and short names to use for the components of the profile as shown in Figure 4-8.

Cell names	
Short name:	CSCCELL
Long name:	cscell
Node names	
Short name:	CSDM
Long name:	cscdm
Server names	
Short name:	CSDMGR
Long name:	dmgr
Cluster transition name:	
CSDMGR	

Figure 4-8 Specify long and short names

In this window, complete the following fields:

- **Cell short name:** Identifies the cell to z/OS facilities, such as SAF.
- **Cell long name:** Specifies the primary external identification of this WebSphere Application Server for this z/OS cell. This name identifies the cell as displayed through the administrative console.
- **Node short name:** Specifies a name that identifies the node to z/OS facilities, such as SAF.

- **Node long name:** Specifies the primary external identification of this WebSphere Application Server for the z/OS node. This name identifies the node as displayed through the administrative console.
- **Cluster transition name:** Specifies the WLM application environment (WLM APPLENV) name for the deployment manager. If this is a server that is converted into a clustered server, this name becomes the cluster short name. The cluster short name is the WLM APPLENV name for all servers that are of the same cluster.

After completing the required fields, click **Next**.

11. The next window contains the file system information for your z/OS system, as shown in Figure 4-9. The file system can be either HFS or zFS.

Mount point:
/wasconfig/cscell/dmgr

Name:
OMVS.WAS61.CSCCELL.CSDM.HFS

Volume, or '*' for SMS:
TSTO3F

Primary allocation in cylinders: 420

Secondary allocation in cylinders: 100

File system type:
☐ Hierarchical File System (HFS)
☒ zSeries File System (ZFS)

Figure 4-9 System environment: configuration file system

In this window, complete the following fields:

- **Mount point:** Sets the read/write HFS directory where application data and environment files are written. The customization process creates this mount point if it does not already exist.
- **Name:** Specifies the file system data set that you create and mount at the specified mount point above.
- **Volume, or '*' for SMS:** Specifies either the DASD volume serial number to contain the data set or '*' to let SMS select a volume. Using '*' requires

that SMS automatic class selection (ACS) routines be in place to select the volume. If you do not have SMS set up to handle data set allocation automatically, list the volume explicitly.

- **Primary allocation in cylinders:** Sets the initial size allocation for the configuration file system data set. In the application server, the total space needed for this data set increases with the size and number of the installed applications. The minimum suggested size is 250 cylinders (3390).
- **Secondary allocation in cylinders:** Sets the size of each secondary extent. The minimum suggested size is 100 cylinders.
- **File System type:** Allocates and mounts your configuration file system data set using HFS or zFS.

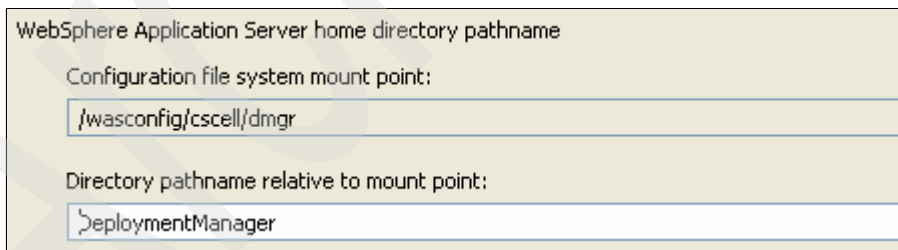
After completing the required fields for the deploy manager configuration file system, click **Next**.

12. In the next window, complete the required information to set up the log stream as follows:

- **Error log stream name:** The name of the WebSphere error log stream that you create.
- **Trace Parmlib member suffix:** The value that is appended to CTIBBO to form the member name for the CTRACE parmlib member.

After completing the required fields, click **Next**.

13. The next window allows you to enter the required information for the deployment manager home directory (as shown in Figure 4-10).



WebSphere Application Server home directory pathname

Configuration file system mount point:

/wasconfig/cscell/dmgr

Directory pathname relative to mount point:

DeploymentManager

Figure 4-10 Defining App_Server_Root

Complete the following information:

- **Configuration file system mount point:** Specifies the read/write file system directory where the application data and environment files are written. This field is not writable here, but was specified earlier on the configuration file system information window (see Figure 4-9 on page 108).
- **Deployment path name relative to mount point:** Specifies the relative path name of the directory within the configuration file system in which the deployment manager configuration resides.

After completing the required fields, click **Next**.

14. The next window allows you to select the administrative console. We recommend that you select it. After selecting the applications, click **Next**.
15. The next window allows you to define the job names, procedure names, and user IDs to use for each process as follows:
 - **Deploy manager controller process:** Specifies the job name in the MVS START command JOBNAME parameter that is associated with the control region. This is the same as the server short name and it cannot be changed during customization. The procedure name is the member name in your procedure library to start the control region. The User ID is the user ID associated with the control region.
 - **Deploy manager servant process:** Specify the job name used by WLM to start the servant regions. This is set to the server short name, followed by the letter *S*, and it cannot be changed during customization. The procedure name is the member name in your procedure library to start the servant regions. The User ID is the user ID associated with the servant regions.

After completing the required fields, click **Next**.

16. The next window allows you to specify the ports to use for each process. Again, good planning is very important to avoid port conflicts, so be sure that you have all values you need in order to fill out this window. The required fields for a deployment manager are:
 - Node host name
 - SOAP JMX™ connector port
 - Cell Discovery Address port
 - ORB Listener host name
 - ORB port
 - ORB SSL port
 - HTTP transport host name
 - Administrative console port
 - Administrative secure console port
 - High availability manager communication port

After completing the required ports, click **Next**.

17. The next window allows you to specify the location daemon settings. The location daemon service is the initial point of client contact in WebSphere Application Server for z/OS. The server contains the CORBA-based location service agent which places sessions in a cell. All RMI/IIOP IORs (for example, enterprise beans) establish connections to the location service daemon first, then forward them to the target application server.

Complete the following settings:

- **Daemon home directory:** Specifies the directory in which the location service daemon resides. This is set to the configuration file system mount point/Daemon and cannot be changed.
- **Daemon job name:** Specifies the jobname of the location service daemon, specified in the JOBNAME parameter of the MVS start command used to start the location service daemon. When configuring a new cell, be sure to choose a new daemon jobname value. A server automatically starts the location service daemon if it is not already running.
- **Procedure name:** Specifies the member name in your procedure library to start the location service daemon.
- **User ID:** Specify the user ID that is associated with the location service daemon.
- **UID:** Specifies the user identifier associated with this user ID. UIDs must be unique numbers within the system.
- **IP name:** Specifies the fully qualified IP name, registered with the Domain Name Server (DNS), that the location service daemon uses. The default is your node host name. In a sysplex, you should consider using a virtual IP address (VIPA) for the location service daemon IP name. Select the IP name for the location service daemon carefully. You can choose any name that you want, but after you choose a name, it is difficult to change it, even in the middle of customization.
- **Listen IP:** Specifies the address at which the daemon listens. Select either an asterisk (*) or a dotted IP address for this value.
- **Port number:** Specify the port number on which the location service daemon listens.
- **SSL port:** The port number on which the location service daemon listens for SSL connections.

Note: Choose the IP name and port number carefully because these values are difficult to change, even in the middle of customization.

- Register daemon with WLM DNS check box: If you use the WLM DNS (connection optimization), you must register your location service daemon. Otherwise, do not register your location service daemon. Only one location service daemon per LPAR can register its domain name with WLM DNS; if you have multiple cells in the same LPAR and register more than one location service, it will fail to start.

After completing the required fields, click **Next**.

18. The next window allows you to enter SSL configuration values. as follows:

- **Certificate authority keylabel:** Specifies the name that identifies the certificate authority (CA) to be used in generating server certificates.
- **Generate certificate authority certificate:** Selected to generate a new CA certificate. Do not select this option to have an existing CA certificate generate server certificates.
- **Expiration date for certificates:** Used for any X509 Certificate Authority certificates created during customization, as well as the expiration date for the personal certificates generated for WebSphere Application Server for z/OS servers. You must specify this even if you have not selected **Generate Certificate Authority (CA) certificate**.
- **Default SAF keyring name:** Specifies the default name given to the RACF® keyring used by WebSphere Application Server for z/OS. The keyring names created for repertoires are all the same within a cell.
- **Enable SSL on location service daemon:** Supports secure communications using Inter-ORB Request Protocol (IIOP) to the location service daemon using SSL. If selected, a RACF keyring is generated for the location service daemon to use.

After completing the required SSL information, click **Next**.

19. The next window allows you to select the user registry to be used for administrative security. You can choose from the following options:

- **z/OS security product option:** This option uses the z/OS system's SAF-compliant security product, such as IBM RACF or equivalent, to manage WebSphere Application Server identities and authorization according to the following:
 - The SAF security database is used as the WebSphere user repository.
 - SAF EJBROLE profiles is used to control role-based authorization, including administrative authority.
 - Digital certificates is stored in the SAF security database.

Note: Select the z/OS security product option if you are planning to use the SAF security database as your WebSphere Application Server registry or if you plan to set up an LDAP or custom user registry whose identities are mapped to SAF user IDs for authorization checking. For this security option, you must decide whether to set a security domain name and choose an administrator user ID and an unauthenticated (guest) user ID.

- **WebSphere Application Server security option:** The WebSphere Application Server administrative security option is used to manage the Application Server identities and authorization according to the following:
 - A simple file-based user registry is built as part of the customization process.
 - Application-specific role binds is used to control role-based authorization.
 - The WebSphere Application Server console users and groups list controls administrative authority.
 - Digital certificates is stored in the configuration file system as keystores.

Note: Choose this option if you plan to use an LDAP or custom user registry without mapping to SAF user IDs. (The file-based user registry is not recommend for production use.)

- **No security:** Although it is not recommended, you can disable administrative security. If you choose this security option, there are no other choices to make. Your WebSphere Application Server environment is not secured until you configure and enable security manually. You can enable security manually later using the administrative console or using Jython scripts.

Select an option and click **Next**.

20. The options in the next window depend on the security option that you choose. In this example, we chose the z/OS product option. Complete the following information:

- **Security domain identifier:** (Optional) Used to distinguish between APPL or EJBROLE profiles based on security domain name; provides an alphanumeric security domain name of one to eight characters. Internally, this sets SecurityDomainType to the string cellQualified.

All servers in the cell prepend the security domain name you specify to the application-specific J2EE role name to create the SAF EJBROLE profile for checking. The security domain name is not used; however, if role checking is performed using WebSphere Application Server for z/OS bindings. The security domain name is also used as the APPL profile name and inserted into the profile name used for CBIND checks. The RACF jobs that the Customization Dialog generates create and authorize the appropriate RACF profiles for the created nodes and servers. If you do not want to use a security domain identifier, leave this field blank.

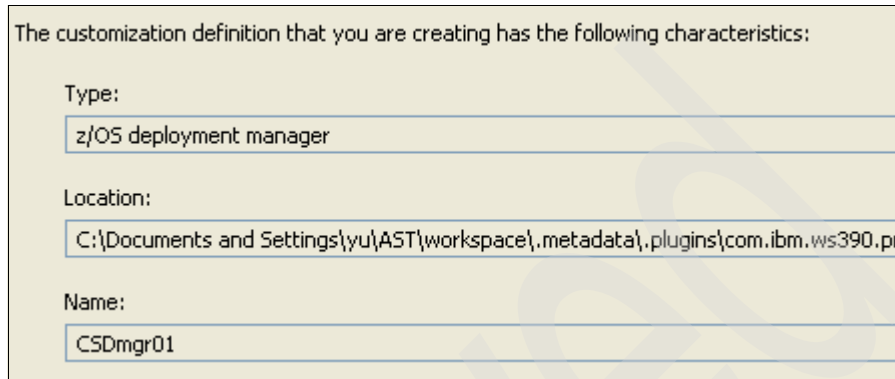
- **WebSphere Application Server administrator user ID:** The initial WebSphere Application Server administrator. It must have the WebSphere Application Server configuration group as its default UNIX System Services group. The UNIX System Services UID number for the administrator user ID is specified here, and must be a unique numeric value between 1 and 2 147 483 647.
- **WebSphere Application Server unauthenticated user ID:** Associated with unauthenticated client requests. It is sometimes referred to as the *guest* user ID. Give it the RESTRICTED attribute in RACF, to prevent it from inheriting UACC-based access privileges. The UNIX System Services UID number for the user ID is specified here and is associated with unauthenticated client requests. The UID value must be unique numeric values between 1 and 2 147 483 647.

Click **Next**.

21. The next window allows you to tailor the JCL for the customization jobs. Enter a valid job statement for your installation. The profile creation process updates the job name for you in all the generated jobs, so you need not be concerned with that portion of the job statement. If continuation lines are needed, replace the comment lines with continuation lines.

Click **Next**.

22. The last window shows a short summary of the customization, including profile type and where the generated jobs are stored (Figure 4-11). To change the characteristics of this profile, click **Back**. Otherwise, click **Create** to generate your z/OS Customization jobs.



The customization definition that you are creating has the following characteristics:

Type:	z/OS deployment manager
Location:	C:\Documents and Settings\yu\AST\workspace\metadata\plugins\com.ibm.ws390.pr
Name:	CSDmgr01

Figure 4-11 Creating a profile: Customization creation summary window

23. When zPMT is done, it displays a summary window that indicates whether the jobs were created successfully. If the jobs were not created, a log file containing failure information is identified. If successful, the next step in the z/OS customization process is to upload these jobs and the associated instructions to a pair of z/OS partitioned data sets. Click **Finish** to return to the WebSphere for z/OS Customization screen.
24. Now, select the profile that you created, and then click **Upload**.
- In the upload customization definition window, enter the target z/OS system using a fully qualified host name or the upload will fail.
- Use the **Allocate target z/OS data sets** check box to specify whether to allocate the data sets if they do not exist (box check). If the data sets exist and are to be reused, clear the box. Then, click **Upload** again.
25. You see a progress information window while the upload is occurring. After you upload the customization profile, follow the instructions in the BBOSSINS member of the CNTL data set. You can view this data set on the host or select the configuration and use the View button. These instructions can help you determine what jobs to run, the order to run them in, and the expected results. It also tells you how to start the environment when you are done. From this point, the process is identical to the process that you use with the ISPF Customization Dialog.

4.4 Creating a custom node

In this section, we are going to explain how to use the zPMT to generate the jobs, scripts, and data files that you need to create two custom nodes that are augmented for WebSphere Process Server and then federated to the deployment manager.

Many of the choices that you make here are similar to those made for the deployment manager. This example takes you through the process and describes settings unique to a custom node configuration.

To create a custom profile:

1. Open the Application Server Toolkit and select **Window** → **Preferences**.
2. From the Preferences window, expand **Server** category and select **WebSphere for z/OS**.
3. Click **Create** (on the top right corner), and then click **Next**.
4. Select **z/OS managed (custom) node** (as shown in Figure 4-12), and click **Next**.

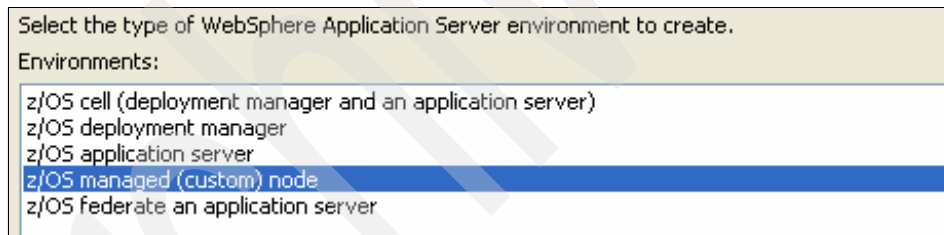


Figure 4-12 Creating a profile: Environment selection window

5. Define a name for the customization definition and the location to store the files (as shown in Figure 4-13).

If using a response file to populate values in the wizard, enter the name of the response file. Response files created from previous configurations are stored in `customization_root/customization_definition_name.responseFile`.

instructions that will be used to define the selected z/OS run-time environment.

Customization definition name:

CSCustom01

Customization definition directory:

C:\Documents and Settings\yu\AST\workspace\metadata\plugins\com.ibm.w

Response file pathname (optional)

Specify the full pathname of the response file that contains the default values. When this value is specified, the input fields in the tool will be pre-loaded with the values in the response file.

Figure 4-13 Creating a profile: customization name and location window

After completing the required fields, click **Next**.

6. Review and modify, if necessary, the default values for GID, UID, cell and system identifiers for naming conventions, and a specific port range to use.
7. Specify a high-level qualifier for the target z/OS data sets that contains the generated jobs and instructions (Figure 4-14) and click **Next**.

Specify a high-level qualifier for the target z/OS data sets that will contain the generated jobs and instructions.

High-level qualifier (HLQ):

BBC56162

Figure 4-14 Creating a profile: target data sets window

8. The next window contains the fields to configure common groups and users (Figure 4-15 on page 119). There are five main sections that you need to complete on this window:
- **WebSphere Application Server configuration group information:** Specify the group name for the WebSphere Application Server administrator user ID and all server user IDs.
 - **WebSphere Application Server file system owner information:** Specify the user ID that owns the file system.
 - **WebSphere Application Server servant group information:** Used to connect all servant user IDs to this group. You can use it to assign subsystem permissions, such as DB2 authorizations, to all servants in the security domain.
 - **WebSphere Application Server local user group information:** Specify the local client group and unauthorized user IDs. This group provides minimal access to the cell.
 - **WebSphere Application Server user ID home directory:** Specify a new or existing z/OS file system directory in which home directories for WebSphere Application Server for z/OS user IDs are created by the customization process. Note that this directory does not need to be shared among z/OS systems in a WebSphere Application Server cell.

After completing the required fields, click **Next**.

WebSphere Application Server configuration group information

Group:

GID

☒ Allow OS security to assign GID

☐ Assign user-specified GID:

WebSphere Application Server file system owner information

User ID:

UID

☒ Allow OS security to assign UID

☐ Assign user-specified UID:

WebSphere Application Server servant group information

Group:

GID

☒ Allow OS security to assign GID

☐ Assign user-specified GID:

WebSphere Application Server local user group information

Group:

GID

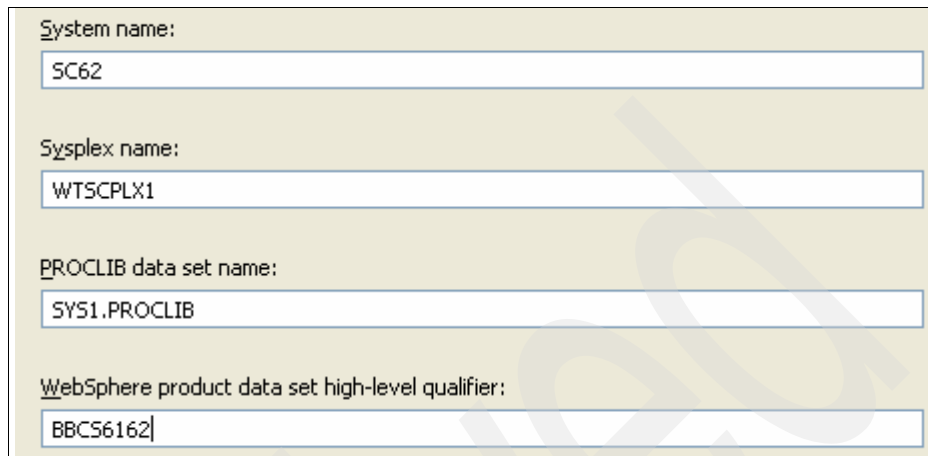
☒ Allow OS security to assign GID

☐ Assign user-specified GID:

WebSphere Application Server user ID home directory:

Figure 4-15 Creating a profile: Configure common groups and users window

9. The next window asks for information about the z/OS system as shown in Figure 4-16.



System name:
SC62

Sysplex name:
WTSCPLX1

PROCLIB data set name:
SYS1.PROCLIB

WebSphere product data set high-level qualifier:
BBCS6162

Figure 4-16 Creating a profile: System Locations: Names and data set qualifier window

You can use the console command `D SYMBOLS` on the target z/OS system to display the system and sysplex name.

After completing the required fields, click **Next**.

10. The next window asks for the product data set location information (see Figure 4-17). In this window, you specify the product file system directory and the data sets. The considerations here are the same as for the deployment manager profile

WebSphere Application Server product file system directory:
/usr/lpp/zWebSphereCS/V6R1

SBBOLPA data set name:
BBCS6162.SBBOLPA

SBBOEEXEC data set name:
BBCS6162.SBBOEXEC

SBBOMSG data set name:
BBCS6162.SBBOMSG

SBBOLOAD data set name:
BBCS6162.SBBOLOAD

SBBOGLOAD data set name:
BBCS6162.SBBGLOAD

SBBOLD2 data set name:
BBCS6162.SBBOLD2

☒ Run WebSphere Application Server from STEPLIB

Figure 4-17 zPMT: System locations

After completing the required fields, click **Next**.

11. The next window allows you to specify the long and short names for the cell and the node (Figure 4-18).

z/OS Profile Management Tool

Server: Cell and Node Names
z/OS managed (custom) node

Cell names

Short name: CEMPTYB
Long name: cemptyb

Node names

Short name: CSNODEB
Long name: csnodeb

? < Back Next > Finish Cancel

Figure 4-18 Custom node short and long names

In this window, complete these fields:

- **Cell short name:** Identifies the cell to z/OS facilities, such as SAF.
- **Cell long name:** Specifies the primary external identification of this WebSphere Application Server for this z/OS cell. This name identifies the cell as displayed through the administrative console.
- **Node short name:** Specifies a name that identifies the node to z/OS facilities, such as SAF.
- **Node long name:** Specifies the primary external identification of this WebSphere Application Server for the z/OS node. This name identifies the node as displayed through the administrative console.

After completing the required fields, click **Next**.

12. The next window contains the file system information for the node configuration files (Figure 4-19). The file system can be either HFS or zFS. It is used to hold WebSphere Application Server configuration information.

Mount point:
/wasconfig/cscell/csnodeb

Name:
OMVS.WAS61.CSCCELL.CSNODEB.HFS

Volume, or '*' for SMS:
TSTO48

Primary allocation in cylinders: 450

Secondary allocation in cylinders: 100

File system type:
☐ Hierarchical File System (HFS)
☒ zSeries File System (ZFS)

Figure 4-19 zPMT: Configuration file system

After completing the required fields for the custom node file system, click **Next**.

13. In the next window complete the required information to set up the log stream (Figure 4-20 on page 124) as follows:

- **Error log stream name:** Specifies the name of the WebSphere error log stream that you create.
- **Trace Parmlib member suffix:** Specifies the value that is appended to CTIBBO to form the member name for the CTRACE parmlib member.

After completing the required fields, click **Next**.

14. The next window allows you to enter information about the node (Figure 4-20).

The screenshot shows a configuration window titled "WebSphere Application Server home directory pathname". It contains several input fields and a section for asynchronous administration tasks. The fields are filled with the following values:

- Configuration file system mount point: `/wascsconfig/cscell/csnodeb`
- Directory pathname relative to mount point: `AppServer`
- Node host name: `wtsc62.itso.ibm.com`
- Admin asynch operations procedure name: `CSADMSH`
- Asynchronous Administration Task:
 - User ID: `CSADMIN`
 - UID section:
 - ☒ Allow OS security to assign UID
 - ☐ Assign user-specified UID: `2504`

Figure 4-20 zPMT: Defining Directory, Host and Procedure Names

Complete the following information:

- **Node host name:** Defines the host name where custom node locates.
- **Admin asynch operations procedure name:** Specifies the JCL procedure name of a started task that is launched by way of the START command by application servers or node agents to perform certain asynchronous administrative operations.
- **Asynchronous Administration Task User ID:** Used to run the asynchronous administration operations procedure. The ID must be a member of the WebSphere Application Server configuration group.

After completing the required fields, click **Next**.

15. The next window allows you to define the job names, procedure names, and user IDs to use for each process as follows:

- **Controller process:** Specify the job name in the MVS START command JOBNAME parameter that is associated with the control region. This is the same as the server short name and it cannot be changed during customization. The procedure name is the member name in your procedure library to start the control region. The User ID is the user ID that is associated with the control region.
- **Controller adjunct process:** Specify the name of member in your procedure library that starts the control region adjunct. The User ID is the user ID associated with the control region adjunct.
- **Servant process:** Specify the job name used by WLM to start the servant regions. This is set to the server short name, followed by the letter "S", and it cannot be changed during customization. The procedure name is the member name in your procedure library to start the servant regions. The User ID is the user ID associated with the servant regions.

After completing the required fields, click **Next**.

16. The next window allows you to specify the location daemon settings. The location daemon service is the initial point of client contact in WebSphere Application Server for z/OS. The server contains the CORBA-based location service agent which places sessions in a cell. All RMI/IIOP IORs (for example, enterprise beans) establish connections to the location service daemon first, then forward them to the target application server. Complete the following information:

- **Daemon home directory:** Specifies the directory in which the location service daemon resides. This is set to the configuration file system mount point/Daemon and cannot be changed.
- **Daemon job name:** Specifies the job name of the location service daemon, specified in the JOBNAME parameter of the MVS start command used to start the location service daemon. When configuring a new cell, be sure to choose a new daemon jobname value. A server starts the location service daemon automatically if it is not already running.
- **Procedure name:** Specifies the member name in your procedure library to start the location service daemon.
- **User ID:** Specify the user ID that is associated with the location service daemon.
- **UID:** Specifies the user identifier associated with this user ID. UIDs must be unique numbers within the system.

After completing the required fields, click **Next**.

17. The next window allows you to enter SSL configuration values as follows:

- **Certificate authority keylabel:** Specifies the name that identifies the certificate authority (CA) to be used in generating server certificates.
- **Generate certificate authority certificate:** Selected to generate a new CA certificate. Do not select this option to have an existing CA certificate generate server certificates.
- **Expiration date for certificates:** Used for any X509 Certificate Authority certificates created during customization, as well as the expiration date for the personal certificates generated for WebSphere Application Server for z/OS servers. You must specify this even if you have not selected **Generate Certificate Authority (CA) certificate**.

After completing the required SSL information, click **Next**.

18. The next window allows you to select the user registry to be used for administrative security. You can choose from the following options:

- **z/OS security product option:** This option uses the z/OS system's SAF compliant security product, such as IBM RACF or equivalent, to manage WebSphere Application Server identities and authorization according to:
 - The SAF security database is used as the WebSphere user repository.
 - SAF EJBROLE profiles is used to control role-based authorization, including administrative authority.
 - Digital certificates is stored in the SAF security database.

Note: Select the z/OS security product option if you are planning to use the SAF security database as your WebSphere Application Server registry or if you plan to set up an LDAP or custom user registry whose identities are mapped to SAF user IDs for authorization checking. For this security option, you must decide whether to set a security domain name and choose an administrator user ID and an unauthenticated (guest) user ID.

- **WebSphere Application Server security:** The WebSphere Application Server administrative security option is used to manage the Application Server identities and authorization according to:
 - A simple file-based user registry is built as part of the customization process.
 - Application-specific role binds are used to control role-based authorization.
 - The WebSphere Application Server console users and groups list controls administrative authority.

- Digital certificates are stored in the configuration file system as keystores.

Note: Choose this option if you plan to use an LDAP or custom user registry without mapping to SAF user IDs. (The file-based user registry is not recommended for production use.)

- **No security:** Although it is not recommended, you can disable administrative security. If you choose this security option, there are no other choices to make. Your WebSphere Application Server environment is not secured until you configure and enable security manually. You can enable security manually later using the administrative console or using Jython scripts.

Select an option, and click **Next**.

19. The next window allows you to input the related information about the deployment manager which the custom node is federated into:

- **Configuration file system mount point:** Specifies the read/write file system directory where the application data and environment files are written. This field is not writable here, but was specified earlier on the “System Environment: Configuration file system information” window.
- **Directory path name relative to mount point:** Specifies the relative path name of the directory within the configuration file system in which the deployment manager configuration resides.
- **Deployment manager host name:** Define the host name or IP address of the deployment manager.
- **JMX SOAP port:** Define the port so that the adding node request can use it to connect deployment manager.
- **Deployment manager security is enabled:** Used to indicate whether security enable in the deployment manager. If enabled, the user and password need for federating.
- **Node group name:** Specify the node group into which the node is placed.
- **ORB Listener IP Address:** Set the IP address that the server’s ORB listens to for the incoming IIOP request.
- **ORB port:** Set the port for IIOP request that acts as the bootstrap port for this server and also the port through which the ORB accepts IIOP request
- **ORB SSL port:** Set the port for secure IIOP requests.

Node agent definitions include:

- Server short name
- Server long name
- SOAP JMX connector port
- Node Discovery port
- Node Multicast Discovery port
- Node IPV6 Multicast Discovery port
- High availability manager communication port

20. The next window allows you to tailor the JCL for the customization jobs. Enter a valid job statement for your installation. The profile creation process updates the job name for you in all the generated jobs, so you need not be concerned with that portion of the job statement. If continuation lines are needed, replace the comment lines with continuation lines.

Click **Next**.

21. The last window shows a short summary of the customization, including the profile type and where the generated jobs are stored. To change the characteristics of this profile, click **Back**. Otherwise, click **Create** to generate your z/OS Customization jobs.

When zPMT is finished, it displays a summary window that indicates whether the jobs were created successfully or not. If the jobs were not created, a log file that contains failure information is identified. If successful, the next step in the z/OS customization process is to upload these jobs and the associated instructions to a pair of z/OS partitioned data sets. Click **Finish** to return to the WebSphere for z/OS Customization screen.

22. Now, select the profile that you created and then click **Upload**.

In the upload customization definition window (Figure 4-21 on page 129), enter the target z/OS system using a fully qualified host name or the upload will fail.

Use the **Allocate target z/OS data sets** option to specify whether to allocate the data sets if they do not exist. If the data sets exist and are to be reused, clear the box, and click **Upload**.

Specify the following ftp information and then click **Upload**.

Target z/OS system:	wtsc62.itso.ibm.com
User ID:	ericyu
Password:	••••••
Server port:	21
Timeout (in seconds):	20

The customization data will be uploaded to the following partitioned data sets on the target z/OS s

BBCS6162.CNTL
BBCS6162.DATA

☐ Allocate target z/OS data sets

Optionally, specify both volume and unit for target data sets.

Volume:	
Unit:	

Figure 4-21 Creating a profile: Upload customization definition window

You see a progress information window while the upload is occurring.

23. After you upload the customization profile, follow the instructions in the BBOSSINS member of the CNTL data set. You can view this data set on the host, or select the configuration and use the View button.

These instructions can help you determine what jobs to run, the order in which to run them, and the expected results. It also tells you how to start the environment after you are finished. From this point, the process is identical to the process that you use with the ISPF Customization Dialog.

Import Note: Do not run job BBOWMNAN, which is used to federate the custom node into deployment manager. We federated the custom node after augmenting it for WebSphere Process Server.

24.Repeat this process to create additional custom nodes. In our case, we created two custom nodes. Example 4-1 shows the response file that we used to create the second node.

Example 4-1 Second custom node response file

```
create
cellName=csemtya
hostName=wtsc61.itso.ibm.com
nodeName=csnodea
profileName=default
profilePath=C:\download\AST\workspace\metadata\plugins\com.ibm.ws390.
pmt.config\profiles\csnodea61.1
templatePath=C:\download\AST\workspace\metadata\plugins\com.ibm.ws390
.pmt.config\profileTemplates\zos-managed
zAdjunctProcName=CSCRAA
zAdjunctUid=*
zAdjunctUserid=CSASRU
zAdminAsynchProcName=CSADMSH
zAdminAsynchTaskUid=*
zAdminAsynchTaskUserid=CSADMIN
zAdminSecurityType=websphereForZos
zBBOMSGName=BBCS6161.SBBOMSG
zBbgloadName=BBCS6161.SBBGLOAD
zBboexecName=BBCS6161.SBB0EXEC
zBboloadName=BBCS6161.SBB0LOAD
zBbolod2Name=BBCS6161.SBBOLD2
zBbolpaName=BBCS6161.SBBOLPA
zCaAuthorityExpirationDate=2010/12/31
zCaKeyLabel=WebSphereCA
zCellShortName=CSEMTYA
zConfigHfsName=OMVS.WAS61.CSCELL.CSNODEA.HFS
zConfigHfsPrimaryCylinders=450
zConfigHfsSecondaryCylinders=100
zConfigHfsVolume=TST034
zConfigMountPoint=/wascsconfig/cscell/csnodea
zConfigurationGroup=CSCFG
zConfigurationGroupGID=*
zControlProcName=CSACRA
zControlUid=*
zControlUserid=CSACRU
zCtraceParmLibSuffix=60
zDaemonHomePath=/wascsconfig/cscell/csnodea/Daemon
zDaemonJobName=CSDEMNA
zDaemonProcName=CSDEMNA
```

```

zDaemonUid=*
zDaemonUserid=CSACRU
zDefaultSAFKeyringName=WASKeyring.CSCCELL
zErrorLogstreamName=CSCCELL.ERROR.LOG
zFederateDmaNodeHostName=wtsc61.itso.ibm.com
zFederateDmaPort=9510
zFederateDmaPortType=SOAP
zFederateDmaSecurity=true
zFederateDmaSecurityPassword=csadmin
zFederateDmaSecurityUserID=CSADMIN
zFederateHamCommPort=9525
zFederateJmxSoapConnectorPort=9520
zFederateNodeAgentAfterFederation=true
zFederateNodeDiscoveryPort=9523
zFederateNodeIPv6MulticastDiscoveryPort=9526
zFederateNodeMulticastDiscoveryPort=9524
zFederateOrbListenerHostName=*
zFederateOrbPortName=9521
zFederateOrbSslPortName=9522
zFederateServerName=nodeagent
zFederateServerShortName=CSAGNTA
zFilesystemType=ZFS
zHFSOwnerUID=*
zHFSOwnerUserID=CSOWNER
zJobStatement1=(999,POK),'SC61',CLASS=A,REGION=OM,NOTIFY=&SYSUID
zJobStatement2=/*JOBPARM S=SC61
zJobStatement3=/*
zJobStatement4=/*
zLocalUserGroup=CSGUESTG
zLocalUserGroupGID=*
zNodeGroupName=DefaultNodeGroup
zNodeShortName=CSNODEA
zProclibName=SYS1.PROCLIB
zProductHLQ=BBCS6161
zRunWasFromSteplib=true
zServantGroup=CSSRVG
zServantGroupGID=*
zServantProcName=CSASRA
zServantUid=*
zServantUserid=CSASRU
zSmpePath=/usr/lpp/zWebSphereCS/V6R1
zSysplexName=WTSCPLX1
zSystemName=SC61
zTargetHLQ=BBCS6161.EMPTY
zUserIDHomeDirectory=/var/WebSphere/home
zWasServerDir=AppServer

```

4.5 Configuring the deployment manager node

In this section, we explain the tasks to configure WebSphere Process Server to the deployment manager (DMGR). The tasks are:

1. Run the `zSMPInstall.sh` script against the deployment manager node to prepare the cell for WebSphere Process Server functionality.
2. Prepare `DmgrDB2.rsp`.
3. Run the `zWPSConfig.sh` script to configure WebSphere Process Server in the deployment manager.
4. Create the necessary DB2 data objects using `createDB.sh`
5. Take a backup of the configuration HFS datasets.

4.5.1 Running the `zSMPInstall.sh`

In this step, you add the WebSphere Process Server product profiles into the deployment manager's configuration HFS using the `zSMPInstall` command.

The `zSMPInstall.sh` performs the following tasks:

- ▶ Creates symbolic links from the node's configuration HFS to files in the WebSphere Process Server product HFS.
- ▶ Updates the administrative console to support WebSphere Process Server.
- ▶ Runs `applyPTF` to upgrade the configuration HFS if required.
- ▶ Copies the `zWPSConfig.sh` to `dmgr_home/bin`.

The `zSMPInstall.sh` script takes the following arguments:

-smproot	The WebSphere Process Server product HFS is mount point, for example, <code>/usr/lpp/zWPS/V6R1</code> .
-runtime	The deployment manager's config root directory, for example, <code>/WebSphere/V6R1/DeploymentManager</code> .
-install	The keyword to install the various symbolic links that are associated with WebSphere Process Server into the WebSphere Application Server profile directory.

To use zSMPInstall:

1. Stop the deployment manager.
2. Run the shell script `wps_smpe_root/zos.config/bin/zSMPInstall.sh`, where `wps_smpe_root` is the path to WebSphere Process Server product SMP/E install root. For example:

```
/usr/lpp/zWPSCS/V6R1/zos.config/bin/zSMPInstall.sh -smproot  
/usr/lpp/zWPSCS/V6R1 -runtime /WebSphere/V6R1/DeploymentManager  
-installs
```

Example 4-2 shows the output that you can expect.

Example 4-2 Stdout of zSMPInstall.sh

```
CWPIZ0253I: parsing command arguments...  
CWPIZ0254I: parsing arguments complete  
CWPIZ0255I: setting up configuration...  
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...  
CWPIZ0264I: Configuration Manager update complete
```

4.5.2 Preparing DmgrDB2.rsp

In this step, you prepare a response file that contains the input parameters used by the `zWPSCfg.sh` script to customize the deployment manager configuration and generate DB2 Data Definition Language (DDL) statements that you are used later to define the databases to DB2.

1. Copy the `DmgrDB2.rsp` from the WebSphere Process Server's product HFS, for example `/usr/lpp/zWPSCS/V6R1/zos.config`, into a work directory, for example, `/u/ericyu/wpswork`.

For example, using an OMVS or Telnet session, use the following command:

```
cp /usr/lpp/zWPSCS/V6R1/zos.config/DmgrDB2.rsp /u/ericyu/wpswork
```

2. Edit the new copy of DmgrDB2.rsp file to match the environment you are setting up. Example 4-3 shows the DmgrDB2.rsp file that we used in our cell.

Note: The original version of DmgrDB2.rsp contains extensive comments, but we have removed these in the example and instead added some notes to highlight important points. You need to read the comments and consult the WebSphere Process Server Information Center to understand the meaning of properties that are not self-evident.

Example 4-3 DmgrDB2.rsp

```
JMSUSER=csadmin
JMSPASS=csadmin
CONFIGSERVER=dmgr
DBPRODUCT=DB2UDBOS390_V8_1
DBLOCATION=DB8Q
DBPROPERTIES=/u/db8qu/properties
DBJDBCCLASPPATH=/usr/lpp/db2/d8qg/jcc/classes
DBUSER=csadmin
DBPASS=csadmin
DBHOSTNAME=wtsc61.itso.ibm.com
DBSERVERPORT=38310
SQLID=CSCELL
SQLDB=CSCELLDB
SQLSTO=CSDBSTO
augment
profileName=default
profilePath=/wascsconfig/cscell/dmgr/DeploymentManager/profiles/default
templatePath=/wascsconfig/cscell/dmgr/DeploymentManager/profileTemplate
s/dmgr.wbiserver
cellName=cscell
nodeName=cscdm
enableAdminSecurity=true
adminUserName=$JMSUSER
adminPassword=$JMSPASS
serverName=$CONFIGSERVER
dbCreateNew=false
dbDelayConfig=true
configureScaSecurity=true
scaSecurityUserId=$JMSUSER
scaSecurityPassword=$JMSPASS
configureAppScheduler=false
appSchedulerServer=$CONFIGSERVER
dbName=$SQLDB
dbStorageGroup=$SQLSTO
```

```
dbType=$DBPRODUCT
dbConnectionLocation=$DBLOCATION
dbJDBCProperties=$DBPROPERTIES
dbJDBCClasspath=$DBJDBCCLASSPATH
dbUserId=$DBUSER
dbPassword=$DBPASS
dbHostName=$DBHOSTNAME
dbServerPort=$DBSERVERPORT
dbSchemaName=$SQLID
```

The following notes refer to the DmgrDb2.rsp response file in Example 4-3:

- ▶ The CONFIGSERVER global variable must be set to the long name of the Network Deployment server you are configuring (see Figure 4-8 on page 107).
- ▶ The DBLOCATION global variable is the location name (not the subsystem name) of your DB2 system.
- ▶ Do not change the profile name from default. Only default is supported in WebSphere Application Server for z/OS.
- ▶ The profilePath is the path to the /profiles/default directory under *dmgr_home*. Note that you need to code this property on one line, but in the example it is split across two lines in order to fit on the page.
- ▶ The cellName, nodeName, and serverName tell the zWPSConfig.sh which node to configure. You should make sure the cellName and nodeName are long names you specified when you created the profiles.
- ▶ You must indicate the version of DB2 you are using with DBPRODUCT. Note that if you are using DB2 V8.1, but it is running in compatibility mode with DB2 7.1, then you need to specify DB2UDBOS390_V7_1.
- ▶ SQLID is the schema name of database. SQLDB is the name of database and SQLSTO is the storage group name of database.
- ▶ Set dbCreateNew=false because you create the storage group and databases manually.
- ▶ Set dbDelayConfig=true to create the database tables later.

Warning: The DmgrDB2.rsp response file contains the minimum number of properties you need in the response file in order to configure the deployment manager. Do not delete properties from the sample DmgrDB2.rsp even if you want to allow them to take default values. Removing properties can cause the script to fail.

4.5.3 Running the zWPSConfig.sh

Run the configuration script `zWPSConfig.sh` directly from UNIX System Services with the absolute path of your edited response file. For example:

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

In this command, *working_directory_path* is the location of the `DmgrDB2.rsp` file.

When the script has completed, review the messages that are written to the console. If the script runs successfully, no error messages are displayed and the informational messages state “augmenting profile(s) complete” (see Example 4-4).

Example 4-4 Stdout of zWPSConfig.sh

```
CWPIZ0253I: parsing command arguments...
CWPIZ0254I: parsing arguments complete
CWPIZ0255I: setting up configuration...
CWPIZ0256I: set up configuration complete
CWPIZ0265I: augmenting profile(s)...
INSTCONFSUCCESS: Profile augmentation succeeded.
CWPIZ0267I: augmenting profile(s) complete
```

4.5.4 Creating DB2 database objects using the createDB.sh script

Next, use `createDB.sh` script to create a database and populate it with 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.

The `createDB.sh` script is located in *wps_smpe_root/zos.config/samples*. To use this script:

1. Copy the sample `createDB.sh` file to your working directory. For example:

```
cp /usr/lpp/zWPS/V6R1/zos.config/samples/createDB.sh
/u/ericyu/wpswork
```

2. Assign the appropriate permissions to the copy of the `createDB.sh` file:

```
chmod 755 createDB.sh
```

3. Customize the parameters in createDB.sh file as required by your system.
Example 4-5 shows the parameters for our installation.

Example 4-5 Customized parameters of createDB.sh

```
WAS_HOME=/wascsconfig/cscell/dmgr/DeploymentManager
LIBPATH=/usr/lpp/db2/d8qg/jcc/lib:$LIBPATH
export LIBPATH
```

```
SCRIPTHOME=`pwd`
export SCRIPTHOME
SKIP=0
INVALIDoption=0;
```

```
DBC_CREATE=true
DBUSESQLID=true
```

```
DBWPSCreate=true
DBBPCCreate=true
DBCEICreate=true
DBSIBCreate=true
DBGenerate=true
```

```
DBPREFIX=CS
DBSCOPE=S61
```

```
DBJDBCClasspath=/usr/lpp/db2/d8qg/jcc/classes
DBUSER=csadmin
DBPASSWORD=csadmin
DBJDBCProperties=/u/ericyu/wpswork
DBConnectionLocation=DB8Q
DBDelayConfig=false
DBHostName=wtsc61.itso.ibm.com
DBServerPort=38310
DBVOLUMES=TOTDB9
DBVCAT=DB8QU
```

```
CEIBP4K=BP0
CEIBP8K=BP8K0
CEIBP16K=BP16K0
CEISIZE=10
```

```
DBSTO=${DBPREFIX}DBSTO
DBSCHEMA=${DBPREFIX}CELL
DBNAME=${DBPREFIX}CELLDB
```

In this example:

- *WAS_HOME* is set to the mount point of deployment manager configuration system
 - The values of *DBSTO*, *DBSCHEMA*, and *DBNAME* must be the same values that you specified for *SIBSTO*, *SIBID*, *SQLDB* in the *DmgrDB2.rsp* file.
 - *DBSCHEMA* is the schema name of database schema. *DBNAME* is the name of database and *DBSTO* is the storage group name of database.
4. Run the customized *createDB.sh* script from UNIX System Services. For example:
- ```
createDB.sh +All
```

**Note:** For a list of parameters that you must customize and syntax, see [http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.z.610.doc/doc/rins\\_zos\\_createdbscript.html](http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.z.610.doc/doc/rins_zos_createdbscript.html)

#### 4.5.5 Backing up the WebSphere configuration HFS data sets

Take a backup of the configuration HFS datasets before proceeding. Back up both the deployment manager configuration HFS and the custom node configuration HFS.

### 4.6 Configuring a custom node

To configure WebSphere Process Server in the un-federated custom node, follow these steps:

1. Run the *zSMPInstall.sh* script against the custom node to prepare the node for WebSphere Process Server functionality.
2. Prepare the *ManagedDB2.rsp* response file.
3. Run the *zWPSConfig.sh* script to configure WebSphere Process Server in the custom node.
4. Federate the custom node.
5. Configure and federate any additional custom nodes for the cell.
6. Take a backup of the configuration HFS datasets.

## 4.6.1 Running the zSMPInstall script

This step is very similar to the step performed against the deployment manager node. You must execute the zSMPInstall.sh script from the custom node's *wps\_smpe\_root/bin* directory, and you must set the **-runtime** property to the custom node's *app\_server\_root/bin*.

Example 4-6 shows an example of running the zSMPInstall script.

*Example 4-6 Execute zSMPInstall.sh*

---

```
/usr/lpp/zWPSCS/V6R1/zos.config/bin/zSMPInstall.sh -smproot
/usr/lpp/zWPSCS/V6R1 -runtime /wascconfig/cscell/csnodb/AppServer
-install
```

---

The zSMPInstall.sh script takes the following arguments:

|                 |                                                                                                                                                     |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-smproot</b> | The mount point for the WebSphere Process Server product HFS, for example, /usr/lpp/zWPSCS/V6R1.                                                    |
| <b>-runtime</b> | The custom node configuration root directory, for example, /wascconfig/cscell/csnodb/AppServer.                                                     |
| <b>-install</b> | The keyword to install the various symbolic links associated with WebSphere Process Server into the WebSphere Application Server profile directory. |

Run zSMPInstall.sh against the custom node and check the output (shown in Example 4-7) carefully before proceeding.

*Example 4-7 The stdout output from a successful execution of zSMPInstall.sh*

---

```
CWPIZ0253I: parsing command arguments...
CWPIZ0254I: parsing arguments complete
CWPIZ0255I: setting up configuration...
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...
CWPIZ0264I: Configuration Manager update complete
```

---

## 4.6.2 Preparing ManagedDB2.rsp

This step is similar to the step you performed against the deployment manager node. However, for a custom node, you use the ManagedDB2.rsp response file to configure the custom node.

1. Copy the ManagedDB2.rsp from the WebSphere Process Server's product HFS (for example, /usr/lpp/zWPSCS/V6R1/zos.config) into a work directory, for example, /u/ericyu/wpswork.

The following command use an OMVS or Telnet session:

```
cp /wps_smpe_root/zos.config/ManagedDB2.rsp /<wpswork>
```

2. Edit the new copy of ManagedDB2.rsp file to match the environment you are setting up. Example 4-8 shows the ManagedDB2.rsp file that we used in our custom node.

*Example 4-8 ManagedDB2.rsp*

```
#####
#GLOBAL PROPERTIES
#####JMS
USER=csadmin
JMSPASS=csadmin
DBPRODUCT=DB2UDBOS390_V8_1
DBLOCATION=DB8Q
DBPROPERTIES=/u/db8qu/properties
DBJDBCCLASPPATH=/usr/lpp/db2/d8qg/jcc/classes
DBUSER=csadmin
DBPASS=csadmin
DBHOSTNAME=wtsc61.itso.ibm.com
DBSERVERPORT=38310
SQLDB=CSCELLDB
SQLSTO=CSDBSTO

#####
Common Properties
#####aug
ment
#####
Profile name
#####
profileName=default

#####
Profile path
#####
```

```

profilePath=/wascsconfig/cscell/csnodeb/AppServer/profiles/default
#####
Template path
templatePath=/wascsconfig/cscell/csnodeb/AppServer/profileTemplates/man
aged.wbiserver

#####
Cell name
cellName=cseptyb

#####
Node name
nodeName=csnodeb

#####
Create new or use existing database
dbCreateNew=false

#####
Delay Configuration of the database table and data source def
dbDelayConfig=true

#####
WBI Core Properties
#####con
figureScaSecurity=true
scaSecurityUserId=$JMSUSER
scaSecurityPassword=$JMSPASS
configureAppScheduler=false
dbName=$SQLDB
dbStorageGroup=$SQLSTO
dbType=$DBPRODUCT
dbConnectionLocation=$DBLOCATION
dbJDBCProperties=$DBPROPERTIES
dbJDBCClasspath=$DBJDBCCLASSPATH
dbPassword=$DBPASS
dbHostName=$DBHOSTNAME
dbServerPort=$DBSERVERPORT

```

---

In this example:

- The DBLOCATION global variable is the location name (not the subsystem name) of your DB2 system.
- Do not change the profile name from default. Only default is supported in WebSphere Application Server for z/OS.

- The *profilePath* is the path to the /profiles/default directory under *app\_server\_root*.  
Notice that the cell name in the path is *cemptyb* rather than *cscell* because the node has not yet been federated into the cell.
- These properties tell the *zWPSConfig.sh* which node to configure. The *cellName* is *not* <xx>cell because you are augmenting the custom node, which is not yet part of the cell. For our CS cell, the cell name of the custom node is *csnodeb*. Similarly, there must not be a *serverName* property in this response file because this is a custom node.
- You must indicate the version of DB2 you are using with *DBPRODUCT*. Note that if you are using DB2 V8.1, but it is running in compatibility mode with DB2 7.1, then you need to specify *DB2UDBOS390\_V7\_1*.
- Set *dbCreateNew=false* because you have already manually created the storage group and databases.

**Warning:** The *ManagedDB2.rsp* response file contains the minimum number of properties you need in the response file in order to configure the custom node.

Do not delete properties from the sample *ManagedDB2.rsp* even if you want to allow them to take default values. Removing properties can cause the script to fail.

### 4.6.3 Running *zWPSConfig.sh*

You can run *zWPSConfig.sh* from a command line in OMVS or telnet. Example 4-9 shows an example of running *zWPSConfig.sh* on a custom node.

*Example 4-9 Running zWPSConfig.sh on a custom node*

---

```
export LIBPATH=/usr/lpp/db2/d8qg/jcc/lib:$LIBPATH
/wascsconfig/cscell/csnodeb/AppServer/bin/zWPSConfig.sh -response
/u/ericyu/wpswork/ManagedDB2.rsp -augment
```

---

Run *zWPSConfig.sh* from *app\_server\_root/bin*, for example:

```
/wascsconfig/cscell/csnodeb/AppServer/bin
```

Check the output from the `zWPSConfig.sh`. The `zWPSConfig.sh` takes anywhere between 30 and 75 minutes to complete depending on your system's performance. The output looks similar to that shown in Example 4-10.

*Example 4-10 Sample output from the `zWPSConfig.sh` script. for the custom node*

---

```
CWPIZ0253I: parsing command arguments...
CWPIZ0254I: parsing arguments complete
CWPIZ0255I: setting up configuration...
CWPIZ0256I: set up configuration complete
CWPIZ0265I: augmenting profile(s)...
INSTCONFSUCCESS: Profile augmentation succeeded.
CWPIZ0267I: augmenting profile(s) complete
```

---

#### 4.6.4 Federating a custom node

After you have augmented the deployment manager and custom node for WebSphere Process Server, you can federate the custom node into the cell. There are two methods of federating a custom node into the deployment manager:

- ▶ Run the `BBOWMNAN` job created when you defined the profile
- ▶ Run `addNode.sh`

Before you begin to federate, you must make sure the deployment manager is running.

Example 4-11 shows the `addNode` command.

*Example 4-11 Run `addNode.sh` to federate*

---

```
app_server_root/profiles/default/bin/addNode.sh DM_HOST DM_PORT
-includeapps -nodegroupname DefaultNodeGroup -username xxADMIN
-password password -nodeagentshortname xxAGNT1 -repalcelog
```

---

In this example:

- ▶ `DM_HOST` is the host name of deployment manager.
- ▶ `DM_PORT` is the `soap_connector_address` of the deployment manager. If you do not know the port number, you can login to the administrative console and select **System Administration** → **Deployment Manager** → **Ports**.
- ▶ `-username` and `-password` are needed if administrative security is enabled.
- ▶ `-nodeagentshortname` is the short name of node agent. such as `CSAGNTB` in this case.

The output looks similar to that shown in Example 4-12.

*Example 4-12 Stand output of running addNode.sh*

---

```
CSADMIN @
SC62:/wascsconfig/cscell/csnodeb/AppServer/profiles/default/bin>ericAdd
Node.sh
ADMU0116I: Tool information is being logged in file
/wascsconfig/cscell/csnodeb/AppServer/profiles/default/logs/addNode.log
CWPki0309I: All signers from remote keystore already exist in local
keystore.
ADMU0001I: Begin federation of node csnodeb with Deployment Manager at
wtsc61.itso.ibm.com:9510.
ADMU0001I: Begin federation of node csnodeb with Deployment Manager at
wtsc61.itso.ibm.com:9510.
ADMU0009I: Successfully connected to Deployment Manager Server:
wtsc61.itso.ibm.com:9510
ADMU0507I: No servers found in configuration under:

/wascsconfig/cscell/csnodeb/AppServer/profiles/default/config/cells/cse
mptyb/nodes/csnodeb/servers
ADMU2010I: Stopping all server processes for node csnodeb
ADMU0024I: Deleting the old backup directory.
ADMU0015I: Backing up the original cell repository.
ADMU0012I: Creating Node Agent configuration for node: csnodeb
ADMU0014I: Adding node csnodeb configuration to cell: cscell
ADMU0016I: Synchronizing configuration between node and cell.
ADMU0018I: Launching Node Agent process for node: csnodeb
ADMU0020I: Reading configuration for Node Agent process: nodeagent
ADMU0022I: Node Agent launched. Waiting for initialization status.
ADMU0030I: Node Agent initialization completed successfully. Process id
is:000000800000008b
ADMU0308I: The node csnodeb and associated applications were
successfully added to the cscell cell.
```

---

#### 4.6.5 Configuring and federating a secondary custom node

Depending on your topology, you might need to configure more than one custom node. In our case, we configured and federated two custom nodes.

### 4.6.6 Backing up the WebSphere configuration HFS data sets

Take another backup of the configuration HFS datasets before proceeding. Back up both the deployment manager's configuration HFS and the custom node's configuration HFS at the same time.

## 4.7 Creating a WebSphere Process Server cluster

Now that you have the augmented node federated into the cell, you must define a cluster with at least one server, where you run a WebSphere Process Server or WebSphere Enterprise Service Bus workload.

Clusters are created from the administrative console. Servers are added to the cluster when you define it. The names and ports that you use when creating the server should conform to your naming convention. If you are following the guidance in *WebSphere z/OS V6 -- WSC Sample ND Configuration* and using the spreadsheet that is provided with that document to plan your naming convention, the worksheet for the stand-alone server contains the names and ports that you use when creating the server.

The following tasks define a WebSphere Process Server-enabled cluster or server:

1. Add DB2 libraries to the STEPLIB of the servant and Adjunct region JCL.
2. Add a cluster with two servers using the WebSphere Process Server administrative console.
3. Configure Service Component Architecture.
4. Configure Business Process Manager and Human Task Manager.
5. Configure Common Event Infrastructure.

### 4.7.1 Adding DB2 libraries to the Servant and Adjunct JCL

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

If you do not have STEPLIB set correctly, you receive the error shown in Example 4-13, issued in the adjunct caused by the service integration bus failing during initialization. You can see similar messages in the servant region if the servant cannot load the DB2 programs in SDSNLOD2.

*Example 4-13 DB2 Error message when test data source*

---

```
error message: BB000220E:
[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:
Chapter 11. Troubleshooting 431
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
```

---

Locate the JCL for your WebSphere cell in your system PROCLIB and change the members with the Z suffix for the servant and the adjunct. For our CS cell, for example, we added the STEPLIBs for the DB2 libraries to CSASRBZ (servant) and CSCRABZ (adjunct) for nodeb and CSDSRZ (servant) for DM.

## 4.7.2 Creating a cluster/server

If you want to create a cluster, you start by defining the cluster using the WebSphere Process Server administrative console. During this process, you create the first WebSphere Process Server *server* by adding two cluster members to the cluster.

To create a cluster with two cluster member (server) in it:

1. Log in to the administrative console and navigate to **Servers** → **Clusters**. Click **New**.
2. Enter a cluster name (Figure 4-22 on page 147). It is a very good idea to choose a cluster name that is in lower case and prefixed by your two-character cell prefix.

**Note:** The cluster short name is important because RACF profiles in class CBIND and SERVER control access to the members of a cluster.

You should have defined some RACF CBIND and SERVER profiles for your cell when you defined your deployment manager.

If you followed the guidance in the *WebSphere z/OS V6 -- WSC Sample ND Configuration*, you created generic RACF profiles in the form CB.\*.MK\*, which are valid provided that you used a cluster name that was prefixed by your two-character cell prefix. If you chose some other name, you need to define some CBIND and SERVER class profiles that match the cluster name you have chosen.

Click **Next**.

→ Step 1: Enter basic cluster information

Step 2: Create first cluster member

Step 3: Create additional cluster members

Step 4: Summary

**Enter basic cluster information**

\* Cluster name  
CSSR00

Short name  
CSSR00

☒ Prefer local. Specifies whether enterprise bean requests will be routed to the node on which the client resides when possible.

☐ Configure HTTP session memory-to-memory replication

Next Cancel

Figure 4-22 Choosing a cluster name

3. Enter a name for the new server that is created in the cluster (Figure 4-23 on page 148).

The name that you specify here is the long name for the server. If you follow the naming convention in WP100653 this is the same as the short name for the server but in lower case.

Select **Default application server template** and expand the drop-down list box to select **Default process server**, as shown in Figure 4-23.

Use the drop-down list box in the Select node field to choose the node in which you want to define the server, as shown in Figure 4-23.

Click **Next**.

Step 1: Enter basic cluster information

→ **Step 2: Create first cluster member**

Step 3: Create additional cluster members

Step 4: Summary

### Create first cluster member

The first cluster member determines the server settings for the cluster members. A server configuration template is created from the first member and stored as part of the cluster data. Additional cluster members are copied from this template.

\* Member name  
CSSR01A

Select node  
csnodea(ND 6.1.0.13)

Short name  
CSSR01A

\* Weight  
2 (0..20)

☒ Generate unique HTTP ports

**Select basis for first cluster member:**

☒ Create the member using an application server template.  
defaultProcessServerZOS

☐ Create the member using an existing application server as a template.  
cscell/csnodea(ND 6.1.0.13)/CSSR01A

☐ Create the member by converting an existing application server.  
(none)

☐ None. Create an empty cluster.

Figure 4-23 Setting server name and choosing the process server template

4. You are given another blank form in which you can add another server to the cluster (Figure 4-24). Enter the second server name in the Member name field and select the node that hosts the server.

Click **Add member**. The new server is listed at the bottom of the panel. Click **Next**.

Step 1: Enter basic cluster information

Step 2: Create first cluster member

→ Step 3: Create additional cluster members

Step 4: Summary

### Create additional cluster members

Enter information about this new cluster member, and click Add Member to add this cluster member to the member list. A server configuration template is created from the first member and stored as part of the cluster data. Additional cluster members are copied from this template.

\* Member name

Select node

Short name

\* Weight  
 (0..20)

☒ Generate unique HTTP ports

Use the Edit function to edit the properties of a cluster member that is already included in this list. Use the Delete function to remove a cluster member from this list. You are not allowed to edit or remove the first cluster member or an already existing cluster member.

☒ ☐

| Select                   | Member name | Nodes   | Version                                            | Weight |
|--------------------------|-------------|---------|----------------------------------------------------|--------|
|                          | CSSR01A     | csnodea | Business Process Choreographer 6.1.0.0 ND 6.1.0.13 | 2      |
| <input type="checkbox"/> | CSSR01B     | csnodeb | Business Process Choreographer 6.1.0.0 ND 6.1.0.13 | 2      |

Figure 4-24 Create additional cluster members

5. On the Summary panel, check the details and click **Next**.
6. Save the configuration changes. The cluster is listed as shown in Figure 4-25.








|                                                                                                                                                                                                                                                                                                                                         |                                                                                        |                                                                                           |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| New Delete Start Stop Ripplestart ImmediateStop                                                                                                                                                                                                                                                                                         |                                                                                        |                                                                                           |
|     |                                                                                        |                                                                                           |
| Select                                                                                                                                                                                                                                                                                                                                  | Name  | Status  |
| <input type="checkbox"/>                                                                                                                                                                                                                                                                                                                | <a href="#">CSSR00</a>                                                                 |          |
| Total 1                                                                                                                                                                                                                                                                                                                                 |                                                                                        |                                                                                           |

Figure 4-25 Click the new cluster to update it

### 4.7.3 Configuring Service Component Architecture

To configure Service Component Architecture support from the administrative console, use the following steps:

1. Select **Server** → **Cluster**. Click the cluster name to open the configuration page. In the Business Integration section, select **Service Component Architecture**, as shown in Figure 4-26.

**General Properties**

\* Cluster name  
CSSR01

Short name  
CSSR01

Unique ID  
C1FBEB9414990385000003A400000001090C0420

Bounding node group name  
DefaultNodeGroup ▼

☒ Prefer local

☐ Enable failover of transaction log recovery

Apply OK Reset Cancel

**Container Settings**

- Business Process Choreographer
- Container Settings

**Cluster messaging**

- Messaging engines

**Business Integration**

- Business Integration Configuration
- Service Component Architecture**
- Common Event Infrastructure
- Business Process

Figure 4-26 Configuration Tab

Figure 4-27 shows the SCA configuration page.

☒ Support the Service Component Architecture components

**Bus Member Location**

☒ Local

☐ Remote

WebSphere:cluster=CSSR01

**System Bus Member**

System bus destinations support the asynchronous communication of Service Oriented Architecture components with each other.

| Database Instance | Schema | Create Tables            | User name | Password |
|-------------------|--------|--------------------------|-----------|----------|
| DB8Q              | S      | <input type="checkbox"/> | csadmin   | *****    |

**Application Bus Member**

Application bus destinations support the asynchronous communication of WebSphere Business components.

☒ Configure the application bus

| Database Instance | Schema | Create Tables            | User name | Password |
|-------------------|--------|--------------------------|-----------|----------|
| DB8Q              | A      | <input type="checkbox"/> | csadmin   | *****    |

Figure 4-27 Service Component Architecture Configuration Wizard

2. In this wizard:
  - a. Select **Support the Service Component Architecture components** to activate the Bus Member Location section.
  - b. Specify whether the SCA system bus (and, optionally, the SCA application bus) destinations are hosted on the local deployment target or on a remote target as follows:
    - Select **Local** if you want to create and host SCA applications and their required messaging engines and JMS queue destinations on the current cluster or server.
    - Select **Remote** to host SCA applications on the local cluster or server while using a remote cluster or server to host the JMS queue destinations and messaging engines.

**Note:** If you use the Browse Deployment Target page to add a new target to the list and navigate away from the Service Component Architecture page before completing your SCA configuration, that target is removed from the list.

- c. If you select a remote deployment target, the appropriate information for that target is shown in the System Bus Member panel. If the application bus is enabled on the remote deployment target, the page also updates the table in the Application Bus Member panel. If the application bus is not enabled on the deployment target, you can enable it.

Specify the properties of the data source used by the SCA system bus. This panel is active whenever you are creating a new SCA configuration for the cluster or server and when you are editing the system bus data source properties for a previous SCA configuration. Many fields in this panel contain default values based on the WebSphere Process Server and WebSphere Enterprise Service Bus Common database (by default, WPRCSDB) configured on the deployment target you have selected. You can accept these default values or edit them.

- d. Select **Edit** in the System Bus Member section to specify the database information for SCA. You have the option to provide the following information:
    - **Database Instance:** Specifies the database instance used for this data source. The value must be the name of an existing database instance.
    - **Schema:** Enter the name of the database schema that contains the tables for the system bus data source. This field is required if you are creating a new data source with a database that supports schema names. Each messaging engine stores its resources, such as tables, in a single schema. Each database schema is used by one messaging

engine only. Although every messaging engine uses the same table names, its relationship with the schema gives each messaging engine exclusive use of its own tables.

- **Create tables:** Do not select this option.

**Note:** Use the `wps_smpe_root/zos.config/samples/createDB.sh` to create these tables.

- **User name and password:** Used to connect to the system bus data source.
  - **Server:** Specify the name of the database server used by the system bus.
  - **Provider:** Specify the database provider type used to create the messaging resources for the system bus.
- e. Select **Test Connection** to verify that the data source can contact and authenticate with the database. If a component manages the data source, this test also verifies whether the data can be reached from the configured scope and, if applicable, whether the schema is configured correctly.
- f. Select **Edit** in the System Application Bus Member section to specify the database information for SCA. The configuration properties are similar to those for the system bus, but in addition, you have the option to select **Enable the WebSphere Business Integration Adapter components** if you plan to deploy SCA applications that use WebSphere Business Integration Adapters.

**Note:** Use the `wps_smpe_root/zos.config/samples/createDB.sh` to create these tables.

3. After you input value, click **OK** to finish configuring Service component architecture.

**Note:** For more details, see the WebSphere Process Server V6.1 Information Center:

[http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.websphere.wesb610.zseries.doc/doc/clusters/tadm\\_configscasupport.html](http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.websphere.wesb610.zseries.doc/doc/clusters/tadm_configscasupport.html)

**Note:** You might encounter a message engine failure with the following error:

“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.b.SqlException: UNSUCCESSFUL EXECUTION CAUSED BY AN UNAVAILABLE RESOURCE. REASON 00E7009A, TYPE OF RESOURCE 100, AND RESOURCE NAME TEMP DATABASE“.

This problem indicates that either the TEMP database is not defined or that its page size is not suitable for your use. Define a suitable TEMP database for the DB2 Universal JDBC driver so that it works correctly with DB2 for z/OS. Work with the database administrator to ensure that a TEMP database is created for each member of a data sharing group. The TEMP database has a 32 KB page size by defining it in a 32 K buffer pool.

Refer to the following resource on Code 00E7009A for further guidance on this problem:

<http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2.doc.msgs/msgs/code00e7009a.html>

## 4.7.4 Configuring Common Event Infrastructure

To configure the common event infrastructure for a cluster:

1. In the Business Integration section of the cluster's configuration page, expand Common Event Infrastructure and click **Common Event Infrastructure Server**, as shown in Figure 4-28.

**General Properties**

☒ Enable the event infrastructure server

**Common Event Infrastructure Event Database**

The Common Event infrastructure event database stores Common Base Events for historic data

Edit ... Test Connection

| Database Instance | Schema  | Create Tables            | User name | Password |
|-------------------|---------|--------------------------|-----------|----------|
| DB8Q              | CSCCELL | <input type="checkbox"/> | csadmin   | *****    |

**Common Event Infrastructure Bus Member Location**

☒ Local

☐ Remote

**Common Event Infrastructure Bus Member**

Common Event Infrastructure bus destination support the asynchronous transmission and distribution

Edit ... Test Connection

| Database Instance | Schema | Create Tables            | User name | Password |
|-------------------|--------|--------------------------|-----------|----------|
| DB8Q              | C      | <input type="checkbox"/> | csadmin   | *****    |

Figure 4-28 CEI configuration

2. Select **Enable the event infrastructure server**.
3. Specify the database that you use to store events emitted by the CEI service on your server. Select **Edit** to update the configuration.

Common event database fields include:

- **Database Instance:** Name of the database you are using for storing CEI-emitted events.
- **Schema:** Not supported for the CEI event database. The CEI server does not support dynamic database schemas for event databases on distributed platforms.

To support several CEI server configurations, you need to configure a different event database instance for each additional CEI server.

Alternatively, you can set up different event database users for each CEI server configuration for databases that map user IDs to schemas.

- **Create Tables:** Do not select this option.

**Note:** Use the `wps_smpe_root/zos.config/samples/createDB.sh` to create these tables.

- **User name:** User ID required by the database.
- **Password:** Password for the user ID required by the database.
- **Server:** Fully qualified IP address or name and, optionally, the port used by the database
- **Provider:** The database vendor. If the CEI has not been configured, you must select the database vendor. After you configure the CEI, you cannot edit the provider field.

Select **Test connection** button to test the connection.

4. Select **Edit** to update the database configuration for the messaging infrastructure as follows:
  - **Database Instance:** Name of the database that you use for CEI messaging infrastructure.
  - **Schema:** Name of the schema you use with the CEI messaging infrastructure database.
  - **Create Tables:** Do not select this option.

**Note:** Use the `wps_smpe_root/zos.config/samples/createDB.sh` to create these tables.

- **User name:** User ID required by the database.
- **Password:** Password for the user ID required by the database.
- **Server:** Fully qualified IP address or name and, optionally, the port used by the database
- **Provider:** The database vendor or file store. You must select a database vendor from the menu if the CEI has not been configured. After you configure the CEI, you cannot edit the provider field.

Select **Test connection** to test the connection.

5. After you input value, click **OK** to finish configuring Common event infrastructure.

**Note:** For more details, see the following article in the WebSphere Process Server Information Center:

[http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.cei.z.610.doc/doc/tcei\\_configceiconsle.html](http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.cei.z.610.doc/doc/tcei_configceiconsle.html)

## 4.7.5 Configuring the Business Flow and Human Task Managers

**Note:** For more information, see the following article about Business Process Choreographer configuration in the WebSphere Process Server Information Center:

[http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.bpc.z.610.doc/doc/bpc/help\\_bpcconfiguration/bpc\\_config.html](http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.bpc.z.610.doc/doc/bpc/help_bpcconfiguration/bpc_config.html)

To configure the Business Flow and Human Task Managers:

1. In the Business Integration section of the cluster's configuration page expand Business Process Choreographer and select **Business Flow Manager**.
2. In the containers section of this page, select **Business Process Choreographer Containers**.
3. The configuration page contains properties for the databases to be used by the business flow and human task managers. Select **Edit** to update the configuration.

The following properties are for the Business Process Choreographer database:

- **Database Instance:** The name of the database used by the Business Flow Manager and Human Task Manager.
- **Schema Name:** The name of the schema to be used. You only need to specify a schema name if you want to use your own schema instead of the default schema.
- **Create Tables:** Do not select this option.

**Note:** Use the `wps_smpe_root/zos.config/samples/createDB.sh` to create these tables.

- **User Name:** A user ID that has the authority to connect to the database and to modify the data.
- **Password:** The password for the data source user ID.
- **Server:** The address of the database server. Specify either the host name or the IP address, and the port number.
- **Provider:** The JDBC provider for Business Process Choreographer.

| Database Instance | Schema Name | Create Tables            | User Name | Password | Server |
|-------------------|-------------|--------------------------|-----------|----------|--------|
| DB8Q              | CSCELL      | <input type="checkbox"/> | csadmin   | *****    | wtsc61 |

Buttons: Edit..., Test Connection...

Other sections: Human Task Manager Mail Session, Security, State Observers, SCA Bindings

Figure 4-29 BPC Data source configuration

Other items that you can configure on this page include:

- E-mail service for escalation of human tasks. For our example, we did not enable this function.
- Specify the mapping of functional roles to user IDs and groups, and the authentication credentials that are necessary for Business Process Choreographer as shown in Figure 4-30.

▼

Security

| Role          | User                                 | Group                              | Description                                                                                                                                     |
|---------------|--------------------------------------|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| Administrator | <input type="text" value="csadmin"/> | <input type="text" value="CSCFG"/> | User name(s) and/or group name(s) for the business process engines assigned to this role have all privileges.                                   |
| Monitor       | <input type="text" value="csadmin"/> | <input type="text" value="CSCFG"/> | User name(s) and/or group name(s) for the business process engines to this role can view the properties of all of the business process engines. |

| Authentication                 | User                                 | Password                               | Confirm Password                       | Description                                                           |
|--------------------------------|--------------------------------------|----------------------------------------|----------------------------------------|-----------------------------------------------------------------------|
| JMS Authentication             | <input type="text" value="csadmin"/> | <input type="password" value="*****"/> | <input type="password" value="*****"/> | Authentication user for business process engines on the system.       |
| JMS API Authentication         | <input type="text" value="csadmin"/> | <input type="password" value="*****"/> | <input type="password" value="*****"/> | Authentication for business process engines for process asynchronous. |
| Escalation User Authentication | <input type="text" value="csadmin"/> | <input type="password" value="*****"/> | <input type="password" value="*****"/> | Authentication for business process engines for process asynchronous. |

Figure 4-30 Security settings for the Business Process Choreographer

- Enable audit logging and Common Event Infrastructure (CEI) logging for the Business Flow Manager and Human Task Manager as shown in Figure 4-31.

▼

State Observers

| Logging                             | Business Flow Manager               | Human Task Manager                  |
|-------------------------------------|-------------------------------------|-------------------------------------|
| Audit Logging                       | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Common Event Infrastructure Logging | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

Figure 4-31 Enable audit and CEI logging

- Set the context root for the Web services API of the SCA bindings as shown in Figure 4-32.

| SCA Bindings     |              |                                |                      |
|------------------|--------------|--------------------------------|----------------------|
| Host             | Context Root | Relative Path                  | Description          |
| http://host:port | IF_CSSR01    | /sca/com/ibm/bpe/spi/sca/BFMWS | Business Flow Manage |
| http://host:port | /HTMIF_CS    | /sca/com/ibm/bpe/spi/sca/HTMWS | Human Task Manage    |

Figure 4-32 Context root settings

- The data source for the Business Process Choreographer messaging engine.  
If **Use the default configuration** is selected, the current configuration settings of the SCA messaging engine are used. To use different settings, clear the check box to enable the other fields in this section.

☐ Use the default configuration

**Bus Member Location**

☒ Local  
☐ Remote

(none)

| Database Instance | Schema Name | Create Tables            | User Name | Password | Server   |
|-------------------|-------------|--------------------------|-----------|----------|----------|
| DB8Q              | B           | <input type="checkbox"/> | csadmin   | *****    | wtsc61.i |

Figure 4-33 Messaging engine settings



## Managing the runtime environment

WebSphere Process Server administrative tools are provided by the underlying WebSphere Application Server Network Deployment environment. In this chapter, we show common methods for managing the WebSphere Process Server environment using some of the administrative tools that are available.

The information that we provide here is by no means a complete guide to these tasks. You can use this information to get started. We recommend that you also use the following resources:

- ▶ *WebSphere Application Server for i5/OS Handbook: Version 6.1*, SG24-7221
- ▶ *WebSphere Application Server V6.1: System Management and Configuration*, SG24-7304
- ▶ WebSphere Application Server V6.1 information center.  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp>
- ▶ WebSphere Process Server V6.1 information center.  
[http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/welcome\\_top\\_wps.htm](http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/welcome_top_wps.htm)

- ▶ WebSphere Enterprise Service Bus (WebSphere ESB) V6.1 information center

<http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wesb.61x.root.doc/info/welcome.html>

This chapter includes the following topics:

- ▶ Administrative tools
- ▶ Managing stand-alone server environments
- ▶ Managing distributed environments
- ▶ Managing the Web server plug-in configuration
- ▶ Viewing the JVM logs

## 5.1 Administrative tools

WebSphere Process Server provides a variety of administrative tools for configuring and managing your runtime environment. The combination of administrative tools that you employ ultimately depends on the size and complexity of your runtime environment.

The administrative tools that we discuss in this chapter are:

- ▶ Administrative console  
Use the administrative console to perform the deployment and system administration tasks through a Web interface.
- ▶ WebSphere scripting client (**wsadmin**)  
The **wsadmin** tool is intended for production environments and unattended operations.
- ▶ Command line tools  
Using the command line tools, you can start and stop application servers, check server status, add or remove nodes, and complete similar tasks.
- ▶ IBM Web Administration for i5/OS

The use of commands, scripting, and the administrative console is common across the operating system platforms. In addition, there might be other methods that are specific to an operating system for starting and stopping processes.

For example, on a Windows system, you can register a WebSphere process as a Windows service when you create the profile. You can start the process (application server, node, and deployment manager) by starting the service. You can also use the Windows Start menu to start or stop processes. For example, you can start and stop the deployment manager using the Start menu, for

example **Start** → **Programs** → **IBM WebSphere** → **Process Server 6.1** → **Profiles** → *profile\_name* → **Start the deployment manager**.

On z/OS systems, processes can be started or stopped by managing the JCL procedure for the process.

On i5/OS, you can use the IBM Web Administration for i5/OS to manage processes.

### 5.1.1 Using commands

Command line tools enable you to perform management tasks, such as starting, stopping, and checking the status of processes and nodes. These tools work only on local servers and nodes. They cannot operate on a remote server or node. To administer a remote server, you need to use the administrative console or a **wsadmin** script that connects to the deployment manager for the cell in which the target server or node is configured.

Command line tools are located in the *WPS\_install\_root/bin* directory. In addition, there is a proxy version of each script that resides in each profile's bin directory. All command line tools function relative to a particular profile. If you run a command from the *WPS\_install\_root/bin* directory, the command runs within the default profile. You can specify a different profile using the `-profileName` parameter. If you invoke a script from the bin directory of the profile, you do not need to specify the `-profileName` parameter.

#### Installation and profile directory locations

We use variables in this book to denote the location of a directory, file, or command. The meaning of these variables differs based on whether you are installing the product on a clean server or on a server with an existing installation of WebSphere Application Server, WebSphere Application Server Network Deployment, WebSphere Process Server, or WebSphere ESB. They can also differ depending on whether you are performing the installation as a root (Administrator on a Windows system) or non-root user.

For information about default directory locations for WebSphere Process Server, including those used when the Network Deployment installation exists before the WebSphere Process Server installation, see the following resources:

- ▶ Default installation directories for the product, profiles, and tools  
[http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/doc/rins\\_installation\\_paths.html](http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/doc/rins_installation_paths.html)
- ▶ WebSphere Process Server file system directories (z/OS)  
[http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.z.610.doc/doc/cins\\_zos\\_filesys.html](http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.z.610.doc/doc/cins_zos_filesys.html)

For information about WebSphere ESB installations, see the following Information Center articles:

- ▶ Default installation directories for the product, profiles, and tools  
[http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wesb610.doc/doc/rins\\_installation\\_paths.html](http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wesb610.doc/doc/rins_installation_paths.html)
- ▶ WebSphere ESB file system directories (z/OS)  
[http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wesb610.zseries.doc/doc/cins\\_zos\\_filesys.html](http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wesb610.zseries.doc/doc/cins_zos_filesys.html)

In this book, we use the following variables to denote the installation and profile directory locations.

- ▶ *WPS\_install\_root*: Indicates the location of the WebSphere Process Server installation
- ▶ *profile\_root*: Indicates the location of a WebSphere Process Server profile

For simplicity, we assume that the installation is a WebSphere Process Server installation, that WebSphere Process Server and WebSphere Application Server are installed at the same time, and that a root user ID was used to perform the installation. Using these assumptions, the default *WPS\_install\_root* directories are as follows:

- ▶ AIX®: /usr/IBM/WebSphere/ProcServer
- ▶ HP-UX and Solaris: /opt/IBM/WebSphere/ProcServer
- ▶ Linux: /opt/ibm/WebSphere/ProcServer
- ▶ Windows: C:\Program Files\IBM\WebSphere\ProcServer
- ▶ i5/OS: /QIBM/ProdData/WebSphere/ProcServer

► **z/OS:**

The installation file systems for WebSphere Process Server and the underlying WebSphere Application Server are stored in two different locations.

The default location of the WebSphere Process Server installation file system is `/usr/lpp/zWPS/V6R1`.

The default location of the WebSphere Application Server installation file system is `/usr/lpp/zWebSphere/V6R1`.

The default *profile\_root* directories are as follows:

- AIX: `/usr/IBM/WebSphere/ProcServer/profiles/profile_name`
- HP-UX and Solaris: `/opt/IBM/WebSphere/ProcServer/profiles/profile_name`
- Linux: `/opt/ibm/WebSphere/ProcServer/profiles/profile_name`
- Windows: `C:\Program Files\IBM\WebSphere\ProcServer\profiles\profile_name`
- i5/OS: `/QIBM/UserData/WebSphere/ProcServer/profiles/profile_name`
- z/OS: `/WebSphere/V6R1/profile_name/config/bin`

## **Using the QShell environment to enter commands in i5/OS**

Qshell is a UNIX-like environment that is available on i5/OS. There are several methods to run QShell commands to ensure that you use (from the correct directory) the correct version:

- Invoke the fully qualified path name of the script, where *script\_name* is the name of the script and *parameters* represents the parameters that are passed to the script as follows:

```
/QIBM/ProdData/WebSphere/ProcServer/bin/script_name parameters
```

- Invoke the script from the CL command line or from an i5/OS CL program, where *script\_name* is the name of the script, and *parameters* represents the parameters that are passed to the script. To use this method, run the STRQSH command, and specify the fully qualified path name of the script as follows:

```
STRQSH CMD(' /QIBM/ProdData/WebSphere/ProcServer/bin/script_name
parameters')
```

- Use the **cd** command to change to the bin directory:

```
cd /QIBM/ProdData/WebSphere/ProcServer/bin
```

Then run the following script:

```
script_name parameters
```

In the script:

- *script\_name* is the name of the script.
- *parameters* represents the parameters that are passed to the script.

To enter commands using the QShell environment, we preferred to issue the commands from the *profile\_root/bin* location to avoid using the *-profileName* parameter. The process used to enter commands is:

1. Start a 5250 emulator session and sign on.
2. Start a QShell session with the STRQSH command.
3. Change to the profile home location as follows:

```
cd profile_root/bin
```
4. Run the script.

### 5.1.2 Using the administrative console

The WebSphere administrative console is the graphical, Web-based tool that you use to configure and manage an entire WebSphere cell. It supports the full range of product administrative activities, such as creating and managing resources and applications, viewing logs and traces, and so on.

The administrative console is a WebSphere application and is hosted on a WebSphere process. Before using the process, the process hosting the console application must be active as follows:

- In a single application server installation, the console is hosted on the application server, so you must start the server in order to reach the console.
- If you are working with a deployment manager and its managed nodes, the console is hosted on the deployment manager. You must start it to use the console.

In a WebSphere Process Server installation, the administrative console contains additional functionality that allows you to manage WebSphere Process Server and WebSphere ESB specific tasks.

Figure 5-1 shows the WebSphere administrative console.

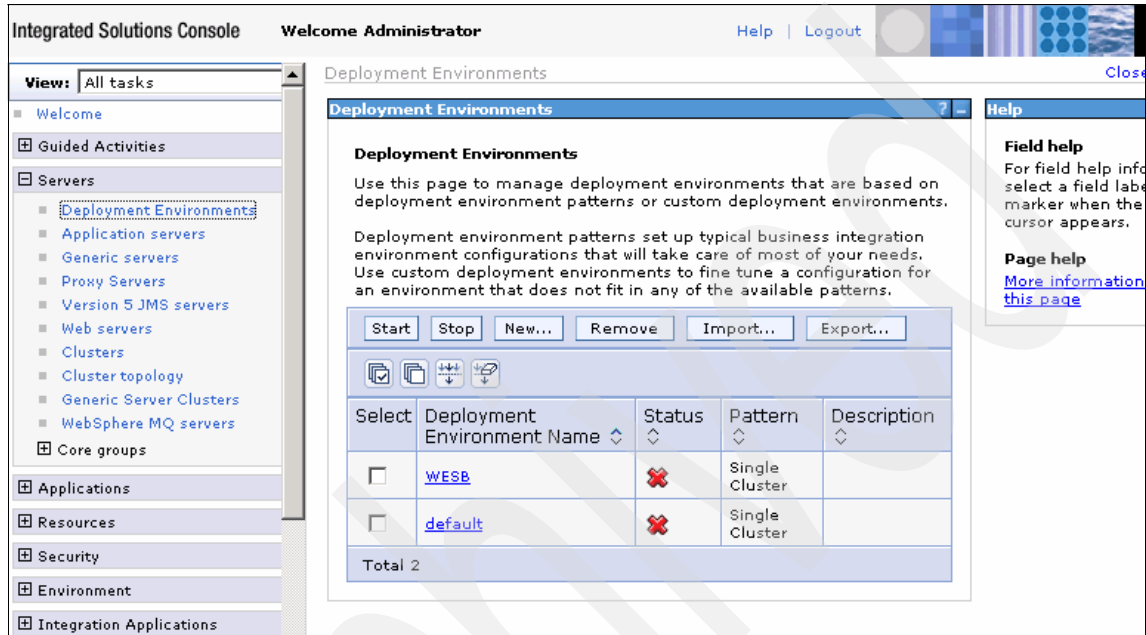


Figure 5-1 The administrative console graphical interface

## Non-secure administration access

If administrative security is not enabled, the console is accessed through a Web browser using the following URL:

`http://hostname:port/ibm/console`

For example, `http://localhost:9060/ibm/console`.

You can gain access to the console without entering a user name. If you do enter a name, it is not validated and is used exclusively for logging purposes and to enable the system to recover the session if it is lost while performing administrative tasks.

## Secure administration access

If administrative security is enabled, the console is accessed through a Web browser through the following URL:

```
https://hostname:port/ibm/console
```

**Note:** The URL uses *https://* versus *http://*.

You must enter an authorized user ID and password to log in. The actions that you can perform within the console are determined by your role assignment.

## Finding the administrative console port

The administrative console port that is used to access the console varies depending on the ports that you specified when you created the application server profile.

You can find the port numbers for an application server or deployment manager in *profile\_root/logs/AboutThisProfile.txt*. If the server is in the WebSphere Integrated Test environment, you can find this in *WID\_ROOT\pf\profile\_name\logs\AboutThisProfile.txt*.

The file contains useful information similar to that shown in Figure 5-2.

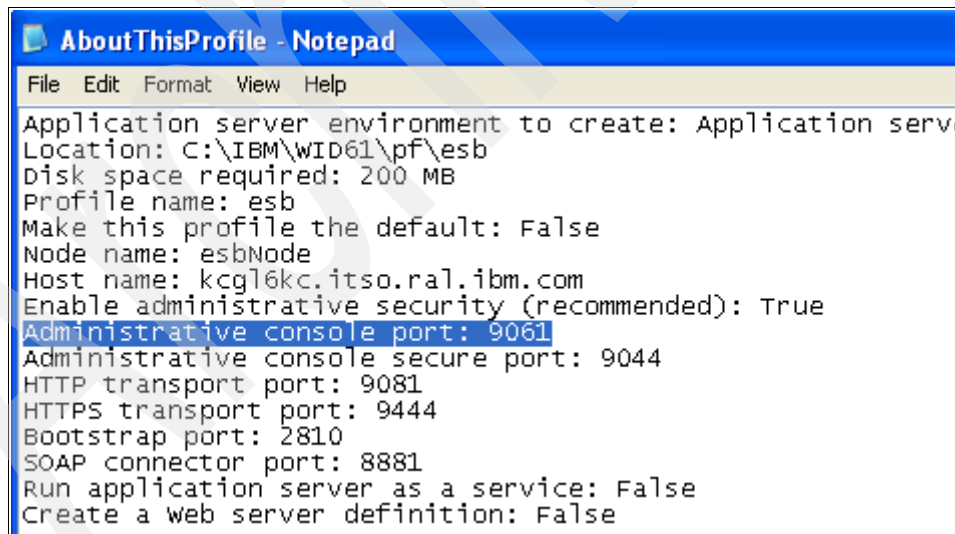


Figure 5-2 Find the port for the administrative console

## User IDs

The user ID that is specified during login is used to track configuration changes that are made by the user. This tracking allows you to recover from unsaved session changes made under the same user ID, for example when a session times out or the user closes the Web browser without saving. The user ID for login depends on whether WebSphere administrative security is enabled as follows:

- ▶ WebSphere administrative security is not enabled.

You can enter any user ID, valid or not, to log in to the administrative console. The user ID is used to track changes to the configuration but is not authenticated. You can also simply leave the User ID field blank and click **Log In**.

**Note:** Logging in without an ID is not a good idea if you have multiple administrators.

- ▶ WebSphere administrative security is enabled.

You must enter a valid user ID and password that is assigned an administrative security role.

### 5.1.3 Using IBM Web Administration for i5/OS

The IBM Web Administration for i5/OS interface combines forms, tools, and wizards to create a simplified environment to set up and manage application servers WebSphere Application Server that are currently configured on your i5/OS. You can also view server properties forms, the administrative console, and problem determination logs through this interface.

**Note:** Be aware that there are some limitations to managing your application server environment using IBM Web Administration for i5/OS that include managing profiles with administrative security enabled, managing deployment managers, or managing federated nodes.

For detailed information about IBM Web Administration for i5/OS, visit the i5/OS V5R4 Information Center:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r4/index.jsp?topic=/rzaie/rzaieconadmin.htm>

You can start IBM Web Administration for i5/OS using one of the following two methods:

- ▶ Use the i5/OS command line, and enter the following command:  
STRTCPSVR SERVER(\*HTTP) HTTPSVR(\*ADMIN)
- ▶ Use iSeries® Navigator as follows:
  - a. Start iSeries Navigator.
  - b. In the iSeries Navigator left panel, expand **My Connections** → **<your i5 host>** → **Network** → **Servers**, and click **TCP/IP**.
  - c. From the list of servers that displays in the right panel, right-click **HTTP Administration**, and select **Start**, as shown in Figure 5-3.

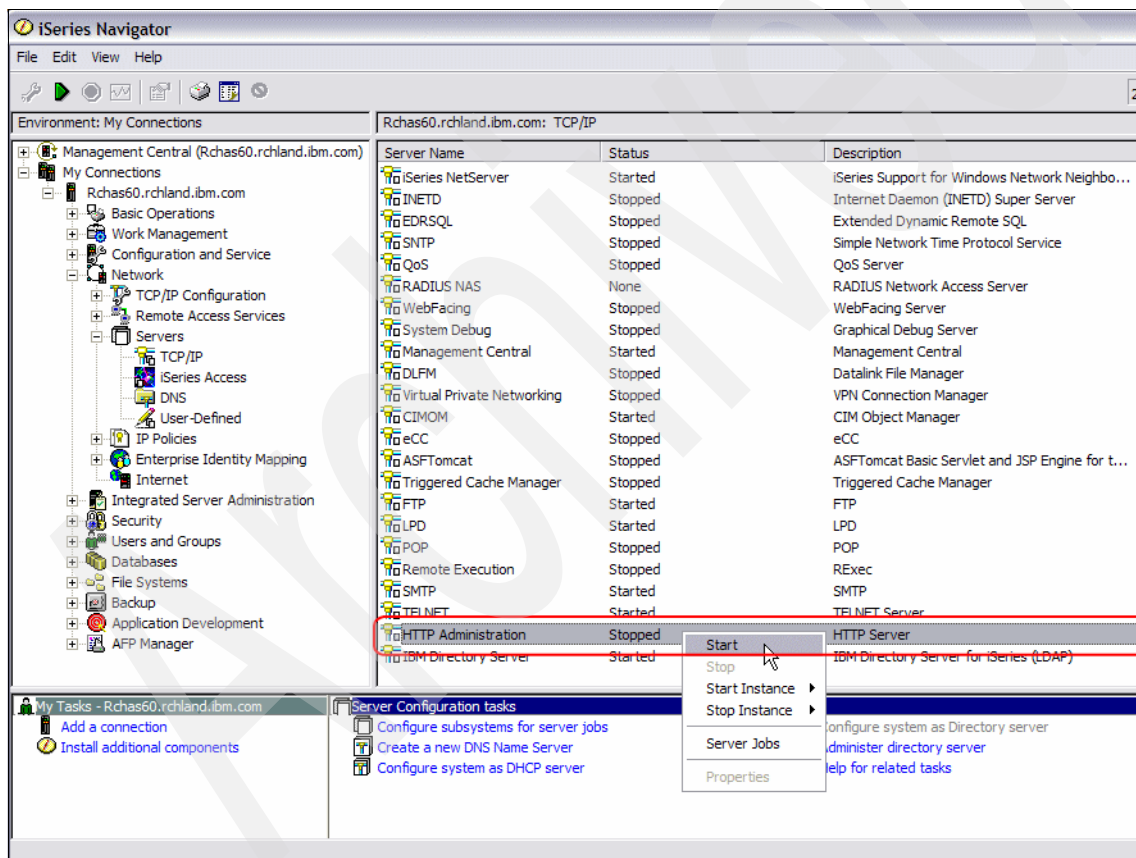
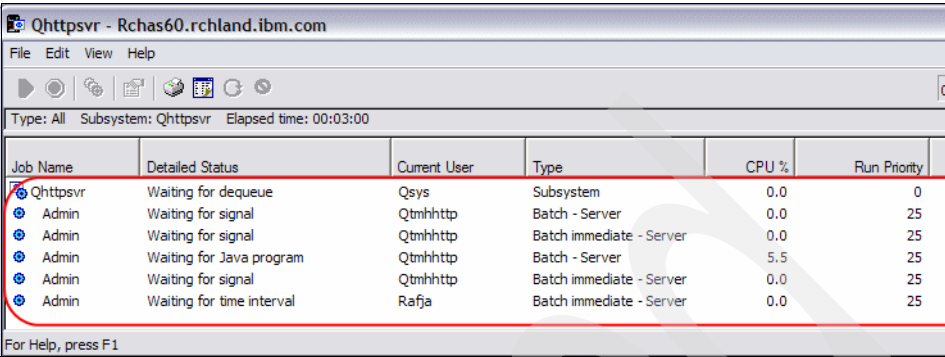


Figure 5-3 iSeries Navigator: HTTP Administration TCP/IP job

After HTTP Administration starts, the QHTTPSVR subsystem is active. Figure 5-4 shows the jobs that are running in this subsystem.



| Job Name | Detailed Status           | Current User | Type                     | CPU % | Run Priority |
|----------|---------------------------|--------------|--------------------------|-------|--------------|
| Qhttpsvr | Waiting for dequeue       | Qsys         | Subsystem                | 0.0   | 0            |
| Admin    | Waiting for signal        | Qtmhhttp     | Batch - Server           | 0.0   | 25           |
| Admin    | Waiting for signal        | Qtmhhttp     | Batch immediate - Server | 0.0   | 25           |
| Admin    | Waiting for Java program  | Qtmhhttp     | Batch - Server           | 5.5   | 25           |
| Admin    | Waiting for signal        | Qtmhhttp     | Batch immediate - Server | 0.0   | 25           |
| Admin    | Waiting for time interval | Rafja        | Batch immediate - Server | 0.0   | 25           |

Figure 5-4 QHTTPSVR jobs

The next step is to open the IBM Web Administration for i5/OS GUI:

1. Navigate to the following Web address:

`http://hostname:2001/`

In the Web address, *hostname* is the host name or IP address of the i5/OS server where the HTTP server was started. Port 2001 is the default port for accessing IBM Web Administration for i5/OS. To view the port that is assigned to this interface on your system, use the WRKSRVTBLE command and find the as-admin-http entry in the table.

2. When the login window opens, enter a valid i5/OS user profile name and password, and click **OK**.

**Important:** To be authorized to log in to IBM Web Administration for i5/OS application, the i5/OS user profile must have \*IOSYSCFG special authority.

3. After a successful login, the i5/OS Tasks page displays, as shown in Figure 5-5. To work with HTTP Server (powered by Apache) and the IBM WebSphere Application Server profiles, click **IBM Web Administration for i5/OS**.



Figure 5-5 i5/OS Tasks page

**Important:** We recommend that you install the latest HTTP Server for i5/OS PTFs on your system, prior to using IBM Web Administration for i5/OS. For more information, see:

<http://www-03.ibm.com/servers/eserver/iseriessoftware/http/services/service.html#PTF>

The IBM Web Administration for i5/OS start page displays, as shown in Figure 5-6.

**Note:** After you use IBM Web Administration for i5/OS for the first time, the state is retained so that you are not necessarily returned to the same window when you next use IBM Web Administration for i5/OS.

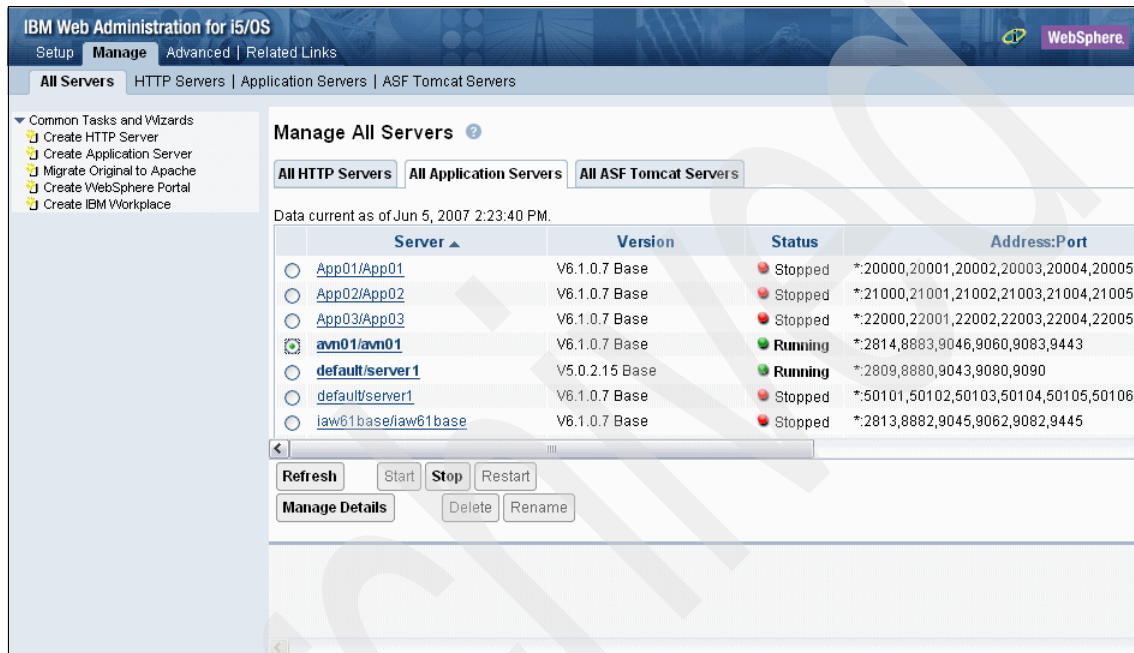


Figure 5-6 IBM Web Administration for i5/OS start page

## 5.1.4 WebSphere scripting client (wsadmin)

The WebSphere administrative scripting program (**wsadmin**) is a powerful, text console-based command interpreter environment that enables you to run scripted administrative operations. The **wsadmin** program supports two scripting languages, JACL (deprecated) and Jython.

You can run the **wsadmin** tool in interactive and unattended mode. Use the **wsadmin** tool to perform the same tasks that you can perform using the administrative console.

Command assistance is available in the administrative console to map your administrative activities to **wsadmin** scripting commands written in Jython. You can view these commands from the console, and if you want, you can log the

command assistance data to a file. You can also allow command assistance to emit Java Management Extensions (JMX) notifications to the WebSphere Application Server Toolkit. The WebSphere Application Server Toolkit has Jython development tools that help you develop and test Jython scripts.

## 5.2 Managing stand-alone server environments

In a stand-alone server environment, you have one profile that describes the run time for one application server. The administrative console runs as an application on that server.

A typical start-up sequence is as follows:

1. Start the application server using the **startServer** command to start the administrative console also.
2. Log on to the administrative console.

### 5.2.1 Managing application servers

By default, the name of a stand-alone application server is *server1*. You must start the server before you can access the administrative console.

#### Starting a server

You use the **startServer** command to start an application server as follows:

- On a Windows platform, use the following commands:

```
cd profile_root
startServer server1 -user username - password password
```

For example:

```
cd WebSphere\ProcServer\profiles\SAServer\bin
startServer server1 -user username - password password
```

If you elected to register the server as a Windows service, you can start the server by starting the service.

- On a Unix platform, use the following commands:

```
cd profile_root
startServer.sh server1 -user username - password password
```

For example, on an AIX system:

```
cd /usr/IBM/WebSphere/ProcServer/profiles/SAServer/bin
startServer server1 -user username - password password
```

- ▶ On an i5/OS system, use the following commands:

```
STRQSH
cd profile_root/bin
startServer server1 -user username - password password
```

For example:

```
STRQSH
cd /QIBM/UserData/WebSphere/ProcServer/profiles/SAServer/bin
startServer server1 -user username - password password
```

You can also manage servers using the IBM Web Administration for i5/OS tool.

- ▶ On a z/OS system:

You can manage application servers on an i5/OS system from a UNIX System Services environment as follows:

Select *uss* (to switch to the UNIX System Services environment)

```
cd profile_root/bin
startServer.sh server1 -user username - password password
```

For example:

```
cd /wascsconfig/cscell/csnodea/AppServer/profiles/default/bin
startServer.sh CSSR01A -user username - password password
```

The home directory path for the profile is specified in the zPMT configuration (see Figure 4-10 on page 109 and Figure 4-20 on page 124).

You can also start a server started by starting the JCL procedure as follows:

```
S procname, JOBNAME=server_shortname,ENV=
cell_shortname.node_shortname.server_shortname
```

## Stopping a server

You can use the **stopServer** command to stop an application server on distributed and i5/OS platforms:

```
stopServer(.sh) server1 -user username - password password
```

## Listing the server status

To list the status of the server:

```
serverStatus(.sh) server1 -user username - password password
```

Alternatively, you can use:

```
serverStatus -all -user username - password password
```

On Windows platforms, you can use the Start menu. You also have the option when you create the profile to register the server as a Windows service. If you choose this option, you can start the server by starting the service or by having the service start automatically.

## 5.3 Managing distributed environments

In a distributed environment, you have a deployment manager that acts as the central administration point for the cell. The administrative console runs as an application on the deployment manager. Application servers reside on nodes that are managed by the deployment manager.

A typical start-up sequence is as follows:

1. Start the deployment manager using the **startManager** command to start the administrative console also.
2. Start the nodes on each remote system using the **startNode** command.
3. Log on to the administrative console.
4. Start the application servers, clusters, and deployment environments using the administrative console.

### 5.3.1 Managing the deployment manager

You can start or stop the deployment manager with commands on all operating systems. As with a stand-alone server, you have other options for starting the deployment manager based on the operating system.

#### Starting the deployment manager with startManager

Using the **startManager** command from the *profile\_root/bin* directory is the most common method to start the deployment manager, as shown in Example 5-1.

*Example 5-1 The startManager command*

---

```
C:\WebSphere\ProcServer\profiles\Dmgr01\bin>startManager -user userID
-password password
```

```
ADMU0116I: Tool information is being logged in file
```

```
C:\WebSphere\ProcServer\profiles\Dmgr01\logs\dmgr\startServer.log
ADMU7701I: Because dmgr is registered to run as a Windows Service, the
request to start this server will be completed by starting the
associated Windows Service.
```

```
ADMU0116I: Tool information is being logged in file
```

```
C:\WebSphere\ProcServer\profiles\Dmgr01\logs\dmgr\startServer.log
ADMU0128I: Starting tool with the Dmgr01 profile
ADMU3100I: Reading configuration for server: dmgr
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server dmgr open for e-business; process id is 3528
```

---

## Stopping the deployment manager

You can stop the deployment manager with the **stopManager** command, as shown in Example 5-2.

### *Example 5-2 The stopManager command*

---

```
C:\WebSphere\ProcServer\profiles\Dmgr01\bin>stopManager
ADMU0116I: Tool information is being logged in file

C:\WebSphere\ProcServer\profiles\Dmgr01\logs\dmgr\stopServer.log
ADMU7702I: Because dmgr is registered to run as a Windows Service, the
request to stop this server will be completed by stopping the
associated Windows Service.
ADMU0116I: Tool information is being logged in file

C:\WebSphere\ProcServer\profiles\Dmgr01\logs\dmgr\stopServer.log
ADMU0128I: Starting tool with the Dmgr01 profile
ADMU3100I: Reading configuration for server: dmgr
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server dmgr stop completed.
```

---

## Starting the deployment manager on z/OS (START command)

On z/OS, you can start the deployment manager using a JCL start procedure. You can find the exact command in the BBOCCINS instruction member of the JCL that was generated to create the profile.

For example:

```
START CSDCR,JOBNAME=CSDMGR,ENV=CSCELL.CSDM.CSDMGR
```

Where:

- ▶ CSDCR is the JCL start procedure.
- ▶ CSDMGR is the JOBNAME.
- ▶ ENV is the concatenation of the cell short name, node short name, and server short name.

When the deployment manager starts, the following processes also start:

- ▶ A daemon: In our example, named *CSDEMN*. There is one daemon per cell per MVS image. One of the functions of the daemon server is to provide the *location name service* for the cell. All daemons in the cell are fully aware of all the objects in the cell and use the same port values.
- ▶ A controller region: In our example, named *CSDMGR*. The controller region serves many functions, including acting as the endpoint for communications.
- ▶ A servant region: In our example, named *CSDMGRS*. The servant region contains the JVM where the applications are run.
- ▶ If you are using messaging, you also see a control region adjunct server start.

### Stopping the deployment manager on z/OS (STOP command)

To stop the deployment manager with a STOP command, use the following format:

```
STOP dmgr_JOBNAME
```

For example:

```
STOP CSDMGR
```

Stopping the daemon server stops all servers for that cell, and all the servers on that daemon instance's MVS image are stopped in an order fashion. For example:

```
STOP CSDEMN
```

## 5.3.2 Managing nodes

For the deployment manager to manage nodes, you must start the node agent on each node.

### Starting a node agent

When a node agent is stopped, the deployment manager has no way to communicate with it. Therefore, the node agent must be started with the **startNode** command run from on the profile node system, as shown in Example 5-3.

*Example 5-3 The startNode command*

---

```
C:\WebSphere\ProcServer\profiles\pxpv812Node01\bin>startNode -user
userID -password password
ADMU0116I: Tool information is being logged in file
```

```
C:\WebSphere\ProcServer\profiles\pxpv812Node01\logs\nodeagent\startServer.log
ADMU0128I: Starting tool with the pxpv812Node01 profile
ADMU3100I: Reading configuration for server: nodeagent
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server nodeagent open for e-business; process id is 6128
```

---

## Starting a node on z/OS using the START command

To start a node agent on z/OS using the START command, use the following format:

```
START nodeagent_procname,JOBNAME=server_shortname,
ENV=cell_shortname.node_shortname.server_shortname
```

For example:

```
S CSACRA,JOBNAME=CSAGNTA,ENV=CSCELL.CSNODEA.CSAGNTA
```

## Stopping a node agent

How you stop the node agent and leave the servers running depends on your preferred method.

From the administrative console:

1. From the administrative console, select **System Administration** → **Node Agents**.
2. Select the node agent for the server, and click **Stop**.

From a command prompt, use the **stopNode** command, as shown in Example 5-4.

*Example 5-4 The stopNode command*

---

```
C:\WebSphere\ProcServer\profiles\pxpv812Node01\bin>stopNode -user
userID -password password
ADMU0116I: Tool information is being logged in file

C:\WebSphere\ProcServer\profiles\pxpv812Node01\logs\nodeagent\stopServer.log
ADMU0128I: Starting tool with the pxpv812Node01 profile
ADMU3100I: Reading configuration for server: nodeagent
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server nodeagent stop completed.
```

---

**Note:** When you stop the node agent, the deployment manager has no way to communicate with the servers on that node. The servers might be up and running, but the administrative console is not able to determine their status.

### Stopping a node on z/OS using the STOP command

To stop a node agent on z/OS, you can use the following command:

```
STOP nodeagent_JOBNAME
```

For example:

```
STOP CSAGNTA
```

### Stopping a node (the node agent and servers)

You can use the administrative console to stop a node and its servers with one action:

1. From the administrative console, select **System Administration** → **Nodes**.
2. Select the node, and click **Stop**.

### Restarting a node agent

You can restart a running node agent from the administrative console by doing the following from the administrative console:

1. Select **System Administration** → **Node Agents**.
2. Select the node agent for the server, and click **Restart**.

## 5.3.3 Managing servers and clusters from the administrative console

Although you can manage application servers using commands, the simplest method is to use the administrative console, as shown in Figure 5-7 on page 183.

To manage an application server using the administrative console, navigate to **Servers** → **Application Servers**. Select the servers that you want to manage, then click the appropriate button (**Start**, **Stop**, and so forth)

| <div> <div>NewDeleteTemplates...StartStopImmediateStopTerminate</div> <div> </div> </div> |                                                |               |                                                    |                |        |
|-------------------------------------------------------------------------------------------|------------------------------------------------|---------------|----------------------------------------------------|----------------|--------|
| Select                                                                                    | Name                                           | Node          | Version                                            | Cluster Name   | Status |
| <input type="checkbox"/>                                                                  | <a href="#">WESB.AppTarget.xpxv812Node01.0</a> | xpxv812Node01 | Business Process Choreographer 6.1.0.0 ND 6.1.0.13 | WESB.AppTarget |        |
| <input type="checkbox"/>                                                                  | <a href="#">WESB.AppTarget.xpxv812Node02.0</a> | xpxv812Node02 | Business Process Choreographer 6.1.0.0 ND 6.1.0.13 | WESB.AppTarget |        |

Figure 5-7 Managing application servers

You can also manage all the servers in a cluster simultaneously, as shown in Figure 5-8. Select **Servers** → **Clusters**.

| <div> <div>NewDeleteStartStopRipplestartImmediateStop</div> <div> </div> </div> |                                   |        |
|---------------------------------------------------------------------------------|-----------------------------------|--------|
| Select                                                                          | Name                              | Status |
| <input type="checkbox"/>                                                        | <a href="#">WESB.AppTarget</a>    |        |
| <input type="checkbox"/>                                                        | <a href="#">default.AppTarget</a> |        |
| Total 2                                                                         |                                   |        |

Figure 5-8 Managing clusters

Finally, you can manage all the clusters in a deployment environment simultaneously, as shown in Figure 5-9. Select **Servers** → **Deployment Environments**.

| <div> <div>Start</div> <div>Stop</div> <div>New...</div> <div>Remove</div> <div>Import...</div> <div>Export...</div> </div> |                             |        |                |             |
|-----------------------------------------------------------------------------------------------------------------------------|-----------------------------|--------|----------------|-------------|
| <div> <div>✓</div> <div>📄</div> <div>↕</div> <div>🔄</div> </div>                                                            |                             |        |                |             |
| Select                                                                                                                      | Deployment Environment Name | Status | Pattern        | Description |
| <input type="checkbox"/>                                                                                                    | <a href="#">WESB</a>        | ➡      | Single Cluster |             |
| <input type="checkbox"/>                                                                                                    | <a href="#">default</a>     | ➡      | Single Cluster |             |
| Total 2                                                                                                                     |                             |        |                |             |

Figure 5-9 Managing deployment environments

## 5.4 Managing the Web server plug-in configuration

You need to regenerate and propagate the plug-in configuration file to the Web servers when there are changes to the WebSphere configuration that affect how requests are routed from the Web server to the application server. These changes include:

- ▶ Installing an application
- ▶ Creating or changing a virtual host
- ▶ Creating a new server
- ▶ Modifying HTTP transport settings
- ▶ Creating or altering a cluster

You can regenerate the plug-in file manually using the administrative tools. You can also set up the plug-in properties of the Web server to enable automatic generation of the file whenever a relevant configuration change is made.

To regenerate the plug-in configuration manually, you can either use the administrative console, or you can issue the **GetPluginCfg** command.

## 5.4.1 Generating the plug-in with administrative console

To generate or regenerate the plug-in configuration file:

1. Select **Servers** → **Web servers**.
2. Select your Web server, and click **Generate Plug-in**.
3. Verify that the generation was successful by looking at the messages. A success message is accompanied with the location of the generated plug-in configuration file, such as:

```
profile_root/config/cells/<cell_name>/nodes/<web_server_node>/servers/<web_server>/plugin-cfg.xml
```

To use the new plugin-cfg.xml, file you must propagate it to the Web server system.

## 5.4.2 Enabling automated plug-in regeneration

The Web server plug-in configuration service by default regenerates the plugin-cfg.xml file automatically. You can view or change the configuration settings for the Web server plug-in configuration service.

To view or change the plug-in generation property:

1. Select **Servers** → **Web servers**.
2. Select your Web server, and select **Plug-in properties** in the Additional Properties section.
3. View or change the **Automatically generate the plug-in configuration file** option.

When selected, the Web server plug-in configuration service generates the plug-in configuration file automatically whenever the Web server environment changes. For example, the plug-in configuration file is regenerated whenever one of the following activities occurs:

- A new application is deployed on an associated application server.
- The Web server definition is saved.
- An application is removed from an associated application server.
- A new virtual host is defined.

Whenever a virtual host definition is updated, the plug-in configuration file is automatically regenerated for all of the Web servers.

### 5.4.3 Propagating the plug-in configuration file

After you regenerate a plug-in configuration file, you need to propagate it to the Web server.

The configuration service can propagate the plugin-cfg.xml file to a Web server automatically if the Web server is configured on a managed node and to an IBM HTTP Server if it is configured on an unmanaged node. For other scenarios, you must manually copy the file to the Web server machines.

You can propagate the file manually by copying it from the application server system to the Web server, or you can do propagate the file from the administrative console.

#### From a command window

To copy the file from one system to another:

1. Copy the file:

```
profile_root/config/cells/<cell_name>/nodes/<web_server_node>/servers/<web_server>/plugin-cfg.xml
```

2. Place the copy of the file in the following directory on the remote Web server machine:

```
<plugins_home>/config/<web_server>
```

#### From the administrative console

To propagate the plug-in configuration manually from the administrative console:

1. Select **Servers** → **Web servers**.
2. Select your Web server, and click **Propagate plug-in**.
3. Verify that the propagation was successful by looking at the messages.

If you are in doubt, check whether the plug-in configuration file was propagated to the Web server plug-in location by viewing it.

### 5.4.4 Activating the new plug-in configuration

The Web server binary plug-in module checks for a new configuration file every 60 seconds. You can wait for the plug-in to find the changes, or you can restart the Web server to invoke the changes immediately.

**Tip:** If you encounter problems restarting your Web server, check the `http_plugin.log` file in `<plug-ins_home>/config/<web_server>` for information about what portion of the `plugin-cfg.xml` file contains an error. The log file states the line number on which the error occurred, along with other details that might help you diagnose why the Web server did not start.

### 5.4.5 Enabling automated plug-in propagation

The Web server plug-in configuration service by default propagates the `plugin-cfg.xml` file automatically. To view or change the plug-in propagation property.

1. Select **Servers** → **Web servers**.
2. Select your Web server, and select **Plug-in properties** in the Additional Properties sub section.
3. View or change the **Automatically propagate plug-in configuration file** option.

## 5.5 Viewing the JVM logs

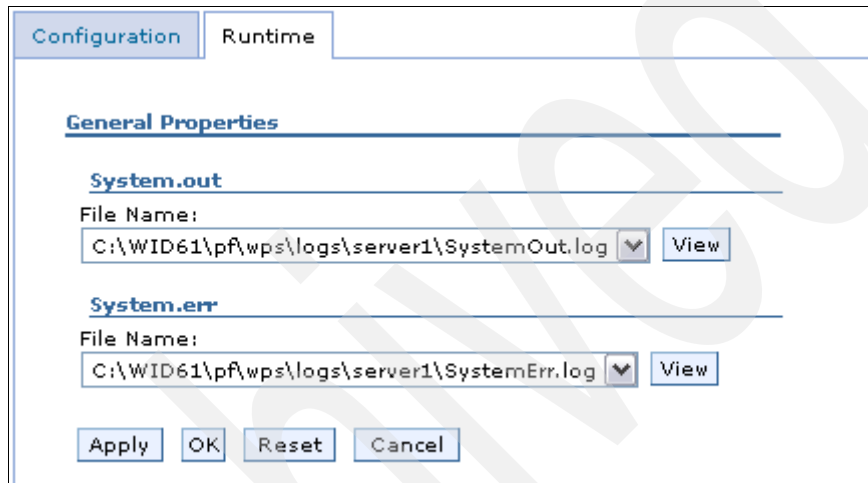
SystemOut and SystemErr logs are created for every process (application server, node agent, and deployment manager). These logs are known as *JVM logs*. WebSphere Application Server writes to these logs. Applications can also write to them by using the `print()`, `println()`, and `printStackTrace()` methods.

You can find the logs in the following directory:

`profile_root/logs/process_name`

You can also view and configure these logs through the administrative console. To find the JVM logs, select **Troubleshooting** → **Logs and Trace**. Select the process whose logs you want to configure, and then click **JVM Logs**. This view has two tabs:

- ▶ The Configuration tab allows you to change the location of the logs, the format, file size, and so forth.
- ▶ The Runtime tab allows you to view the logs. Click **View** (to the right of the log) as shown in Figure 5-10.



The screenshot shows the 'Configuration' tab of the JVM Logs configuration page. It features a 'General Properties' section with two log entries: 'System.out' and 'System.err'. Each entry has a 'File Name' field with a dropdown menu and a 'View' button. The 'File Name' for 'System.out' is 'C:\WID61\pf\wps\logs\server1\SystemOut.log' and for 'System.err' is 'C:\WID61\pf\wps\logs\server1\SystemErr.log'. At the bottom are buttons for 'Apply', 'OK', 'Reset', and 'Cancel'.

Figure 5-10 Viewing the JVM logs in the administrative console.

When applications are running in a cluster, you might not know which server in the cluster is actually executing the application. In this case, you need to browse the log for each server in the cluster until you find the server for which you are looking.

For more information about WebSphere logs and tracing, see *WebSphere Application Server V6: Diagnostic Data*, REDP-4085.

# Managing applications

This chapter provides an overview of the basic tasks that are associated with preparing WebSphere Process Server V6.1 and WebSphere Enterprise Service Bus (WebSphere ESB) V6.1 production environments for the execution of modules and mediation modules. In this chapter, we chose a single cluster topology as the example topology.

This chapter includes the following topics:

- ▶ Scenario: Order Management System
- ▶ Planning for deployment
- ▶ Setting up the prerequisites for Order Management System
- ▶ Defining application resources
- ▶ Deploying applications
- ▶ Testing the application

## Order Management System scenario:

Much of the discussion in this chapter is based on the Order Management System scenario discussed in *Getting Started with IBM WebSphere Process Server and IBM WebSphere Enterprise Service Bus Part 2: Scenario*, SG24-7642.

## 6.1 Scenario: Order Management System

In *Getting Started with IBM WebSphere Process Server and IBM WebSphere Enterprise Service Bus Part 2: Scenario*, SG24-7642, we designed and implemented the Order Management System solution. This chapter uses that solution to illustrate the actions and techniques used to plan and deploy a solution to a WebSphere Process Server runtime environment.

## 6.2 Planning for deployment

In the planning phase, you need to determine:

- ▶ The various components that make up your application
- ▶ The mutual dependencies of these components
- ▶ The external system dependencies
- ▶ Requirements of any kind (operative system, hardware constraints)
- ▶ Your available or desirable hardware

The result of this exercise can help you build a deployment plan (in the form of a UML deployment diagram) that defines where your application components deploy.

Table 6-1 shows the results of this analysis for the Order Management System scenario.

Table 6-1 Scenario application deployment systems

| System name                           | System description                                                                        | Requirements and constraints                                                                                           |
|---------------------------------------|-------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| <b>Repository System</b>              | Hosts the application database.                                                           | Apache Derby installed;<br>ORDERDB database created                                                                    |
| <b>Warehouse Availability Systems</b> | Three systems that each provide warehouse availability information through a Web service. | Minimum of WebSphere ESB capability on the servers because the Web services have been implemented as mediation modules |
| <b>E-mail Server System</b>           | Provides an SMTP service for mailing.                                                     | An SMTP Server installed                                                                                               |
| <b>DB Access System</b>               | Provides access to the application repository. Hosts the DBMSServiceMediation.            | WebSphere ESB capable server;<br>Connectivity with the Repository System                                               |

| System name                          | System description                                                                                                                                       | Requirements and constraints                                                                                                                                                                            |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Warehouse Split System</b>        | Provides the warehouse order split service. Hosts the WarehouseAvailabilityMediation.                                                                    | WebSphere ESB capable server;<br>Connectivity with the Warehouse Availability Systems                                                                                                                   |
| <b>Order Management System</b>       | Hosts the core business process application, OrderManagement and the e-mail module, EmailOutbound.                                                       | WebSphere Process Server capable server;<br>Connectivity with the <ul style="list-style-type: none"> <li>▶ DB Access System</li> <li>▶ Warehouse Split System</li> <li>▶ Email Server System</li> </ul> |
| <b>Special Order Handling System</b> | Filters special order requests to be sent to the file system (hosts OrderPreProcessingMediation). Forward regular order requests to the OrderManagement. | <i>Windows</i> platform (specific to the Flat File configuration);<br>WebSphere ESB capable server;<br>Connectivity with Order Management System                                                        |

Figure 6-1 shows a Deployment Diagram with the minimum *collapsing systems* that are possible for deployment, based on Table 6-1. We look at these requirements and constraints in more detail in the next sections.

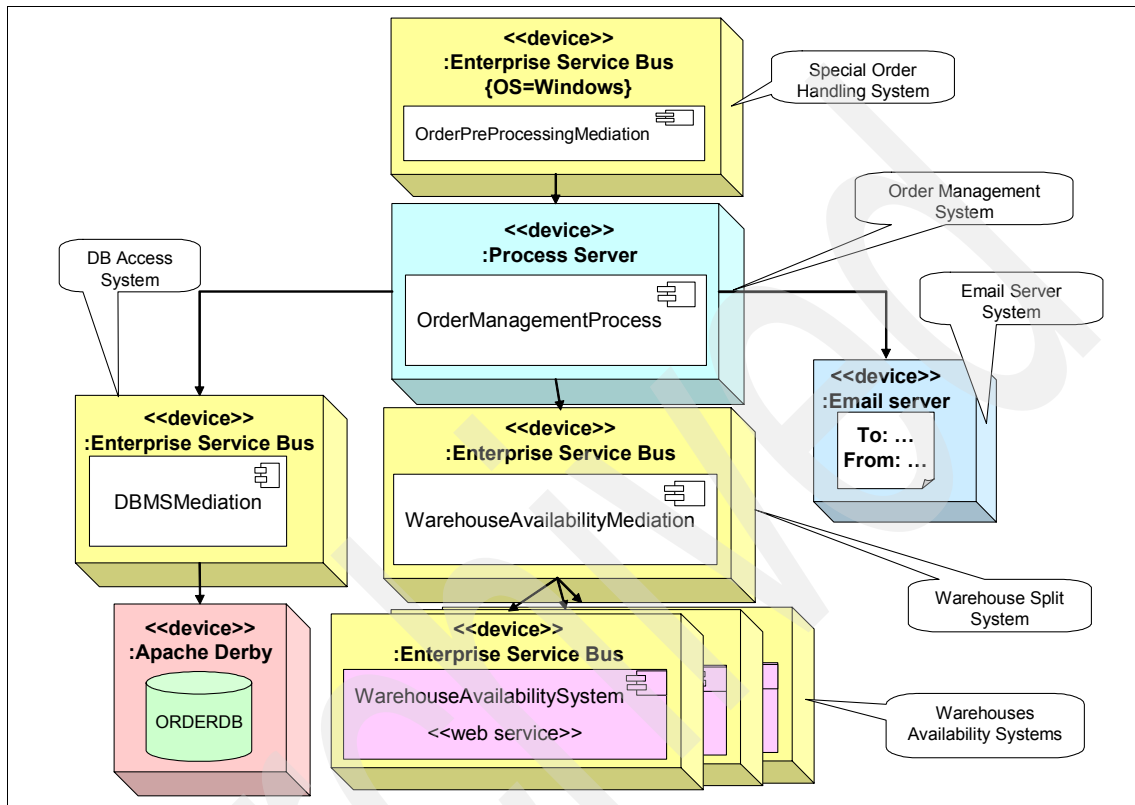


Figure 6-1 Scenario example deployment diagram, in a “completely remote” fashion

The next step is to lay out the topology that you use for the deployment of these systems. The target deployment topology that you choose is typically based on several factors such as:

- ▶ Hardware requirements for any specific system
- ▶ Hardware availability and characteristics
- ▶ Performance, scalability and failover requirements of your application.

Figure 6-2 shows the topology that we use in this chapter.

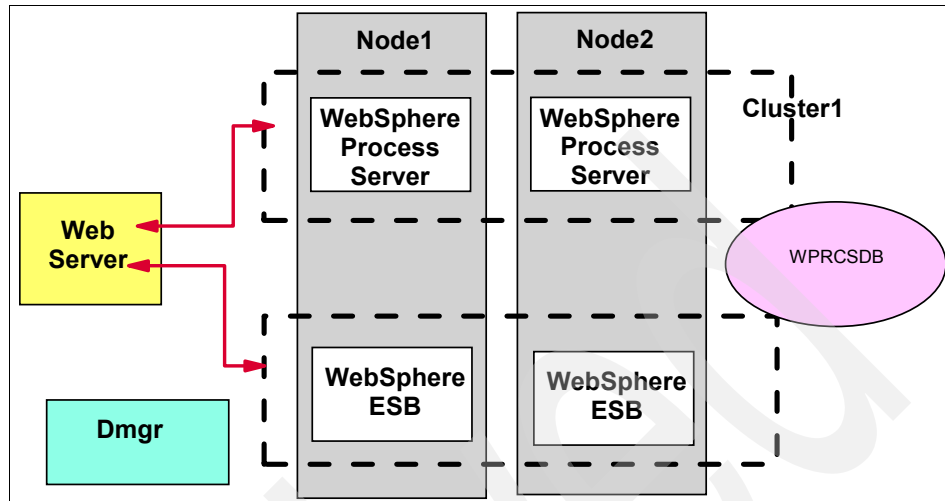


Figure 6-2 Single cluster topology pattern

The Web server handles all requests between the two clusters as well as the initial request from the client. It uses the Web server plug-in to balance requests among servers in each cluster. The process is as follows:

1. When a request for the application comes into the system from a client, the Web server plug-in is used to forward the request to one of the servers in the WebSphere ESB cluster for processing by the OrderPreProcessingMediation application.
2. When a mediation running in the WebSphere ESB cluster needs to access a WebSphere Process Server module (that is the OrderPreProcessingMediation needs to access the OrderManagement), the request is routed to one of the servers in the WebSphere Process Server cluster by the Web server plug-in.
3. When a module executing in the WebSphere Process Server cluster needs to access a mediation module (that is the OrderManagement needs to access the DBMS Mediation), the request is routed to one of the servers in the WebSphere ESB cluster by the Web server plug-in.

Figure 6-3 shows the final deployment diagram for the Order Management System using a single cluster topology. The topology is spread over two systems, *host1* and *host2*.

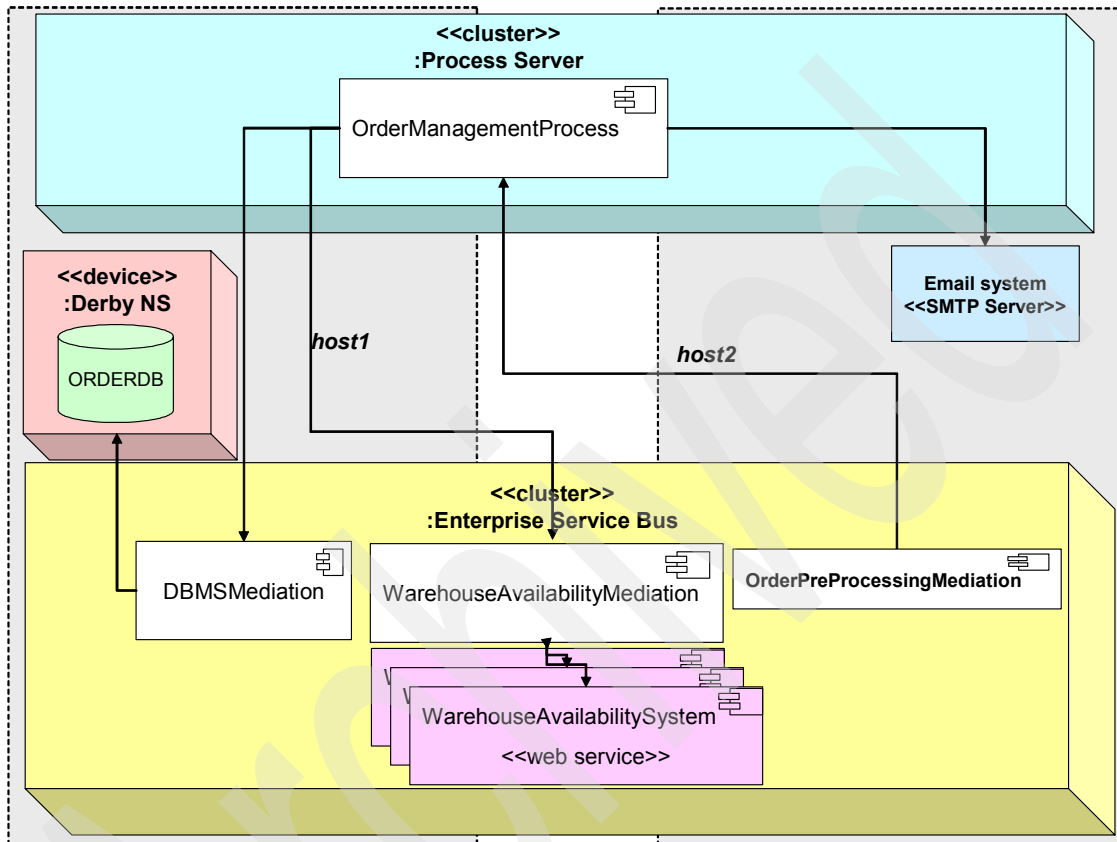


Figure 6-3 Scenario example over single cluster pattern deployment diagram

The WebSphere Process Server cluster hosts the OrderManagement, while the WebSphere ESB cluster hosts the mediation modules. Each cluster spans *host1* and *host2*.

The e-mail system is on the *host1* machine, which is accessible from the WebSphere Process Server cluster.

**Note:** If the e-mail system connectivity properties are set in the application during development, remember to update the properties using WebSphere Integration Developer to reflect the production e-mail server address and to repack the module for deployment.

You can find the properties in the Binding tab in the Properties view for the import component that calls the e-mail server.

The Repository system is on host1 and is accessed by the WebSphere ESB cluster through a data source. Information on how to configure the data source is in 6.3, “Defining application resources” on page 197.

In our example, we deploy to a cluster, thus the Derby Embedded provider used in the test environment is no longer feasible for our deployment. So, we must move the repository to a Derby Network Server. Derby Network Server is available with a WebSphere Process Server or WebSphere ESB installation. It mediates the access to the database through a server that can be accessed from multiple clients, which is what we need in a cluster environment. Not that Derby is not intended for production. In a production system, you need to plan to use the database product that provides the function and performance that you require.

## 6.2.1 Setting up the prerequisites for Order Management System

In most cases, a business integration application has some external system dependencies. General examples include Enterprise Information Systems, Web services (can be implemented using a wide variety of technologies), and technologies or applications that are accessed through adapters.

In this phase, you must ensure that these external systems are started, accessible, working, and configured as stated in the planning phase. You must also ensure that the topology that is required for hosting the modules and mediation modules is in place. For the Order Management System the outcome of this phase is that host1, host2, and the external systems are ready for use.

### WebSphere runtime topology setup

Figure 6-2 on page 193 shows the runtime topology for the Order Management System. We assume that the components in this topology are built and are active so that we can begin setting up and testing the prerequisite systems.

## External systems setup

We provided a list of the application systems and their requirements in Table 6-1 on page 190. We now go through every system in that list and show how we completed the preparation of the external systems.

### *Special Order Handling System setup*

The Special Order Handling System hosts the OrderPreProcessingMediation. This mediation receives the request for an order sent by a client. It filters the incoming order, placing special order requests on a file system, and forwarding regular order requests to OrderManagement in the Order Management System.

To prepare this system, you must create the file system that is defined by the flat file adapter implementation.

Because the decision was made to put the flat files to a local Windows file system, it is assumed The Special Order Handling System runs on a Windows platform. The Flat File Adapter is configured to put files to a local folder *C:\ITSOSpecialOrders*. Because we know the entire OrderPreProcessingMediation is deployed later over a WebSphere ESB cluster, we must ensure this single folder is accessible from every cluster member. A possible solution to such a requirement is a shared file system or a shared folder with the same name on every cluster machine. You can change this requirement by reconfiguring the Flat File adapter.

### *Repository System setup*

The Repository System must have an Apache Derby Network Server installed and running. The application repository, the ORDERDB database, must be created.

Apache Derby is a lightweight database included in every WebSphere Process Server V6.1 and WebSphere ESB V6.1 installation. We chose this database type because it is readily available to users of these products. However, Derby is not intended to be used in a production environment.

The database provider is something that you want to decide early in the process. Databases that are created on one type of provider might not be migrated easily to another. The database must be created on the selected provider system before configuring the JDBC adapter.

To change the database provider type after development is complete, follow these steps:

1. Migrate the database from one type of provider to another.
2. Change the JDBC adapter configuration to use the new provider.
3. Configure the runtime resources (JDBC provider and data source).

For the Order Management System, using Derby, the repository is prepared for the production environment on host1 as follows:

1. Create the database.

The ORDERDB database is created and populated with data in the following location:

C:\itso\sampleDB\ORDERDB

2. Start the Network Server server:

derby\bin\networkServer\startNetworkServer.bat

### ***E-mail Server System setup***

The E-mail Server System must have an SMTP server installed, configured, running and working. In our case, we installed an SMTP server on host2.

The E-mail Server location and port is configured within the E-mail Adapter configuration, which in our case, is in the OrderManagement module.

### ***Setting up the remaining systems***

The following systems host the business integration and mediation modules. These systems do not have any external requirements but must have the following network connectivity:

- ▶ The Order Management System machines must be able to connect to the e-mail system.
- ▶ The DB Access System accesses the Repository system using a data source. We describe data source setup in 6.3, “Defining application resources” on page 197 and check the connectivity to the database then.
- ▶ The Warehouse Split Systems machines must be able to connect to Warehouse Availability Systems.

## **6.3 Defining application resources**

This section discusses how to define resources that are required by your applications to the runtime environment. These resources include data sources for access to databases, a user registry and authentication aliases for security, and JMS resources for messaging.

### 6.3.1 Setting the resource scope

Resources are defined at a *scope* level in the WebSphere Process Server configuration. The scope can be:

- ▶ **Cell:** The resource is available to all servers in the cell
- ▶ **Node:** The resource is available to all servers in the node
- ▶ **Cluster:** The resource is available to all servers in the cluster
- ▶ **Server:** The resource is only available to a specific server
- ▶ **Application:** The resource is only available to an application and applies only to enhanced EAR files

When you define a resource, you select the most appropriate scope. For example, JDBC providers provide the location of the driver classes. If the location of these files vary between systems, you define a JDBC provider for each node that requires access to a database and then define a data source for the database to each JDBC provider at the node level.

Resources that are defined at more specific scopes override duplicate resources that are defined at more general scopes. For example, if a JDBC provider is defined at both the cell and node scope, a server in that node uses the JDBC provider defined at the node scope. A server that is not in that node uses the JDBC provider defined at the cell scope.

### 6.3.2 Defining security credentials to access a resource

An application that accesses a resource often has to pass credentials to the resource manager to gain access to the resource. For example, when the JDBC adapter accesses a database, it most likely needs to provide a user ID and password with the appropriate authorities to read and write from the database. The best way to pass these credentials is to define them in a J2C authentication alias and then have the resource configuration point to the appropriate alias. A J2C authentication alias simply contains a user name and password.

#### **Example: Authentication alias for ORDERDB**

For the Order Management System, you need to define an authentication alias with the user ID and password that is required to access the ORDERDB database.

To define a J2C authentication from the administrative console:

1. Select **Security** → **Secure administration, applications and infrastructure**. On the right of the security configuration page in the Java Authentication and Authorization Services section, select **J2C Authentication data**.

**Note:** Most resource configuration pages also contain a shortcut to the J2C authentication alias page.

Also there is no scope selection required. J2C aliases are available at the cell scope.

2. Click **New** to create a new alias.
3. Enter an alias name and enter the user ID and password that are required to access the resource, as shown in Figure 6-4. Click **OK**. Save the configuration.

The screenshot shows the 'JAAS - J2C authentication' configuration page. The breadcrumb trail is 'Secure administration, applications, and infrastructure > JAAS - J2C authentication'. Below the breadcrumb, a description reads: 'Specifies a list of user identities and passwords for Java(TM) 2 connector security to'. The 'Configuration' tab is selected. Under the 'General Properties' section, there are four fields: 'Alias' with the value 'OrderMgmtAuthAlias', 'User ID' with the value 'dbadmin', 'Password' with masked characters '••••••', and an empty 'Description' field. At the bottom, there are four buttons: 'Apply', 'OK', 'Reset', and 'Cancel'.

Figure 6-4 J2C Authentication alias properties

When the alias is stored, the name that you specify is appended to the node name for the deployment manager. For example, in this case, the node name for

the deployment manager is kpmgyw0CellManager01. The name the alias is stored under is kpmgyw0CellManager01/OrderManagementAuthAlias.

### 6.3.3 Defining JDBC resources

JDBC providers and data sources provide access to databases. A JDBC provider configuration contains a pointer to the implementation classes for a specific vendor database. Multiple data source configurations can be associated with a JDBC provider, one for each database of that type that deployed applications need to access.

To define a database, you need to configure:

- ▶ A JDBC provider that defines the location of the Derby driver classes and connection pool properties.
- ▶ A data source that defines the database properties. The configuration includes a pointer to a J2C authentication alias.
- ▶ A J2C authentication alias that contains the credentials needed to access the database.

#### **Example: JDBC resources for DBMSServiceMediation**

The DBMSServiceMediation application in the Order Management System accesses a Derby database called ORDERDB. Because the mediation is deployed to a cluster, you need an Apache Derby Network Server JDBC provider. This type of provider is actually configured using a DB2 Universal JDBC Driver Provider (XA) JDBC provider.

**Attention:** A DB2 JDBC provider is needed for Derby Network Server. While a Derby JDBC provider is used for accessing a Derby Embedded database. The DB2 JDBC provider is compatible with Derby Network Server.

**Creating a JDBC provider to access the Derby Network provider**

To create a JDBC provider using the administrative console:

- 1. Select **Resources** → **JDBC** → **JDBC Providers**.
- 2. Select the scope and click **New**.

In the Order Management System environment, conventions are used to ensure that products are installed on every system in the same location. We also have an environment that consists entirely of Windows systems. Thus, the DB2 product is installed in the same location on every system and we can use the cell scope. If we add new systems later that reside on different platforms, we can create a new JDBC provider at the node scope for that system type. See Figure 6-5.

☐ Scope: Cell=**xpxv812Cell01**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, [see the scope settings help](#)

Cell=xpxv812Cell01 ▾

⊕ Preferences

New Delete

☑

📄

↕

🔄

| Select  | Name ▾      | Scope ▾           | Description ▾ |
|---------|-------------|-------------------|---------------|
|         | Filter: xx* |                   |               |
| None    |             |                   |               |
| Total 1 |             | Filtered total: 0 |               |

Figure 6-5 Create a new JDBC provider

3. Select the database type and the provider options, as shown in Figure 6-6. Click **Next**.

→ **Step 1: Create new JDBC provider**

Step 2: Enter database class path information

Step 3: Summary

### Create new JDBC provider

Set the basic configuration values of a JDBC provider, which encapsulates the specific vendor JDBC driver implementation classes that are required to access the database. The wizard fills in the name and the description fields, but you can type different values.

Scope  
cells:pxv812Cell01

\* Database type  
DB2

\* Provider type  
DB2 Universal JDBC Driver Provider

\* Implementation type  
XA data source

\* Name  
DB2 driver for ORDERDB Derby Network database

Description  
JDBC Provider for the ORDERDB

Next Cancel

Figure 6-6 Select the provider type

4. Enter the information that is required for the provider to find the driver classes, as shown in Figure 6-7 on page 203. This panel contains default properties based on the selections made in the previous dialog box.

Variables are used in the default specifications. Depending on the provider that you selected, you might have the option to set the values for them on this panel. If there are variables used that you cannot set on this panel, or you are not sure how setting the values here might affect other applications, you can configure these variables after you have created the JDBC provider. Variables can be viewed and edited in the console by selecting **Environment** → **WebSphere variables**.

Click **Next**.

The screenshot shows a wizard window titled "Enter database class path information". On the left, a sidebar lists three steps: "Step 1: Create new JDBC provider", "Step 2: Enter database class path information" (which is highlighted with a yellow arrow), and "Step 3: Summary". The main area contains instructions: "Set the environment variables that represent the JDBC driver class files, which WebSphere(R) Application Server uses to define your JDBC provider. This wizard page displays the file names; you supply only the directory locations of the files. Use complete directory paths when you type the JDBC driver file locations. For example: /home/db2inst1/sqllib/java on Linux(TM). If a value is specified for you, you may click Next to accept the value." Below this, there are three sections: "Class path:" with a text area containing three lines of paths using environment variables like \${DB2UNIVERSAL\_JDBC\_DRIVER\_PATH}; "Directory location for 'db2jcc.jar, db2jcc\_license\_cisuz.jar' which is saved as WebSphere variable \${DB2UNIVERSAL\_JDBC\_DRIVER\_PATH}:" with a text area containing the path C:/WebSphere/ProcServer/universalDriver\_wbi/lib; and "Native library path" with a text area containing the variable \${DB2UNIVERSAL\_JDBC\_DRIVER\_NATIVEPATH}. At the bottom are three buttons: "Previous", "Next", and "Cancel".

Step 1: Create new JDBC provider

→ **Step 2: Enter database class path information**

Step 3: Summary

### Enter database class path information

Set the environment variables that represent the JDBC driver class files, which WebSphere(R) Application Server uses to define your JDBC provider. This wizard page displays the file names; you supply only the directory locations of the files. Use complete directory paths when you type the JDBC driver file locations. For example: /home/db2inst1/sqllib/java on Linux(TM). If a value is specified for you, you may click Next to accept the value.

Class path:

```
${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar
${UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar
${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cisuz.jar
```

Directory location for "db2jcc.jar, db2jcc\_license\_cisuz.jar" which is saved as WebSphere variable \${DB2UNIVERSAL\_JDBC\_DRIVER\_PATH}

```
C:/WebSphere/ProcServer/universalDriver_wbi/lib
```

Native library path

Directory location which is saved as WebSphere variable \${DB2UNIVERSAL\_JDBC\_DRIVER\_NATIVEPATH}

Previous Next Cancel

Figure 6-7 Enter the implementation class information

5. Review your settings and click **Finish**. The new provider displays in the list of JDBC providers, as shown in Figure 6-8. Save your configuration.

| <div>NewDelete</div>                                               |                                                               |                   |                               |
|--------------------------------------------------------------------|---------------------------------------------------------------|-------------------|-------------------------------|
| <div><div><div></div><div></div><div></div><div></div></div></div> |                                                               |                   |                               |
| Select                                                             | Name                                                          | Scope             | Description                   |
| Filter: JDBC Provider for the ORDERDB                              |                                                               |                   |                               |
| <input type="checkbox"/>                                           | <a href="#">DB2 driver for ORDERDB Derby Network database</a> | Cell=xxv812Cell01 | JDBC Provider for the ORDERDB |
| Total 2    Filtered total: 1                                       |                                                               |                   |                               |

Figure 6-8 JDBC Provider at cell level for the example application

### ***Creating a data source for the ORDERDB database***

Now that the provider is in place, you need to configure a data source for the database.

To create a new data source:

1. Select **Resources** → **JDBC** → **Data sources**.
2. Set the scope that you created the JDBC provider at and click **New**.
3. Enter the basic properties for the data source (as shown in Figure 6-9 on page 205):
  - A name for the data source.
  - The JNDI name that the applications use to access the database
  - The authentication alias that contains the credentials for the database.  
Note that when the authentication alias was created the name you provided has been prepended with the cell name.

Click **Next**.

**Step 1: Enter basic data source information**

Step 2: Select JDBC provider

Step 3: Enter database specific properties for the data source

Step 4: Summary

### Enter basic data source information

Set the basic configuration values of a data source for association with your JDBC provider. A data source supplies the physical connections between the application server and the database.

Requirement: Use the Data sources (WebSphere(R) Application Server V4) console pages if your applications are based on the Enterprise JavaBeans (TM) (EJB) 1.0 specification or the Java(TM) Servlet 2.2 specification.

Scope  
cells:pxw812Cell01

\* Data source name  
ORDERDB\_Derby\_viaDB2\_cell

\* JNDI name  
jdbc/DerbyORDERDB

**Component-managed authentication alias and XA recovery authentication alias**

Select a component-managed authentication alias. The selected authentication alias will also be set as the XA recovery authentication alias if your JDBC Provider supports XA. If you choose to [create a new J2C authentication alias](#), the wizard will be canceled.

pxw812CellManager01/OrderMgmtAuthAlias

Next Cancel

Figure 6-9 Enter the basic properties for the data source

4. Select the JDBC provider as shown in Figure 6-10 and click **Next**.

Step 1: Enter basic data source information

**Step 2: Select JDBC provider**

Step 3: Enter database specific properties for the data source

Step 4: Summary

### Select JDBC provider

Specify a JDBC provider to support this data source.

☐ Create new JDBC provider

☒ Select an existing JDBC provider

DB2 driver for ORDERDB Derby Network database

Previous Next Cancel

Figure 6-10 Select the JDBC provider

5. Enter the database properties (Figure 6-11):
  - a. The server address and port must correspond to the database server. Typically, the port for a DB2 server is 50000. In this case, we specify the address and port of the Derby Network Server.
  - b. The database name. For Derby locations, the name corresponds to the file location of the database. Click **Next**.

Step 1: Enter basic data source information

Step 2: Select JDBC provider

→ Step 3: Enter database specific properties for the data source

Step 4: Summary

### Enter database specific properties for the data source

Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through this data source.

\* Database name  
C:\ITSO\sampleDB\ORDERDB

\* Driver type  
4 ▼

\* Server name  
xpxv812.itso.ral.ibm.com

\* Port number  
1527

☒ Use this data source in container managed persistence (CMP)

Previous Next Cancel

Figure 6-11 Specify the database properties

6. Review your settings and click **Finish**.
7. Save the configuration and test the connection using the **Test connection** button.

### 6.3.4 Defining users and groups

If your application contains human tasks, you need to populate the user registry with the users and groups used by the application.

A user registry, referred to as a *repository*, contains information about users and groups. This information is used during run time to perform security-related functions, including authentication and authorization. WebSphere Application Server based run times support the following user repository types:

- ▶ Local operating system
- ▶ Stand-alone LDAP registry
- ▶ Stand-alone custom registry
- ▶ Federated repositories, which can include a file-based repository that is managed by WebSphere

Federated repositories enable you to use multiple repositories, which can be file-based repositories, LDAP repositories, or a sub-tree of an LDAP repository. These repositories are combined under a single realm. All repositories are active, and user IDs must be unique within the federated repositories. You can add more user registries to the federated repository using the administrative console. Users and groups in a federated repository can be managed using the administrative console.

**Note:** WebSphere Application Server federated repositories do not support a z/OS LDAP server with an SDBM back-end (resource access control facility (RACF)).

You can configure security repositories from the administrative console by selecting **Security** → **Secure administration, applications and infrastructure**.

You can use the Security Configuration wizard to define your configuration. Alternatively, you can use the options in the “User account repository” area to configure and select an active repository.

**Note:** Enabling administrative security during profile creation creates a federated repository with one file-based repository automatically and makes this the active repository.

When the repository is in place, define the users and groups expected by the human task.

Security Configuration Wizard

Security Configuration Report

**Administrative security**

☒ Enable administrative security
 

[Administrative User Roles](#)
[Administrative Group Roles](#)

**Application security**

☒ Enable application security

**Java 2 security**

☐ Use Java 2 security to restrict application access to local resources
 

☒ Warn if applications are granted custom permissions
 ☐ Restrict access to resource authentication data

**User account repository**

Current realm definition

Available realm definitions  

Configure

Set as current

**Authentication**

☐ Use domain-qualified users
 

☒ Web security
 ☒ RMI/IIOP security
 ☒ Java Authentication and Authorization
 

[Authentication mechanism](#)

[External authorization provider](#)
[Custom properties](#)

Figure 6-12 Configure repositories

### Example: Federated repository for Order Management System

In the Order Management System, the human task requires that you add users and groups to the repository. Because we are using a federated repository in this example, we use the administrative console.

#### Groups

Start by creating groups that hold users with identical authorities as follows:

1. In the administrative console, select **Users and Groups** → **Manage groups** to create new groups and search for existing groups. Figure 6-13 shows the groups that are used in the Order Management System solution.

Manage Groups

Search for Groups

Search by

\* Search for

\* Maximum results

Group name

\*

100





Search

4 groups matched the search criteria.

Create...

Delete

Select an action...



| Select                   | Group name                       | Description | Unique Name                                    |
|--------------------------|----------------------------------|-------------|------------------------------------------------|
| <input type="checkbox"/> | <a href="#">Financial Office</a> |             | cn=Financial Office,o=defaultWIMFileBasedRealm |
| <input type="checkbox"/> | <a href="#">Order Management</a> |             | cn=Order Management,o=defaultWIMFileBasedRealm |
| <input type="checkbox"/> | <a href="#">Warehouse Office</a> |             | cn=Warehouse Office,o=defaultWIMFileBasedRealm |
| <input type="checkbox"/> | <a href="#">managers</a>         |             | cn=managers,o=defaultWIMFileBasedRealm         |

Page 1 of 1

Total: 4

Figure 6-13 Manage groups in the federated repository

2. Click a group name to display or add members (Figure 6-14).




**Manage Groups**



**Group Properties**

**General** | **Members** | **Groups**

Group name

The group has 2 members.

**Add Users...** **Add Groups...** **Remove**   

| Select                   | ID      | Type                                                                              | Unique Name                            |
|--------------------------|---------|-----------------------------------------------------------------------------------|----------------------------------------|
| <input type="checkbox"/> | janedoe |  | uid=janedoe,o=defaultWIMFileBasedRealm |
| <input type="checkbox"/> | johndoe |  | uid=johndoe,o=defaultWIMFileBasedRealm |

Page 1 of 1 Total: 2

Figure 6-14 Group properties

- To add users to the repository, select **Users and Groups** → **Manage users**. Figure 6-15 shows the users that are defined for the Order Management System solution.

Manage Users

Search for Users

Search by

\* Search for

\* Maximum results

User ID

\*

100

Search

5 users matched the search criteria.

Create...

Delete

Select an action...

| Select                   | User ID                       | First name    | Last name     | E-mail             | Unique ID                      |
|--------------------------|-------------------------------|---------------|---------------|--------------------|--------------------------------|
| <input type="checkbox"/> | <a href="#">Administrator</a> | Administrator | Administrator |                    | uid=Administrator,o=defaultWIM |
| <input type="checkbox"/> | <a href="#">janedoe</a>       | Jane          | Doe           | janedoe@itso.com   | uid=janedoe,o=defaultWIM       |
| <input type="checkbox"/> | <a href="#">janesmith</a>     | Jane          | Smith         | janesmith@itso.com | uid=janesmith,o=defaultWIM     |
| <input type="checkbox"/> | <a href="#">johndoe</a>       | John          | Doe           | johndoe@itso.com   | uid=johndoe,o=defaultWIM       |
| <input type="checkbox"/> | <a href="#">sadtler</a>       | carla         | sadtler       |                    | uid=sadtler,o=defaultWIM       |

Page 1 of 1

Total: 5

Figure 6-15 Users defined to the federated repository

4. Click a user name to display or change the properties for the user (Figure 6-16).

The screenshot shows a 'Manage Users' window with a 'User Properties' section. Inside, there are two sub-tabs: 'General' and 'Groups'. The 'Groups' sub-tab is active. The form contains the following fields:

- \*User ID:
- \*First name:
- \*Last name:
- E-mail:
- Password:
- Confirm password:

At the bottom of the form are three buttons: **OK**, **Apply**, and **Cancel**.

Figure 6-16 User properties

5. Click the Groups tab to display the groups to which the user belongs and to add or remove the user from groups (Figure 6-17).

The screenshot shows a 'Manage Users' window with a 'User Properties' section. The 'Groups' tab is selected. The 'User ID' field contains 'janedoe'. Below it, a message states 'The user is a member of 2 groups.' A table lists the groups: 'Financial Office' and 'managers', each with an unchecked 'Select' checkbox. The table has a footer showing 'Page 1 of 1' and 'Total: 2'.

| Select                   | Group name       |
|--------------------------|------------------|
| <input type="checkbox"/> | Financial Office |
| <input type="checkbox"/> | managers         |

Page 1 of 1      Total: 2

Figure 6-17 User properties: Group memberships

### 6.3.5 JMS resources

When an import or export uses a messaging binding (JMS or MQ), it assumes that corresponding queues or topics have been created on the underlying messaging provider.

#### MQ bindings

When you create an MQ binding, you specify the information required to connect to the WebSphere MQ queue manager and the queue. You must define these resources in the WebSphere MQ installation.

#### JMS bindings

When you create a JMS binding, you specify the JNDI name of the send and receive queues and topics. You must define two types of resources:

1. The queues or topics must be defined to the messaging provider.

If you are using the WebSphere default messaging provider, queues and topics are defined as destinations on the service integration bus. You can use

the default SCA application bus that is created with the deployment environment, or you can create a new bus. Buses and destinations are created from the administrative console:

- Select **Service** → **Integration Buses** to view or create new buses.
  - Select **Service** → **Integration Buses** → *bus\_name* → **Destination** to view or create new destinations.
2. JMS queues or topics must be defined as resources to WebSphere Process Server. The JMS resource points to the queue or topic and specifies the JNDI name used to access it. This JNDI name must match the one specified in the binding.
    - Select **Resources** → **JMS** → **JMS providers** to view or create JMS providers.
    - Select **Resources** → **JMS** → *resource\_type* to view or create JMS queues and topics.

### MQ JMS bindings

The underlying JMS provider must have the queues, topics, and JNDI entries required by the binding defined.

## 6.4 Deploying applications

Modules and mediation modules are packaged for deployment by WebSphere Integration Developer in the form of an enterprise archive (EAR) file. You can deploy modules that are exported as EAR files to a WebSphere Process Server application server or cluster. Mediation modules exported as EAR files can be deployed to a WebSphere Process Server or WebSphere ESB application server or cluster.

A typical sequence for deploying applications is:

1. Export the modules from WebSphere Integration Developer as EAR files ready for deployment.
2. Deploy the application to the appropriate cluster, also assigning it to the Web server.
3. Configure the binding properties for the import components in the module when applicable.
4. Generate the configuration file for the Web server plug-in and propagate it to the Web server system.

## 6.4.1 Exporting applications for deployment

You can export applications in the workspace as EAR files for deployment using the Export wizard.

### Example: Exporting Order Management System

To export the Order Management System:

1. Open the workspace with all the WebSphere Process Server and WebSphere ESB components
2. Select **File** → **Export** → **Business Integration** → **Integration module**, and click **Next**.
3. Click **Select All**.
4. Under Export usage, select **EAR files for server deployment** as shown in Figure 6-18.

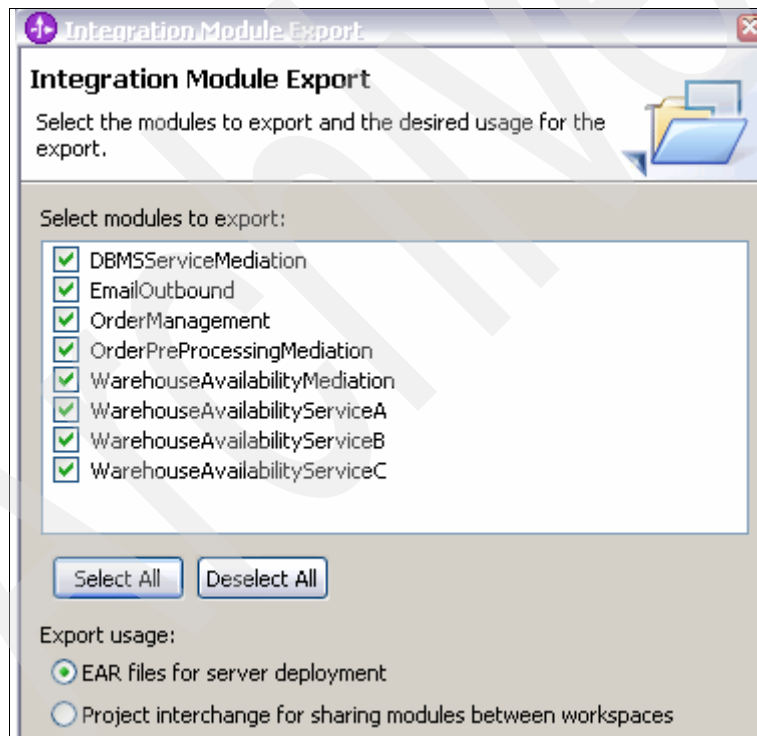


Figure 6-18 Export projects as EAR files

5. Click **Next**.
6. Select a Target directory using **Browse**. Then, click **Finish**.

You can now use the EAR files to deploy to an external server.

## 6.4.2 Installing applications

The administrative console provides a wizard that leads you through the process of installing an application. To install an EAR file, it must be available to the deployment manager. The deployment manager can access an EAR file on the administrator's local system (where the Web browser is) or on a file system that resides on a node defined to the cell.

The basic process to deploy an application is as follows:

1. Log in to the administrative console for WebSphere Process Server and select **Enterprise Application** → **Install New Application**.
2. Select the EAR file to deploy, as shown in Figure 6-19. Note that “Local file system” is relative to the Web browser. The “Remote file system” allows you to find the EAR file on any of the nodes that are defined to the cell. Select the option to Prompt only when additional information is required. Click **Next**.

**Preparing for the application installation**

Specify the EAR, WAR, JAR, or SAR module to upload and install.

**Path to the new application**

☒ Local file system

Full path  
eAvailabilityServiceA.ear

☐ Remote file system

Full path

Context root  
 Used only for standalone Web modules (.war files) and SIP modules (.sar files)

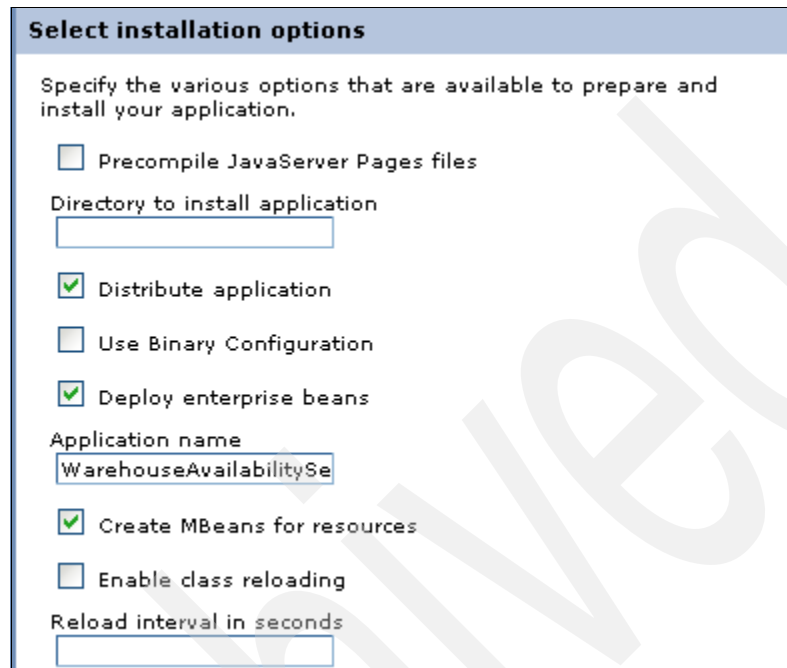
**How do you want to install the application?**

☒ Prompt me only when additional information is required.

☐ Show me all installation options and parameters.

Figure 6-19 Select the EAR file

3. Select various installation options in the next panel, as shown in Figure 6-20. For this example, take the defaults, and click **Next**.



**Select installation options**

Specify the various options that are available to prepare and install your application.

☐ Precompile JavaServer Pages files

Directory to install application

☒ Distribute application

☐ Use Binary Configuration

☒ Deploy enterprise beans

Application name

☒ Create MBeans for resources

☐ Enable class reloading

Reload interval in seconds

Figure 6-20 Select installation options

4. Select the modules in the application, as shown in Figure 6-21 on page 218. Map all the modules to the target deployment cluster. For example, in this case, we map a mediation module to the WebSphere ESB cluster. Map all the Web modules to both Web server node as well. (There are no issues with mapping all the modules.)

The simplest method to do map the modules is to use the Ctrl key to select both the cluster and the Web server. Then, select all the modules. Click **Apply**.

### Map modules to servers

Specify targets such as application servers or clusters of application servers where you want to install the application. Modules can be installed on the same application server or dispersed among multiple application servers as targets that serve as routers for requests to this application. The plug-in configuration is based on the applications that are routed through.

Clusters and Servers:

WebSphere:cell=kpmgyw0Cell01,cluster=default.Support  
WebSphere:cell=kpmgyw0Cell01,cluster=default.Messaging  
WebSphere:cell=kpmgyw0Cell01,cluster=default.AppTarget  
WebSphere:cell=kpmgyw0Cell01,cluster=WESB.AppTarget  
WebSphere:cell=kpmgyw0Cell01,node=WebServerNode,server=webserver1

Apply

☒ ☐

| Select                              | Module                           | URI                                                       |
|-------------------------------------|----------------------------------|-----------------------------------------------------------|
| <input checked="" type="checkbox"/> | WarehouseAvailabilityServiceAEJB | WarehouseAvailabilityServiceAEJB.jar,META-INF/ejb-jar.xml |
| <input checked="" type="checkbox"/> | WarehouseAvailabilityServiceAWeb | WarehouseAvailabilityServiceAWeb.war,WEB-INF/web.xml      |

Figure 6-21 Map modules to servers

- Click **Next**.
- Review the summary of your options, and click **Finish**.
- Save the configuration.
- Perform any necessary post-installation tasks that are required to update import binding attributes to reflect the production environment.
- Regenerate the plug-in configuration file, and propagate it to the Web server.
- Start the application. Select **Applications** → **Enterprise Applications**. Select the application, and click **Start**.

### 6.4.3 Configure and test Web service URLs

After you deploy a module with a Web service interface, you can publish the associated WSDL file from the application server. The WSDL file includes the information about the service, including the URL that is used by clients to access the service.

You can change the URL prefix of the service (`http://host:port`) published with the WSDL through the administrative console:

1. Select **Applications** → **Enterprise Applications** → **web\_service\_application** → **Provide HTTP endpoint URL information**.
2. Click **Select default HTTP URL prefix** and select the prefix from the drop-down menu (as shown in Figure 6-22). Your options include URLs with the ports that are defined for the server and for the virtual host.

Figure 6-22 Changing the HTTP URL prefix

3. Select the module, and click **Apply**.
4. Click **OK**, and save the configuration.

Now, you can publish the WSDL file for the service for distribution to clients and for testing:

1. Select **Applications** → **Enterprise Applications** → **application** → **Web service properties** → **Publish WSDL files**.
2. Extract the WSDL file from the compressed file that is generated. Note the service port URL in the WSDL. See Example 6-1.

*Example 6-1 WSDL soap address*

---

```
<soap:address
location="http://xpxv812.itso.ral.ibm.com:9080/WarehouseAvailabilityServiceAWeb/sca/WarehouseAvailabilityServiceIFExport1"/>
```

---

3. Test the URL in a Web browser to ensure connectivity to the service directly through the application server, as shown in Figure 6-23.



Figure 6-23 Test the Web service interface

4. Regenerate the Web server plug-in and propagate it to the Web server. (We discuss this topic later in 6.5, "Generating the Web server plug-in configuration" on page 226).
5. Test the URL again, this time substituting the address of the Web server for the URL prefix to ensure connectivity to the service through the Web server.

**Note:** If you cannot connect to the Web service, check the virtual host configuration:

1. Determine the virtual host to which the Web service application was deployed.  
Select **Applications** → **Enterprise applications**. Click the application name to open the configuration. Click **Virtual hosts** to see the virtual host name.
2. Check for this virtual host configuration to make sure it includes the port used to access the Web service.  
Select **Environment** → **Virtual Hosts**. Click the virtual host name to open the configuration. Click **Host Aliases** to see a list of ports for this virtual host.
3. If necessary, update the virtual host configuration to include the port. Save the configuration and restart the environment (servers, nodes, deployment manager), then retry.

## 6.4.4 Updating endpoint URLs in Web service bindings

In a development environment, the URL that is specified in a Web service binding might differ from the URL in a production environment. You can update the endpoint URL for a Web service binding using the following method:

1. Select **Applications** → **SCA Modules** → *module\_name* → **Module components** → **Imports** → *import\_name* → **Binding** → **Web service**.
2. Enter the new endpoint URL and save the configuration.

## 6.4.5 Example: Installing the Order Management System applications

In the Order Management System, the following applications are deployed to the WebSphere ESB cluster:

- ▶ DBMSServiceMediationApp
- ▶ OrderPreProcessingMediationApp
- ▶ WarehouseAvailabilityMediationApp
- ▶ WarehouseAvailabilityServiceAApp
- ▶ WarehouseAvailabilityServiceBApp
- ▶ WarehouseAvailabilityServiceCApp

In addition, the following applications are deployed to the WebSphere Process Server cluster:

- ▶ EmailOutboundApp
- ▶ OrderManagementApp

### DBMSServiceMediation deployment

The DBMSServiceMediation is deployed to the WebSphere ESB cluster.

To deploy this application from the administrative console:

1. Select **Enterprise Application** → **Install New Application** wizard to start the installation.
2. Take the defaults with the exception of the module mapping (as shown in Figure 6-24 on page 222):
  - a. Map every module to the WebSphere Enterprise Service Bus cluster, WESB.AppTarget.
  - b. Map the Web module to the Web server, webserver1. Again, this is because the mediation provides its services through a Web service interface and this access must be managed by the Web server plug-in to apply the cluster workload and routing policies.

Clusters and Servers:

WebSphere:cell=xxv812Cell01,cluster=default.AppTarget  
 WebSphere:cell=xxv812Cell01,cluster=WESB.AppTarget  
 WebSphere:cell=xxv812Cell01,node=HTTPServerNode,server=webserver1

Remove Update Remove File Export File

Apply

| Select                              | Module                                         | URI                                              | Module Type      | Server    |
|-------------------------------------|------------------------------------------------|--------------------------------------------------|------------------|-----------|
| <input type="checkbox"/>            | <a href="#">DBMSServiceMediationEJB</a>        | DBMSServiceMediationEJB.jar,META-INF/ejb-jar.xml | EJB Module       | WebSphere |
| <input checked="" type="checkbox"/> | <a href="#">DBMSServiceMediationWeb</a>        | DBMSServiceMediationWeb.war,WEB-INF/web.xml      | Web Module       | WebSphere |
| <input type="checkbox"/>            | <a href="#">IBM WebSphere Adapter for JDBC</a> | CWYBC_JDBC.rar,META-INF/ra.xml                   | Connector Module | WebSphere |

Figure 6-24 Module to server mapping for the DBMS Mediation

3. Complete the installation and save the configuration.
4. Generate the Web server plug-in, and propagate it to the Web server. If you are deploying multiple applications, wait until they are all deployed and then perform this step one time.
5. Start the application.
6. Test the Web service URL for the mediation. See 6.4.3, "Configure and test Web service URLs" on page 218.

## Warehouse Availability Systems deployment

The Warehouse Availability Systems must host warehouse Web services that provide the availability status of an item in a warehouse. These Web services in our test system have been emulated as mediation modules and must run on a WebSphere ESB capable server. These modules are deployed to the WebSphere ESB cluster.

The steps to prepare this system are:

1. Deploy the Warehouse Availability System applications to the WebSphere ESB cluster using the same method you saw in the previous step.
2. Determine the endpoint URL for each system and test the connectivity.

The mediation that uses these services needs the complete URL from the port definition in the WSDL file for the service:

```
http://xpxv812.itso.ra1.ibm.com:9082/WarehouseAvailabilityServiceAWeb/sca/WarehouseAvailabilityServiceIFExport1
```

Repeat the deployment, URL/port discovery, and testing from the WebSphere ESB client machines for *every* Web service that is accessed from the mediation module. In our case, they are the three Warehouse Availability Systems Web services, so this procedure has been repeated three times.

## WarehouseAvailabilityMediation deployment

The WarehouseAvailabilityMediation deployment requires additional steps after deployment to ensure it can access the WarehouseAvailabilitySystems. To deploy:

1. Deploy WarehouseAvailabilityMediation to the WebSphere ESB cluster.
2. WarehouseAvailabilityMediation uses the WarehouseAvailabilitySystems Web services. When those Web services are deployed, you discover and test the URLs that are used to access them. You must make sure that these URLs are used in the import properties for WarehouseAvailabilityMediation. See Figure 6-25 on page 224.

For example, assume the URL for WarehouseAvailabilitySystemA Web service URL is as follows:

```
http://xpxv812.itso.ra1.ibm.com:9082/WarehouseAvailabilityServiceAWeb/sca/WarehouseAvailabilityServiceIFExport1
```

Find the URL that is actually used by WarehouseAvailabilityMediation for the service by selecting **Applications** → **SCA Modules** →

**WarehouseAvailabilityMediation, Module components** → **Imports** → **GetWarehouseA\_Availability** → **Binding** → **Web service**.

**General Properties**

Module: WarehouseAvailabilityMediation

Application name: WarehouseAvailabilityMediationApp

Description:

**Module components**

- Imports
  - GetWarehouseA\_Availability
    - Interfaces
    - Binding
      - Web service [WarehouseAvailabilitySe]
  - GetWarehouseB\_Availability
  - GetWarehouseC\_Availability
- Exports

Figure 6-25 Looking at the WarehouseAvailabilityMediation imports properties

3. Compare the URL for the Web service application with the URL that you find in the endpoint field (shown in Figure 6-26). If they are not the same, update the endpoint URL.

**General Properties**

Service: WarehouseAvailabilityServiceIFExport1\_WarehouseAvailabilityServiceIFHttpService

Port: WarehouseAvailabilityServiceIFExport1\_WarehouseAvailabilityServiceIFHttpPort

\* Endpoint: http://xpxv812.itso.ral.ibm.com:9082/WarehouseAvailabilityServiceAWeb/sca/Wa

Apply OK Reset Cancel

Figure 6-26 Checking and updating the import endpoint property

Repeat steps 1 through 3 for all three Web services imports.

**Note:** The Web server address in the figure shows the URL of the Web container of a specific server in a cluster, so no load balancing occurs in this example. We did this to simplify our lab environment.

4. Finally, generate and propagate the plug-in for the Web server, and test that the WarehouseAvailabilityMediation module is accessible through the Web server.

## OrderManagement deployment

Deploying the OrderManagement application is similar to the process that you used to deploy the mediation modules as follows:

1. Deploy the OrderManagement application, mapping the modules to the WebSphere Process Server cluster. In addition, map the Web module to the Web server. See Figure 6-27.

| Select                              | Module                             | URI                                         | Module Type | Server                                                             |
|-------------------------------------|------------------------------------|---------------------------------------------|-------------|--------------------------------------------------------------------|
| <input type="checkbox"/>            | <a href="#">OrderManagementEJB</a> | OrderManagementEJB.jar,META-INF/ejb-jar.xml | EJB Module  | WebSphere:cell=xpxv812Cell01,cluster=WESB.AppTarget                |
| <input checked="" type="checkbox"/> | <a href="#">OrderManagementWeb</a> | OrderManagementWeb.war,WEB-INF/web.xml      | Web Module  | WebSphere:cell=xpxv812Cell01,node=HTTPServerNode,server=webserver1 |

Figure 6-27 OrderManagement application modules mapping

2. The OrderManagement application uses the DBMSServiceMediation and WarehouseAvailabilityMediation applications through imports. Determine the URL for each of these mediations, this time using the Web server so that load balancing to the mediation modules occurs. Then, update the endpoint properties for each Web service import in the OrderManagement application with the appropriate URL (as shown in Figure 6-28).

**General Properties**

Service  
CustomerServiceIFExport1\_CustomerServiceIFHttpService

Port  
CustomerServiceIFExport1\_CustomerServiceIFHttpPort

\* Endpoint  
http://xpxv812.itso.ral.ibm.com/DBMSServiceMediationWeb/sca/CustomerServiceIFExport1\_CustomerServiceIFHttpService

Apply OK Reset Cancel

Figure 6-28 Checking and updating OrderManagement imports endpoint property

3. Generate and propagate the Web server plug-in and then test the service availability using a browser.

### **OrderPreProcessingMediation deployment**

The last module to deploy is the OrderPreProcessingMediation application as follows:

1. Deploy the module mapping all of the modules to the WebSphere ESB cluster and the Web module to the Web server.
2. Update the import for the OrderManagement module. The Flat File Adapter import is not a Web service binding and needs no update.
3. Generate and propagate the plug-in.
4. Test access to the service.

## **6.5 Generating the Web server plug-in configuration**

After a new application is deployed, you need to generate the Web server plug-in configuration and propagate it to the Web server to tell the Web server how to forward requests for the application. To generate the plug-in from the administrative console, select **Servers** → **Web servers**. Click **Generate Plug In** to create the plug-in file.

How you propagate the new configuration file depends on the Web server type and your environment. You can propagate the file manually or, in some cases, by selecting **Propagate plug in**.

## **6.6 Testing the application**

The last phase that you go through is testing. This section provides general guidance using our OrderManagement scenario as an example.

A logical way to test is to use a “bottom up” method, performing incremental tests starting with the last service in the chain and working your way to the first service. In our scenario, looking at the deployment diagram in Figure 6-3 on page 194, a possible test order is:

1. DBMSServiceMediation
2. WarehouseAvailabilityMediation
3. OrderManagement
4. OrderPreProcessingMediation

DBMSServiceMediation and WarehouseAvailabilityMediation do not depend on other modules, while OrderManagement uses imports to access these mediations. This test order allows you to make sure the DBMSServiceMediation works as expected before testing a module that uses it.

The way you test a module depends on several factors such as:

- ▶ How the module is accessed (the export binding)  
You must use a client that matches the module access protocols, which can be JMS, HTTP, or Web service (SOAP/HTTP or SOAP/JMS).
- ▶ The tool you use for developing or emulating the client of the module  
You can use any appropriate development tool to create the client of the application. Rational Application Developer or WebSphere Integration Developer can be appropriate in most cases.
- ▶ The module type (business integration versus mediation)  
Business integration modules can be easily tested using the Business Process Choreographer Explorer. This tool cannot be used to test mediation modules.
- ▶ The network accessibility and security restrictions  
You must ensure your client has network connectivity with the modules and complies with security constraints (such as message encryption or credentials to be provided).

We provide some ideas for testing, based on our testing plan for the Order Management system.

### 6.6.1 Starting and preparing your test clients

The scenario application modules are accessible as Web services so we used Web service clients for testing. With the WSDL files for the services and the correct endpoint URLs and ports, you can build a Web service client for each service.

Because we developed all of the modules using WebSphere Integration Developer, we use that also for most of the testing. Our strategy is as follows:

1. We created an OrderTestClient mediation module as a client.  
This client can also be a business integration module, because it contains only import components.
2. We added the service interfaces and WSDL files for the modules to be tested in the OrderTestClient module.

Most of the interfaces for the OrderManagement system are contained in OrderManagementLib, so we simply made this library a dependency of the new mediation module. Just to test OrderPreProcessingMediation, we made a copy of OrderPreProcessingIF and the Web service port in the mediation module. See Figure 6-29.



Figure 6-29 A mediation module as test client with the components interfaces and Web service ports

3. We defined an import for each module to be tested:

- OrderPreProcessing mediation.
- DBMSServiceMediation. We defined an import for the Customer service; however, we can define additional imports to test the other services that are provided by this module.
- WarehouseAvailabilityMediation.

We added the appropriate interface to these Imports and named them in a way that makes it easy to recognize their purpose. See Figure 6-30 on page 228.



Figure 6-30 Imports components for testing the modules

**Note:** The import for OrderManagement is missing here because we test that component using the Business Process Choreographer Explorer.

4. Then we bound the imports to the Web service port.

Be sure to use Web server address and port 80 for the Web service endpoint property to have load balancing performed for the cluster (Figure 6-31). Test the Web service using a browser to make sure the Web service can be accessed through the Web server.

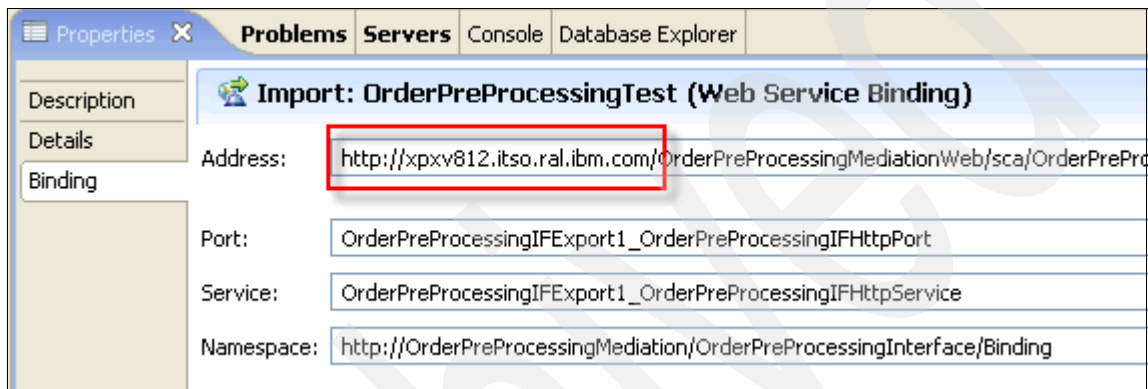


Figure 6-31 Binding Address property must target the Web server address

## 6.6.2 Testing the mediation modules

To test the mediation modules using the OrderTestClient:

1. Start the WebSphere ESB test environment in WebSphere Integration Developer.
2. Deploy OrderTestClient to the server.
3. Right click **OrderTestClient** in the Business Integration view, and select **Test** → **Module** to launch the integration test client.
4. DBMSServiceMediation is tested first. Figure 6-32 shows the properties that we used for the test. Because the test is for the customer retrieval operation, you need only a customer ID value.

**▼ Detailed Properties**

Configuration: Default Module Test

Module: OrderTestClient

Component: DBMSCustomerService

Interface: CustomerServiceIF

Operation: retrieveCustomer

☐ Invoke export using binding

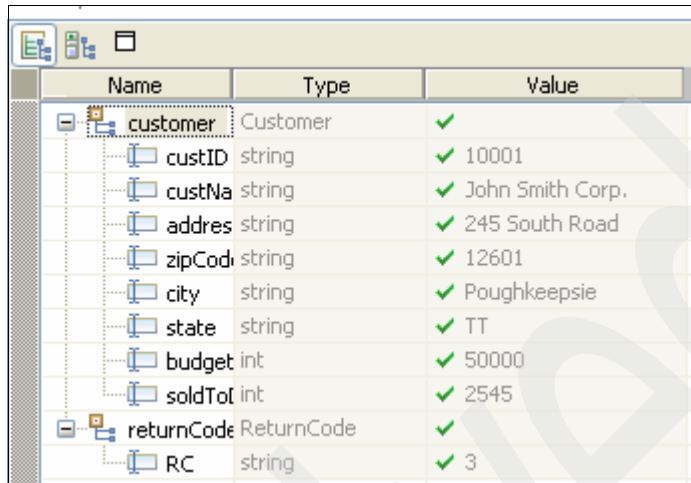
Initial request parameters

| Name     | Type     | Value   |
|----------|----------|---------|
| customer | Customer | ✓       |
| custID   | string   | ✓ 10001 |
| custNa   | string   | ✓       |
| addres   | string   | ✓       |

Figure 6-32 Testing the DBMSServiceMediation Customer Service

**Note:** In our test environment, we found that the first invocation often failed. Retrying the invocation was successful.

The test is successful if the customer fields are populated as shown in Figure 6-33.



| Name       | Type       | Value              |
|------------|------------|--------------------|
| customer   | Customer   | ✓                  |
| custID     | string     | ✓ 10001            |
| custNa     | string     | ✓ John Smith Corp. |
| addres     | string     | ✓ 245 South Road   |
| zipCod     | string     | ✓ 12601            |
| city       | string     | ✓ Poughkeepsie     |
| state      | string     | ✓ TT               |
| budget     | int        | ✓ 50000            |
| soldTot    | int        | ✓ 2545             |
| returnCode | ReturnCode | ✓                  |
| RC         | string     | ✓ 3                |

Figure 6-33 Successful customer service test

5. You can test the WarehouseAvailabilityMediation module using the same method.

### 6.6.3 Testing the business integration modules

Next, you move to OrderManagement testing.

#### Deleting processes automatically

By default, business processes are set to delete instances on successful completion, which might make it difficult to determine whether a test run is successful. To change the setting, open the process. Click in the white space of the canvas to open the properties for the process. You can find the setting in the Details tab of the Properties view, as shown in Figure 6-34 on page 232.

The value that you want for this setting can differ between test and production. For example, during testing we changed the setting so that processes had to be deleted manually using the Business Process Choreographer Explorer. When we deployed to a production environment, we changed the setting so that processes were deleted if they were completed successfully.

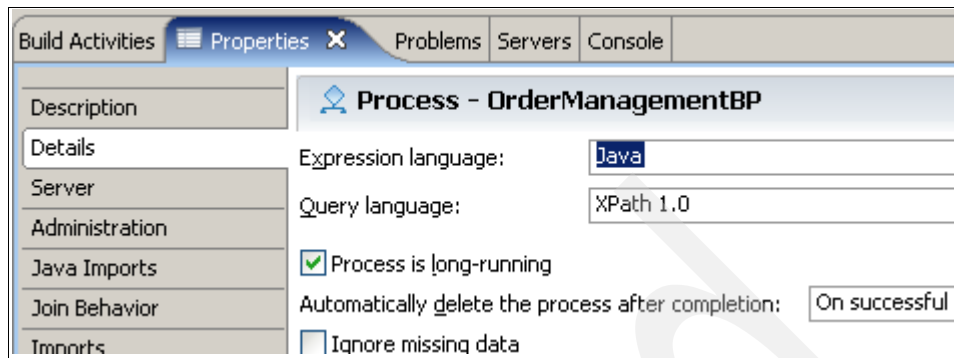


Figure 6-34 Automatic process deletion setting

## Testing OrderManagementBP

To test the OrderManagementBP process:

1. Open a Business Process Choreographer Explorer session, log in as Administrator, and find the **OrderManagementBP** templates.
2. Select the process, and click **Start Instance** (shown in Figure 6-35).

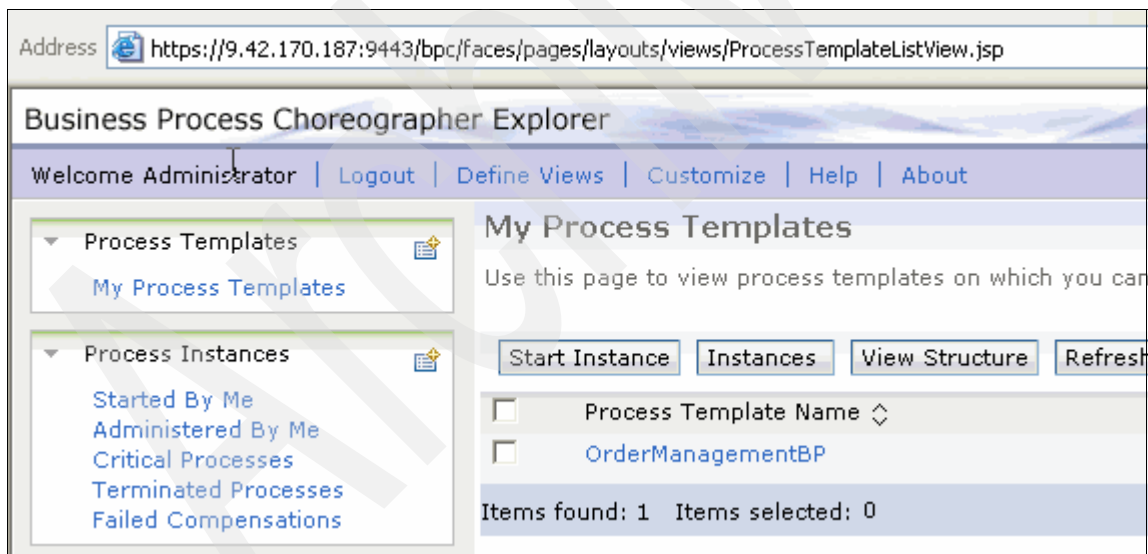


Figure 6-35 OrderManagement template in the Business Process Choreographer Explorer

You can use the following test cases to ensure that the module is working properly:

- ▶ Automatic approval
- ▶ Automatic refusal
- ▶ Human tasks involvement for final decision

The last test case, human tasks, is also the most comprehensive, so we show that test here. Figure 6-36 shows the input data for this test.

Submit

Process Template Name OrderManagementBP  
Operation createOrder

Process Name humanTasks

Process Input Message

**Form View**

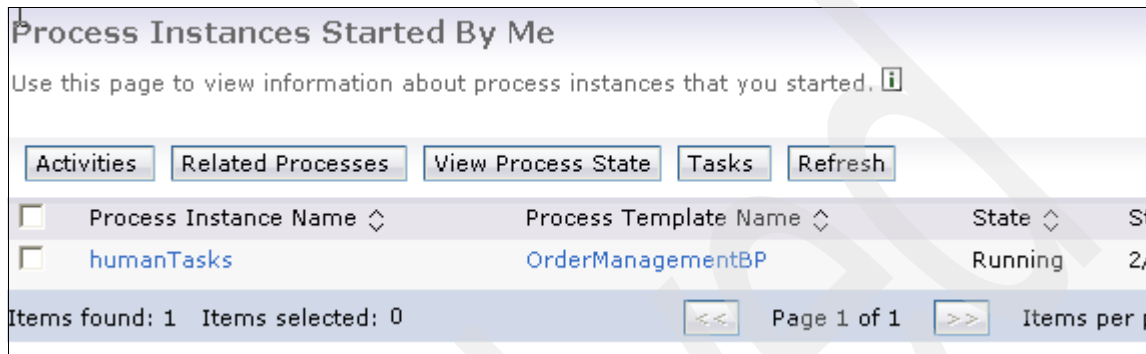
| orderManagementInput |                       |
|----------------------|-----------------------|
| custID               | 10003                 |
| itemID               | IT_003                |
| itemQty              | 15                    |
| submitterID          | user2                 |
| submitterEmail       | user2@xpxv812.itso.ra |

Edit Source

Figure 6-36 Testing the OrderManagement in the human task activation case

Give this instance a specific process name (*humanTasks*) to better recognize it in between the process instances. Then, enter the data, and click **Submit**.

When submitted, the instance remains in the running status in the **Process Instances** → **Started By me** list because the process is suspended while waiting for the human tasks to complete. See Figure 6-37.



Process Instances Started By Me

Use this page to view information about process instances that you started. ⓘ

Activities Related Processes View Process State Tasks Refresh

| <input type="checkbox"/> | Process Instance Name ▾ | Process Template Name ▾ | State ▾ | St |
|--------------------------|-------------------------|-------------------------|---------|----|
| <input type="checkbox"/> | humanTasks              | OrderManagementBP       | Running | 2/ |

Items found: 1 Items selected: 0 << Page 1 of 1 >> Items per p

Figure 6-37 Process instance keep running until you do not complete human task

You can access the human task by logging out of Business Process Choreographer Explorer and logging in again as a member of the Warehouse Office group first, then Financial Office group. Approve both the tasks (copying input data in output data for the warehouse human task and approving) to complete the process successfully.

Figure 6-38 shows the Warehouse human task.

Form View

| item                               | itemID                           | <input type="text" value="IT_003"/>                                                                                                                                                                                                                                                                                                                                                                                                         |  |  |       |       |        |                                    |                                  |                                 |                                    |                                  |                                 |                                    |  |
|------------------------------------|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|-------|-------|--------|------------------------------------|----------------------------------|---------------------------------|------------------------------------|----------------------------------|---------------------------------|------------------------------------|--|
|                                    | itemName                         | <input type="text" value="Item003"/>                                                                                                                                                                                                                                                                                                                                                                                                        |  |  |       |       |        |                                    |                                  |                                 |                                    |                                  |                                 |                                    |  |
|                                    | price                            | <input type="text" value="5"/>                                                                                                                                                                                                                                                                                                                                                                                                              |  |  |       |       |        |                                    |                                  |                                 |                                    |                                  |                                 |                                    |  |
|                                    | warehouses                       | <table><thead><tr><th>whsID</th><th>stock</th><th>indeli</th></tr></thead><tbody><tr><td><input type="text" value="WHS_B"/></td><td><input type="text" value="-22"/></td><td><input type="text" value="67"/></td></tr><tr><td><input type="text" value="WHS_C"/></td><td><input type="text" value="-21"/></td><td><input type="text" value="26"/></td></tr><tr><td colspan="3"><input type="button" value="Add"/></td></tr></tbody></table> |  |  | whsID | stock | indeli | <input type="text" value="WHS_B"/> | <input type="text" value="-22"/> | <input type="text" value="67"/> | <input type="text" value="WHS_C"/> | <input type="text" value="-21"/> | <input type="text" value="26"/> | <input type="button" value="Add"/> |  |
| whsID                              | stock                            | indeli                                                                                                                                                                                                                                                                                                                                                                                                                                      |  |  |       |       |        |                                    |                                  |                                 |                                    |                                  |                                 |                                    |  |
| <input type="text" value="WHS_B"/> | <input type="text" value="-22"/> | <input type="text" value="67"/>                                                                                                                                                                                                                                                                                                                                                                                                             |  |  |       |       |        |                                    |                                  |                                 |                                    |                                  |                                 |                                    |  |
| <input type="text" value="WHS_C"/> | <input type="text" value="-21"/> | <input type="text" value="26"/>                                                                                                                                                                                                                                                                                                                                                                                                             |  |  |       |       |        |                                    |                                  |                                 |                                    |                                  |                                 |                                    |  |
| <input type="button" value="Add"/> |                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                             |  |  |       |       |        |                                    |                                  |                                 |                                    |                                  |                                 |                                    |  |

|                 |    |                                      |
|-----------------|----|--------------------------------------|
| warehouseRuleRC | RC | <input type="text" value="APPROVE"/> |
|-----------------|----|--------------------------------------|

Edit Source

Figure 6-38 Approving warehouse human task after having copied the input in the form

Figure 6-39 shows the Financial Office task.

Financial officer authorization

Form View

|          |            |                    |  |  |
|----------|------------|--------------------|--|--|
| customer | custID     | 10003              |  |  |
|          | custName   | YXZ itso Corp.     |  |  |
|          | address    | 4205 S. Miami Blvd |  |  |
|          | zipCode    | 27709              |  |  |
|          | city       | RTP                |  |  |
|          | state      | NC                 |  |  |
|          | budget     | 5000               |  |  |
|          | soldToDate | 10295              |  |  |

|       |                |                                         |  |  |
|-------|----------------|-----------------------------------------|--|--|
| order | ordID          | _PI:90030118.5cb755e4.4455d5f6.be6200a3 |  |  |
|       | amount         | 75                                      |  |  |
|       | submitterID    | user2                                   |  |  |
|       | state          | CREATED                                 |  |  |
|       | creationDate   | 2/28/08 2:06 AM                         |  |  |
|       | completionDate | 2/28/08 2:06 AM                         |  |  |
|       | itemQty        | 15                                      |  |  |

|      |            |         |       |            |                |
|------|------------|---------|-------|------------|----------------|
| item | itemID     | IT_003  |       |            |                |
|      | itemName   | Item003 |       |            |                |
|      | price      | 5       |       |            |                |
|      | warehouses | whsID   | stock | indelivery | itemQtyPartial |
|      |            | WHS_B   | -22   | 67         | 5              |
|      |            | WHS_C   | -21   | 26         | 5              |

View Source

Form View

|                 |    |         |
|-----------------|----|---------|
| financialRuleRC | RC | APPROVE |
|-----------------|----|---------|

Figure 6-39 Approving the financial human task

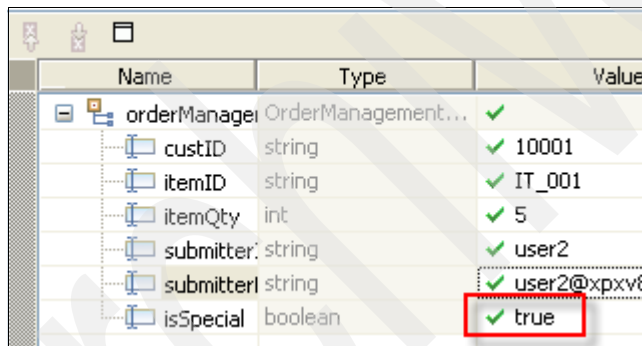
## Verifying the results

There are several methods that you can use to verify the results of the test:

- ▶ Log in as the submitter in Business Process Choreographer Explorer and look at the Process Instance status. If the process was successful, you see only the process if you are not having the instances deleted automatically.
- ▶ Browse the WebSphere Process Server logs. In a cluster, you have to look at the logs for each server in the cluster. The server logs are located in *profile\_root/logs/server\_name/SystemOut.log*.
- ▶ You can look at any external results that you might be expecting, such as files created, database record updates, or e-mail events.

## Testing OrderPreProcessingMediation

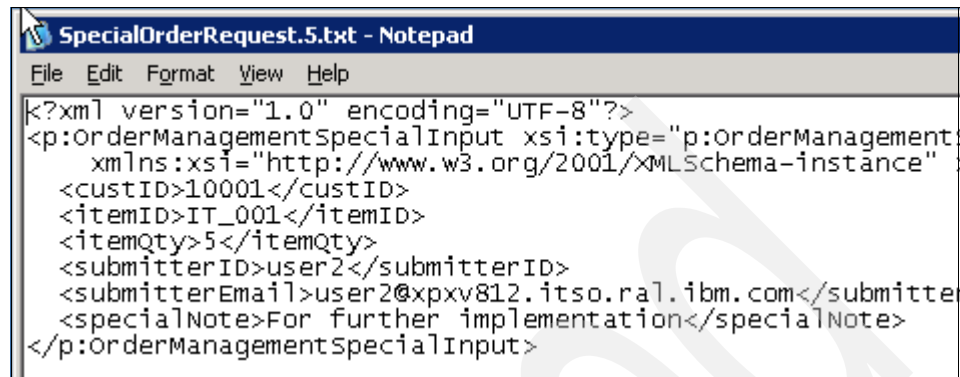
Finally, you test the OrderPreProcessingMediation through the OrderTestClient module. Select the appropriate component, interface, and method from the Integrated test client interface. We tested the special order management case as shown in Figure 6-40.



| Name         | Type               | Value         |
|--------------|--------------------|---------------|
| orderManager | OrderManagement... | ✓             |
| custID       | string             | ✓ 10001       |
| itemID       | string             | ✓ IT_001      |
| itemQty      | int                | ✓ 5           |
| submitter    | string             | ✓ user2       |
| submitterI   | string             | ✓ user2@xpxv6 |
| isSpecial    | boolean            | ✓ true        |

Figure 6-40 Testing the OrderPreProcessing in the special order case

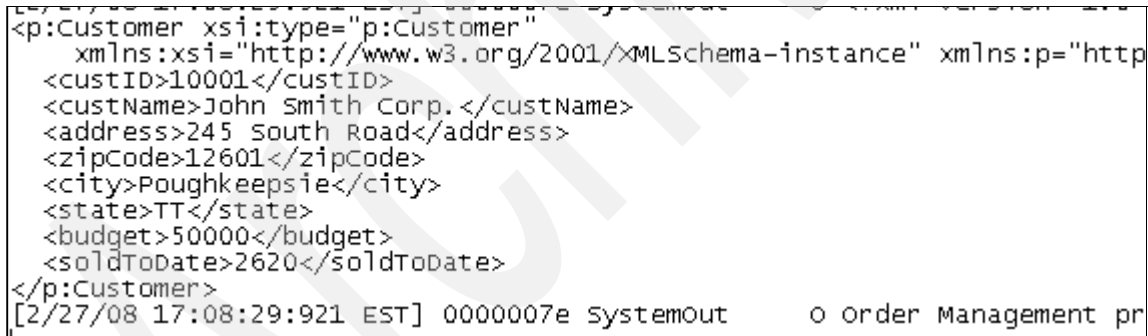
Make sure that the new file is created in the C:\ITSOSpecialOrders folder as shown in Figure 6-41.



```
<?xml version="1.0" encoding="UTF-8"?>
<p:OrderManagementSpecialInput xsi:type="p:OrderManagementSpecialInput"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <custID>10001</custID>
 <itemID>IT_001</itemID>
 <itemQty>5</itemQty>
 <submitterID>user2</submitterID>
 <submitterEmail>user2@xpxv812.itso.ral.ibm.com</submitterEmail>
 <specialNote>For further implementation</specialNote>
</p:OrderManagementSpecialInput>
```

Figure 6-41 Special Order file sent to the file system by the mediation

Lastly, test the OrderPreProcessingMediation by forwarding a regular request to the OrderManagement (isSpecial is false) and the values for automatic approval. This results in the invocation of OrderManagement. Again, verify by looking at the SystemOut logs for the server and by ensuring the e-mail was sent to the user as shown in Figure 6-42.



```
<p:Customer xsi:type="p:Customer"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:p="http://www.ibm.com/xmlns/prod/ibm.com/OrderManagement/2001/01/OrderManagementTypes">
 <custID>10001</custID>
 <custName>John Smith Corp.</custName>
 <address>245 South Road</address>
 <zipCode>12601</zipCode>
 <city>Poughkeepsie</city>
 <state>NY</state>
 <budget>50000</budget>
 <soldToDate>2620</soldToDate>
</p:Customer>
[2/27/08 17:08:29:921 EST] 00000007e SystemOut O Order Management pr
```

Figure 6-42 Looking at logs for successful completion of the OrderManagement

# Related publications

We consider the publications that we list in this section particularly suitable for a more detailed discussion of the topics that we cover in this book.

## IBM Redbooks publications

For information about ordering these publications, see “How to get IBM Redbooks publications” on page 240. Note that some of the documents that we reference here might be available in softcopy only.

- ▶ *Getting Started with IBM WebSphere Process Server and IBM WebSphere Enterprise Service Bus: Part 1: Development*, SG24-7608
- ▶ *Getting Started with IBM WebSphere Process Server and IBM WebSphere Enterprise Service Bus: Part 2: Scenario*, SG24-7642
- ▶ *Business Process Management: Modeling through Monitoring Using WebSphere V6.0.2 Products*, SG24-7148
- ▶ *WebSphere Application Server for i5/OS Handbook: Version 6.1*, SG24-7221
- ▶ *WebSphere Application Server V6: Diagnostic Data*, REDP-4085

## Online resources

The following Web sites are also relevant as further information sources:

- ▶ IBM WebSphere Application Server V6.1 information center  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp>
- ▶ IBM WebSphere Process Server V6.1 information center  
[http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/welcome\\_top\\_wps.htm](http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wps.610.doc/welcome_top_wps.htm)
- ▶ IBM WebSphere Enterprise Service Bus V6.1 information center  
<http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.websphere.wesb.61x.root.doc/info/welcome.html>

- ▶ IBM WebSphere Integration Developer V6.1 information center  
<http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.wbit.610.help.nav.doc/topics/welcome.html>
- ▶ IBM WebSphere Integration Developer V6.1 information center: Configuring and Using Adapters  
<http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.wbit.610.help.adapter.emd.ui.doc/topics/tcreatecmps.html>
- ▶ *WebSphere z/OS V6—WSC Sample ND Configuration*  
<http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100653>

## How to get IBM Redbooks publications

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this Web site:

[ibm.com/redbooks](http://ibm.com/redbooks)

## Help from IBM

IBM Support and downloads

[ibm.com/support](http://ibm.com/support)

IBM Global Services

[ibm.com/services](http://ibm.com/services)

# Index

## A

AboutThisProfile.txt 170  
addNode 143  
address space 8  
adjunct process 125  
administrative console 164  
    starting 168  
administrative domain 2  
administrative security 169–171  
apachectl start 73  
application deployment target 12  
application server 2  
application server profile 99, 170  
Application Server Toolkit 98, 100, 116  
augment 3–4, 25, 29, 34, 54, 56, 94–95, 97, 116,  
129, 134, 136, 140, 142–143, 145

## B

BBOCCINS 179  
BBOWMNAN 129  
BPEDB 25, 29  
BPEODB 25, 29  
Business Flow Manager 29, 158–160  
business integration module 2, 227  
Business Process Choreographer 12, 25, 29,  
64–65, 136, 158–161, 227  
Business Process Choreographer Explorer 229,  
231–232, 234, 237  
Business Process Manager 145  
Business Rules Manager 12

## C

CEI database 25  
cell 2, 168  
cell name 107, 122  
cell profile 99  
certificate 112–113, 126–127  
certificate authority 112, 126  
cluster 182, 237  
command line tools 164–165  
Common Event Infrastructure 136  
Common Event Interface (CEI) bus 12

control region 110, 125  
control region (CR) 8  
control region adjunct 9  
controller process 110, 125  
controller region 180  
CTRACE parmlib member 109, 123  
custom node 4, 6, 35–36, 40, 49, 72, 79, 82, 116,  
123–124, 127, 129–130, 138–140, 142–144  
custom profile 3, 6, 22, 57, 79, 116  
custom user registry 113, 127

## D

daemon 111, 125, 180  
daemon server 10  
default profile 18  
deployment environment pattern 6–7, 11, 22, 53,  
75, 79, 82  
Deployment manager 48  
deployment manager 2–3, 5–8, 22, 25, 28, 30–31,  
34, 36–37, 40, 49, 72, 74–75, 77–79, 82–83,  
99–101, 108–110, 116, 127, 129, 132–133, 135,  
138–140, 143, 145, 147, 165, 168, 178–180, 182,  
199, 216, 220  
    starting 164, 178  
    starting on z/OS 179  
    stopping 164, 179  
    stopping on z/OS 180  
deployment manager name 107, 122  
deployment manager profile 57, 99  
Derby Embedded 28  
Derby Network 28  
distributed server 5  
distributed server configuration 6  
distributed server environment 17  
DMGR 132

## E

EVENT database 25  
EventSequencing (LockManager) 27

## F

failover 5

federate 3–4, 6, 18, 31, 79, 82, 99, 116, 127, 129,  
138, 142–144  
    application server 99  
federated repositories 207  
file-based user registry 113, 126–127

## G

GetPluginCfg 184

## H

HFS 108–109, 123  
HLQ field 103  
human task 160, 207, 209, 233–236  
Human Task Manager 29, 145, 159–160

## I

IBM HTTP Server 71, 73  
IBM Web Administration 173  
IBM Web Administration for i5/OS 165, 171, 175  
ISPF Customization Dialog 98, 115, 129

## J

JMS binding 213  
Jython 113, 127

## L

LDAP 113, 127  
log  
    HTTP 17  
long name 107, 122

## M

managed node 17, 99  
ManagedDB2.rsp 138  
manageprofiles 4, 20–22, 24, 54, 56–57, 98  
manageprofiles script 98  
mediation module 2, 227  
Message Logger primitive 27  
MQ binding 213  
MQ JMS binding 214

## N

node 2  
    stopping 182  
node agent 2, 5, 124, 128, 143, 180–182  
    restarting 182

starting 180  
stopping 181  
stopping on z/OS 182

## P

plug-in configuration file  
    automated propagation 187  
    automatic regeneration 185  
    propagating 186  
    regenerating 184  
port 170  
ports 110  
print() 187  
println() 187  
printStackTrace() 187  
Process Instance status 237  
PROCLIB 105  
profile 3  
profile directory 165  
Profile Management Tool 4, 22, 24, 32, 34, 54–56,  
58, 72, 82, 84, 94, 98, 100  
Profile Management Tool Client 55  
profile\_root 166–167

## Q

QHTTPSVR 173  
QShell 167

## R

RACF 112, 114, 126  
Redbooks Web site 240  
    Contact us xii  
Relationship service 27  
remote messaging and remote support pattern 11,  
14–16, 71, 79, 81–82  
remote messaging pattern 11, 13–14, 70, 79, 82  
response file 101  
responsefile.wbis.txt 49

## S

SAF 112–113, 126  
SAF EJBROLE 112, 114, 126  
SAF keyring name 112  
SBBOEXEC 106  
SBBOLD2 106  
SBBOLoad 105  
SBBOLPA 105

- SBBOMSG 106
- SCA application bus 12, 153, 214
- SCA system bus 12, 153
- scripting client 175
- scripting program 175
- security domain identifier 113
- servant process 110
- servant region 180
- servant region (SR) 9
- serverStatus 177
- service integration bus 9, 13, 146, 213
- short name 107, 122
- single cluster pattern 11–12, 69–71, 79, 82, 194
- single cluster topology 189, 194
- SMS 108
- SOAP/HTTP 227
- SOAP/JMS 227
- SSL 111–112
- stand-alone application server 4
- stand-alone server 3–5, 10, 22, 40, 48, 58, 68, 145
- stand-alone server profile 57
- stand-alone service bus profile 3
- startManager 178
- startNode 178, 180
- startServer 176–177
- STEPLIB 107
- stopManager 179
- stopServer 177
- STRQSH 177
- STRTCPSVR 172

## U

- unmanaged node 16, 90
- Update Installer 24, 50–51, 72, 77, 81

## W

- WAS\_USER\_SCRIPT 18
- Web server
  - adding 91
- Web Server plug-in 193
- Web server plug-in 15, 63, 72–73, 185–187, 193, 214, 221–222, 226
  - regenerating 185
- Web server plug-in configuration service 185
- Web service URL 218, 222
- WebSphere Application Server - Express 4
- WebSphere Application Server for z/OS Profile Management Tool (zPMT) 98

- WebSphere Application Server Network Deployment 5
- WebSphere Process Server Network Deployment
  - Creating a cluster 146
  - DB2 libraries 145
  - ManagedDB2.rsp 140
- Windows service 164
- WLM 108
- WLM APPLENV 108
- workload management 5
- WPRCSDB 25, 27, 29, 69, 76, 78, 153
- WPS\_install\_root 166
- wsadmin 164–165, 175

## X

- X509 112, 126

## Z

- z/OS workload manager 9
- zFS 108–109, 123
- zPMT 21, 25, 54, 98–100, 115–116, 128
- zSMPInstall 21, 132–133, 138–139
- zWESBConfig 4, 21
- zWLM 9
- zWPSCConfig 4, 21, 132–133, 135–136, 138, 142–143











**Redbooks®**

# Getting Started with IBM WebSphere Process Server and IBM WebSphere Enterprise Service Bus

## Part 3: Run time

### Install WebSphere Process Server

### Build topologies based on patterns

### Prepare for deployment

This IBM Redbooks publication provides new users with information about how to install and configure IBM WebSphere Process Server and IBM WebSphere Enterprise Service Bus runtime environment. It includes installation examples on distributed, i5/OS, and z/OS platforms. It also provides basic configuration information for deploying applications.

Some knowledge of IBM WebSphere Application Server Network Deployment is assumed for the readers of this book.

This book is the third book of a three-part series:

*Getting Started with IBM WebSphere Process Server and IBM WebSphere Enterprise Service Bus:*

- ▶ *Part 1: Development, SG24-7608*
- ▶ *Part 2: Scenario, SG24-7642*
- ▶ *Part 3: Run time, SG24-7643*

### INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

### BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:**  
[ibm.com/redbooks](http://ibm.com/redbooks)