



Planning the Installation



Planning the Installation

Note

Before using this information, be sure to read the general information in the Notices section at the end of this document.

31 March 2008

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

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

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

© Copyright International Business Machines Corporation 2007, 2008. All rights reserved.

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

Contents

Chapter 1. Planning for WebSphere Process Server. 1

Chapter 2. Determining your software needs. 3

Assessing your business requirements	3
Identifying available resources	4
Development and deployment version levels	5
Planning for interoperability between WebSphere Process Server and other WebSphere Application Server products	6
Determining the products to install	6
Choosing a database.	8
Identifying necessary security authorizations	10
Considerations for Service Component Architecture support in servers and clusters	11

Chapter 3. Using multiple platforms within a cell 13

Chapter 4. Planning your deployment environment 15

Planning scenarios	17
Planning to install WebSphere Process Server while installing WebSphere Integration Developer	17
Planning to install WebSphere Process Server for use by WebSphere Integration Developer	18
Planning for a default stand-alone environment	19
Planning for a custom stand-alone environment	20
Planning for a deployment environment based on one of the supplied patterns.	22
Planning for a custom deployment environment	25
Profiles	28
Servers	29
Stand-alone server	29
Network deployment	30
Deployment manager	31
Managed nodes	31
Deployment environments	32
Choosing your deployment environment pattern	39

Chapter 5. Implementing a deployment environment 41

Notices 45

Chapter 1. Planning for WebSphere Process Server

Before introducing new software into your enterprise information system you must assess your current environment and your business requirements to ensure that the system you implement meets your needs. Middleware, such as WebSphere Process Server, requires that you evaluate many aspects of your enterprise information system (EIS), such as capacity and security. This section describes how to plan for WebSphere Process Server.

Note: The information presented here is available in Adobe® PDF format at the following link: [WebSphere® Process Server documentation \(in PDF format\)](#).

The answers to the following questions can help you design a deployment environment to meet your needs:

- What are your business objectives and how can software help you achieve those objectives?
- What applications must you integrate?
- Do you want to eliminate duplicate information?
- What are the requirements for system response time and availability?
- What financial, hardware, software, and human resources are available for you to complete the installation?
- Do you require the services of other departments?
- What tasks must be performed? Who will perform them?
- What existing hardware do you need for the installation?
- Do you need additional hardware to achieve the business requirements?
- Can you use existing databases or do you require new databases?
- Can existing user IDs be used by WebSphere Process Server components, or do you require new IDs? What authorization do new IDs require?
- Are there financial considerations that limit the number of product licenses you can purchase?
- How is your system going to evolve? For example, will it need to handle increased load, or handle more concurrent users in the future? Will you need to add additional resources in the future to meet additional demands?
- Will your system need to dynamically add or remove resources to handle daily fluctuations in demand?
- Does your system need to support fluctuations in load or number of concurrent users, on a periodic basis?

Also, think about your current goals: are you planning a test or production environment? Is it small-scale or large-scale? Do you want to set up quickly, with default values, or do you want to customize your environment? At the end of this section, there are planning suggestions for several different scenarios, depending on what you are trying to achieve.

Chapter 2. Determining your software needs

To minimize rework and outages, take the time to study your current environment before moving forward and making purchasing and installation decisions. Your current business needs and design, the hardware and software already installed, and an analysis of current strengths and shortcomings can aid you in determining the optimum design for your deployment environment. This planning could also help you minimize the financial investment needed for your current needs.

The information in this section guides you in analyzing your current and future needs to develop an environment to meet those needs.

Assessing your business requirements

Your current business requirements provide the baseline on which to formulate a plan for streamlining integration of your business components and thus improving functionality. Having a vision for the future of your business provides a guideline that can assist in making decisions that will work not only today but also as your business grows.

Before you begin

You need to know how your product or service is created and delivered.

About this task

As part of the planning process, you need to analyze how your business works. These steps provide a framework for this analysis.

Procedure

1. Plot the course of your product or service from beginning to end.

The process might be strictly linear, or might contain loops, detours and work-arounds. Draw a rough diagram and indicate connections and interactions. For each section of your product life-cycle, analyze the procedures used to move the product forward.

 - Does the procedure use paper forms and notations, is it computerized or both?
 - If it is computerized, what software is used? What hardware?
 - Are there any logjams in the process? Is there any confusion? For example, handwriting may be difficult and time-consuming to decipher; staff may have difficulty in mastering required computer skills.
 - What areas in this process are running smoothly? What are the strengths?
2. Determine how the sections of the diagram you created in step 1 perform their tasks.
 - Do any of the sections use the same software? Hardware? Forms?
 - If sections use different software, how well do the applications communicate with each other, if at all?
 - Does each section interact only with the sections immediately preceding and following it, or does it also detour to a section in a different stage of the cycle? If so, why? Does this cause confusion or delay?

- If there is an existing intranet that sections use for communication, do some sections bypass this intranet? If so, why? Does the intranet have a history of delays or downtime that impacts other processes?
 - What areas of interaction are running smoothly? In which areas are the bottlenecks? How serious are they?
3. Consider processes that interact with outside sources.
 - What comments do you get from customers, both positive and negative? Are there any patterns to complaints? Are particular areas consistently making customers happy?
 - How do other business entities interact with your company? With which departments do they communicate? How does this communication occur – in writing, or web-based? Add these sources to your diagram. Note areas that work smoothly, and areas that might be causing delays or errors.
 4. Plan for the future.
 - Where would you like to see your business in one year? In five years? Ten years?
 - Will you be opening new outlets? Increasing advertising? Increasing your customer base?
 - Is there a possibility of acquiring competitors and incorporating their products and services into your own? Is there a possibility of venturing into new avenues of products or services?

What to do next

Identify your available resources.

Identifying available resources

Identify your assets to make the best use of those resources already available and also to make informed purchasing decisions.

Before you begin

You must be familiar with your current hardware and software. Prepare a list of the available assets.

About this task

You are assessing your current enterprise information system to determine if you require any additional hardware or software to meet your business needs.

Procedure

1. Itemize each piece of physical hardware. Note:
 - Amount of installed memory
 - Number and type of installed microprocessors
 - External media
 - Whether a particular unit can be upgraded
2. Itemize the currently installed software and database applications. Note:
 - Function
 - Breadth of use across the company
 - Security requirements

3. List your current IT personnel and note whether you have available the means for installation and maintenance of WebSphere Process Server, as well as the required expertise in database management. Make sure all involved have user IDs with the appropriate authorizations to successfully install all products and files.

Related concepts



Hardware and software requirements

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

Development and deployment version levels

Your decision about what version levels of WebSphere Process Server you need in your environment will depend on the version levels with which your applications were developed. Generally applications deployed in a previous version of WebSphere Process Server will run on the next available version of WebSphere Process Server.

WebSphere Process Server version 6.1 and WebSphere Integration Developer version 6.1 are compatible with prior releases as follows:

- Deployment from WebSphere Integration Developer version 6.0.x.x (where 6.0.x.x refers to either 6.0.1.x or 6.0.2.x) to WebSphere Process Server 6.1 is supported.
 - Applications authored and generated using WebSphere Integration Developer 6.0.x.x can be published to WebSphere Process Server 6.1 servers.
 - Applications authored in, generated in, and exported from WebSphere Integration Developer 6.0.x.x can be installed on WebSphere Process Server 6.1 servers.

Note: For version 6.0.1 WebSphere Adapters, some additional steps might be required for compatibility. Please check the product technotes at WebSphere Process Server technotes for more information.

- Running WebSphere Process Server 6.1 artifacts on WebSphere Process Server 6.0.x.x is *not* supported.
 - Applications authored with WebSphere Integration Developer 6.1 cannot be published to or installed on WebSphere Process Server 6.0.x.x (any prior release) servers. Such content will not run correctly on WebSphere Process Server 6.0.x.x, and changes in code generation will cause the applications to not run correctly on WebSphere Process Server 6.0.x.x.
 - Applications authored with WebSphere Integration Developer 6.0.x.x and generated in WebSphere Integration Developer 6.1 cannot be published to or installed on WebSphere Process Server 6.0.x.x servers. Changes in code generation will cause the applications to not run correctly on WebSphere Process Server 6.0.x.x.
 - Applications generated using serviceDeploy from a WebSphere Process Server 6.1 server cannot be installed on a WebSphere Process Server 6.0.x.x server. Changes in code generation will cause the applications to not run correctly on WebSphere Process Server 6.0.x.x.

Related concepts

“Planning for interoperability between WebSphere Process Server and other WebSphere Application Server products” on page 6

When analyzing your software environment, you need to know whether requests can pass between the various software levels that exist in your deployment environment.

Related information

 [Migrating to WebSphere Process Server](#)

Migrating refers to the process of moving from one product to another or one version of a product to another while preserving product configuration information and user applications, thus enabling the existing applications and configuration data to be used in the new environment. You can migrate to WebSphere Process Server from certain other IBM® products or from an earlier version of WebSphere Process Server to a later version such as version 6.1.

Planning for interoperability between WebSphere Process Server and other WebSphere Application Server products

When analyzing your software environment, you need to know whether requests can pass between the various software levels that exist in your deployment environment.

To maintain optimum interoperability, after applying any related WebSphere Application Server service and following any applicable guidelines, make sure that you apply any outstanding fixes for WebSphere Process Server.

Related concepts

“Development and deployment version levels” on page 5

Your decision about what version levels of WebSphere Process Server you need in your environment will depend on the version levels with which your applications were developed. Generally applications deployed in a previous version of WebSphere Process Server will run on the next available version of WebSphere Process Server.

Related information

 [Interoperating \(WebSphere Application Server\)](#)

Determining the products to install

The design of your deployment environment includes determining how many and which type of software products you may require. Based on your needs the product requirements may vary among the computer LPAR systems involved in the environment. Not every server in a deployment environment requires a WebSphere Process Server.

Before you begin

You should have a detailed design that lists:

- The clusters and servers involved in the deployment environment
- The physical hardware on which the various servers are located
- The functionality each cluster provides for the deployment environment. For example web application component support, Java™ Platform, Enterprise Edition component support, mediation module support, messaging support, or process server support.

About this task

After designing your deployment environment and before purchasing your software, use this task to determine the appropriate software needed for your deployment environment.

Procedure

1. Count the number of unique LPARs that will host the various components of the deployment environment.

Important: When you are running multiple servers on the same LPAR system, you must install the software that provides the most functionality you need for any server running on that LPAR system.

This count includes:

- The number of deployment managers you need. The software running on the managed servers determines what software you install on the deployment manager.
 - WebSphere Process Server instances
 - WebSphere ESB instances: unique instances that will host mediations only
 - Messaging engines that are not already counted: representing the number of unique WebSphere Application Server instances
2. Determine whether the cost of the software exceeds the budget for the project.
 3. Optional: Adjust your design to meet your financial requirements. You will need to host multiple servers on the largest capacity LPARs to reduce costs.
 - Create multiple server instances of the same type on higher-capacity LPARs rather than separate instances on separate LPARs, so the number of instances remains the same but the software needed decreases.
 - Decide whether you need unique LPARs hosting messaging engines. If not, eliminate them.
 - Remove applications from the design to decrease the number of application server instances required.

Results

You now know the software required to implement the design.

What to do next

Order the software you need.

Related concepts

“Deployment environments” on page 32

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

“Clusters” on page 33

Clusters give your applications more capacity and higher availability than a single server.

“Servers” on page 29

Servers provide the core functionality of WebSphere Process Server. Process servers extend, or augment, the ability of an application server to handle Service Component Architecture (SCA) modules. Other servers (deployment managers and node agents) are used for managing process servers.

Messaging engines

A messaging or queue destination host provides the messaging function within a server. A server becomes the messaging destination host when you configure it as the messaging target.

Choosing a database

Choosing a database depends on your operating system and on the features that you will use. During the installation procedure, the wizards prompt you to select your databases. Under some circumstances you can operate with only one database containing several tables.

Before you begin

To plan your database configuration, you must know the components that you will use.

About this task

After performing this task you will know the number of databases to configure.

The following table shows a mapping between the various WebSphere Process Server components and the corresponding databases which contain the tables related to these components.

Procedure

Select the components your installation will use to determine which database tables you need. Table 1 lists the components and their corresponding required database tables.

Table 1. Databases required by individual components

Component	Database (name specified is the default and can be changed for your use)
AppScheduler	Common database (WPRCSDB)
Business Process Choreographer	Business Process Execution database (BPEDB)
Business Process Choreographer Observer	Business Process Execution Observer database (BPEODB) Important: To prevent performance degradation, make sure that Business Process Choreographer Observer has its own database and not just tables within another database.
Common Event Infrastructure (CEI)	CEI database (EVENT) Important: To prevent performance degradation, make sure that CEI has its own database and not just tables within another database.
Enterprise Service Bus	Common database (WPRCSDB)
EventSequencing (LockManager)	Common database (WPRCSDB)
Mediation	Common database (WPRCSDB)
Recovery	Common database (WPRCSDB)
Relationships	Common database (WPRCSDB)
Selectors/Business Rules	Common database (WPRCSDB)

Table 1. Databases required by individual components (continued)

Component	Database (name specified is the default and can be changed for your use)
Service Integration Bus	SIBDB (created during messaging engine configuration)

Related concepts

Database specifications

WebSphere Process Server uses a number of database tables to hold, store and track information. Some of the components that comprise WebSphere Process Server use their own database tables.

Common database specifications

The Common database specifications contain information about supported database types; scripts and their locations; installation parameters; types of created tables and user ID privileges.

Common Event Infrastructure database specifications

The Common Event Infrastructure database specifications list the types of supported databases, script locations, profile configuration types, and necessary User ID privileges.

Business Process Choreographer database specifications

The Business Process Choreographer database specifications list supported database types, script locations, profile creation types, database restrictions and necessary User ID privileges.

Messaging engine database specifications

The messaging engine database specifications lists supported database type; scripts and their locations; profile creation types; and necessary user ID privileges.

Enterprise service bus logger mediation database specifications

Use the enterprise service bus logger mediation database specifications for locating information about supported database types; script names and their locations; profile creation configuration actions; schema upgrades and user ID privileges.

Selector and business rules group database specifications

Use the Selector and business rules group database specifications to find information about supported database types; scripts and their locations; profile creation configuration actions; restrictions; table names; and user ID privileges.

JDBC providers

JDBC providers allow applications to interact with relational databases.

Data sources

Data sources provide the link between applications and relational databases.

Scripts for configuring DB2 on a remote z/OS server

If you plan to use DB2® on a remote z/OS® machine for the Common Event Infrastructure and Common database repositories, you or the database administrator (DBA) must create relevant databases and correct storage groups on the z/OS workstation.

Identifying necessary security authorizations

Depending on your site's security policy, to successfully implement a design, you might need user IDs and passwords to allow various tasks to complete, such as creating files and folders, and for database access. Identifying necessary authorities prevents problems when the servers attempt to access protected data.

Before you begin

- Complete your design.
- Determine the authentication system to use, for example Lightweight Directory Access Protocol (LDAP).
- Review the security policies for your site to determine what controls are in place that affect the authorizations required for your WebSphere Process Server installation.
- Identify the systems on which you are installing the product.

About this task

The security policies for your site enables global security which indicates that you require specific authorities to install software, create databases or tables, or access databases. To successfully install and operate the product you must do these steps.

- Acquire, or supply to your security administrator, user IDs and passwords that have enough authority to install software on the systems.
You must run the installation wizards for WebSphere Process Server using IDs that have the authority to create files and folders.
- Acquire, or supply to your security administrator, user IDs, passwords, and roles needed for daily operations of the system. These include:
 - Administrative console user IDs and roles to limit capabilities. You can have user IDs for configuring, administrating, or monitoring roles.
 - User IDs for each system bus to use to authenticate system communications.
 - Administrative and monitoring user IDs or groups for each Business Process Choreographer container for authentication with business flow manager and human task manager.
 - User IDs or groups for synchronous calls to authenticate with business flow manager and human task manager.
- Optional: Acquire, or supply to your database administrator, user IDs and passwords that WebSphere will use to create databases or database tables during installation.

Note: Your site policies might restrict this authority to the database administrator. In this case, you will provide generated scripts to the administrator to create the databases or database tables.

- Acquire, or supply to your database administrator, user IDs and passwords that WebSphere will use to access the database tables it uses during operation.

Results

You can install and operate your WebSphere servers in a secure environment.

Related information

 Planning security, user IDs, and authorizations

Considerations for Service Component Architecture support in servers and clusters

Servers and clusters can support Service Component Architecture (SCA) applications, application destinations, or both.

SCA applications (also called service applications) require the use of one or more of the automatically created service integration buses. Each application uses a set of messaging resources, which are called *destinations*. These destinations require configured messaging engines, and they can be hosted on the same server or cluster as the application or on a remote server or cluster. Messaging engines use database data stores.

By default, new servers and clusters in a network deployment configuration are not configured to host SCA applications and their destinations.

Note: A stand-alone server has SCA support automatically configured. You cannot disable this configuration.

To enable this support, use the Service Component Architecture page in the administrative console. For servers, ensure that the application class-loader policy is set to `Multiple`.

Before enabling SCA support for a server or cluster in a network deployment or managed node environment, determine which of the following possible configurations you want to implement:

- **Remote bus member configuration:** The server or cluster hosts SCA applications, but the destinations are hosted on a remote server or cluster. This scenario requires the remote service integration bus members to be configured with the messaging engines needed to host the destination.

While the use of remote messaging requires initial investment in planning for and configuring the service integration bus and its members, that configuration can be reused by multiple members within the application cluster. Messages are distributed to every member. In addition, the initial configuration can be structured to provide failover support.

- **Local bus member configuration:** The server or cluster hosts both SCA applications and application destinations. The required messaging engines are configured using the local bus members on the server or cluster.

Refer to the planning topics to help you decide which configuration is appropriate for your environment.

Related concepts

“Clusters” on page 33

Clusters give your applications more capacity and higher availability than a single server.

 Managed servers

A managed server is a server that is configured in a managed node. It provides a resource within the deployment environment that runs your applications.

 Managing resources for mediation modules

Mediation modules uses resources provided by the service integration technologies of WebSphere Application Server. Mediation modules can also make use of a range of resources, including those provided by the Java Message Service (JMS) and common event infrastructure. To administer the resources for mediation modules, you can use the WebSphere administrative console, commands, and scripting tools.

Related tasks

 [Planning your deployment environment](#)

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

Related information

 [Configuring class loaders of a server](#)

 [Learning about service integration buses](#)

 [Messaging engines](#)

 [Learning about service integration buses](#)

 [Messaging engines](#)

 [Configuring class loaders of a server](#)

Chapter 3. Using multiple platforms within a cell

With careful planning, you can create a deployment manager cell that includes nodes on both distributed, i5/OS[®] and z/OS operating system platforms.

For example, you can create a deployment manager cell that includes i5/OS nodes, z/OS nodes, Linux[®] nodes, UNIX[®] nodes, and Windows[®] nodes. This kind of configuration is referred to as a *heterogeneous* cell.

A heterogeneous cell does require significant planning. Setting up a heterogeneous cell can also take more time as some of the tasks can not be automated. The “Heterogeneous Cells – cells with nodes on mixed operating system platforms” white paper outlines the planning and system considerations required to build a heterogeneous cell.

If you use the administrative console to create a new server, you choose the *server template* that provides the initial configuration settings for the server. After you select a managed node on which to create a server, the administrative console offers you the choice of templates that can be used for the operating system platform of that node.

Important: While cells can be heterogeneous, you cannot mix z/OS nodes with other nodes in a server cluster.

Related concepts


“Deployment manager” on page 31

A deployment manager is a server that manages operations for a logical group, or cell, of other servers. The deployment manager is the central location for administering the servers and clusters.

“Managed nodes” on page 31

A managed node is a node that has been federated into a deployment manager cell. In a managed node, you can configure and run managed servers.

Related information

 Heterogeneous Cells – cells with nodes on mixed operating system platforms

Chapter 4. Planning your deployment environment

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

Before you begin

Make sure you completed these tasks.

- Identified available resources
- Chose a database type
- Identified necessary authorities

About this task

Planning the layout of interconnected servers, requires that you make some decisions. These decisions will influence trade-offs you will make between the available hardware and physical connections, the complexity of the management and configuration and requirements such as performance, availability, scalability, isolation, security and stability.

Procedure

1. Determine the purpose for the deployment environment.
2. Identify the functional requirements of the deployment environment
 - a. Identify the component types you will deploy.
Consider the component types and the interactions between components as part of the requirements.
 - b. Identify the import and export implementation types and transports.
Consider the resources needed for the databases or Java Message Service (JMS) resources and the need for business events and their transmission mechanism.
 - c. Identify any functional requirements not related to applications.
Consider security servers, routers and any other hardware or software requirements to handle business events.
3. Identify the capacity and performance requirements for your environment.
4. Decide on the number of physical servers you need for each function.
5. Identify the redundancy requirements for your environment.
 - a. Identify the number of servers you need for failover.
 - b. Identify the number of routers you need.
Your choice of router will be influenced by exports of deployed modules, the types of queues you define on the service integration bus, Service Component Architecture (SCA) exports, and the type of load balancing you want among your clusters. IBM provides an embedded router used for Web Services exports with Service Object Access Protocol(SOAP)/JMS transports or JMS exports. However, if you choose not to use this embedded router provided by IBM, you will need to determine how to balance the load among your clusters based on the technology you are using.
6. Design your deployment environment.

Decide on the pattern. There are three established cluster patterns to choose from. If none of these three patterns meets your needs, you can create your own custom deployment environment.

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

See “Deployment environment patterns” for more information about the patterns and the differences amongst them.

7. Identify how you plan to install your deployment environment.

The single, remote messaging, and remote messaging and remote support clusters can be installed with a wizard through the administration console. You can install the custom deployment environment through a wizard in the administrative console or by building it yourself through the administrative console. You have the option of using the command line or a silent install for all or some of the installations.

What to do next

Select and follow the planning scenario that best fits your situation.

Related concepts

“Stand-alone server” on page 29

A stand-alone server provides an environment for deploying Service Component Architecture (SCA) modules in one server process. This server process includes, but is not limited to, an administrative console, a deployment target, the messaging support, the business rules manager, and a Common Event Infrastructure server.

“Deployment environments” on page 32

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

“Deployment environment patterns” on page 34

A deployment environment pattern specifies the constraints and requirements of the components and resources involved in a deployment environment. The patterns are designed to meet the needs of most business requirements and are intended to help you create a deployment environment in the most straightforward way.

“Clusters” on page 33

Clusters give your applications more capacity and higher availability than a single server.

Service integration buses

A service integration bus is a managed communication mechanism that supports service integration through synchronous and asynchronous messaging. A bus consists of interconnecting messaging engines that manage bus resources. It is one of the WebSphere Application Server technologies on which WebSphere Process Server is based.

Service components

All integration artifacts running on IBM WebSphere Process Server (for example, business processes, business rules, and human tasks) are represented as components with well defined interfaces.

Planning scenarios

How you plan your deployment environment depends upon how you plan to use your deployment environment. Read through the following scenarios and find the scenario that best matches how you plan to use your deployment environment.

Planning to install WebSphere Process Server while installing WebSphere Integration Developer

Use this scenario when application developers will access your deployment environment using WebSphere Integration Developer and a default configuration will meet your requirements.

Before you begin

Familiarize yourself with the installation processes described in the WebSphere Integration Developer information center. The requirements stated there are additional to requirements for WebSphere Process Server.

About this task

Before installing WebSphere Integration Developer, consider whether it is advantageous to install WebSphere Process Server to provide a server for developers to use to test applications. Moving your development team into an environment that provides testing capabilities at the outset can make your team productive quickly.

If a small test server can meet your requirements, consider installing WebSphere Process Server with WebSphere Integration Developer.

Procedure

1. Design the development and testing environments.
 - a. Determine the requirements for WebSphere Integration Developer.
 - b. Determine the requirements for the test server.

Talk with the development team to obtain their input on availability, capacity and security. In most cases, a single server isolated from the production environment will be sufficient for their use.
 - c. Make sure that the target servers have sufficient hardware to meet the needs.
2. Contact the security administrators to acquire whatever user IDs and accesses you require to complete installation.
3. Optional: Contact your database administrators, if your site policies restrict database creation and access to a centralized department.
4. Schedule and coordinate the installation of WebSphere Integration Developer and WebSphere Process Server to minimize impacts to the development community.

Install the hardware and WebSphere Integration Developer, selecting the option to install the test server on the servers you identified in step 1 and verify that the environment performs as expected.

Related concepts

“Stand-alone server” on page 29


A stand-alone server provides an environment for deploying Service Component Architecture (SCA) modules in one server process. This server

process includes, but is not limited to, an administrative console, a deployment target, the messaging support, the business rules manager, and a Common Event Infrastructure server.

“Servers” on page 29

Servers provide the core functionality of WebSphere Process Server. Process servers extend, or augment, the ability of an application server to handle Service Component Architecture (SCA) modules. Other servers (deployment managers and node agents) are used for managing process servers.

Related information

 [Configuring Business Process Choreographer](#)

Planning to install WebSphere Process Server for use by WebSphere Integration Developer

Use this scenario when applications developers will access your deployment environment using WebSphere Integration Developer and a default configuration does not satisfy your requirements.

Before you begin

Familiarize yourself with the installation processes described in the WebSphere Integration Developer information center. The requirements stated there are additional to requirements for WebSphere Process Server.

About this task

Use this procedure when you have an existing server that you think meets your development team’s need for a test server.

Examples of when you would use this scenario includes:

- Using a remote database such as DB2.
- Using a specific security repository.
- Testing on multiple environments. For example testing an application for both a previous version and a current version of the product.

Procedure

1. Determine your development team’s needs.
2. Design your development environment.
3. Design your testing environment. Use a server that is isolated from your production application environment. Isolating the test environment prevents contamination in your business data.

Location	Considerations
Development and testing servers are the same	<ul style="list-style-type: none">• Make sure that the server has the capacity to handle both workloads.• Make sure all developers can access the server.• Consider installing WebSphere Process Server at the same time you install WebSphere Integration Developer.

Location	Considerations
Development and testing servers are different	<ul style="list-style-type: none"> • Make sure that both servers can communicate. • Make sure all developers can access the server.

4. Contact the security administrators to acquire whatever user IDs and accesses you require to complete installation.
5. Optional: Contact your database administrators, if your site policies restrict database creation and access to a centralized department.
6. Schedule and coordinate the installation of WebSphere Integration Developer and WebSphere Process Server to minimize impacts to the development community.
7. Install WebSphere Process Server on the testing server you selected.
8. Install WebSphere Integration Developer on the development server you selected.

What to do next

Configure WebSphere Integration Developer to use the server you isolated.

Related concepts

“Stand-alone server” on page 29

A stand-alone server provides an environment for deploying Service Component Architecture (SCA) modules in one server process. This server process includes, but is not limited to, an administrative console, a deployment target, the messaging support, the business rules manager, and a Common Event Infrastructure server.

“Servers” on page 29

Servers provide the core functionality of WebSphere Process Server. Process servers extend, or augment, the ability of an application server to handle Service Component Architecture (SCA) modules. Other servers (deployment managers and node agents) are used for managing process servers.

Related tasks



Verifying the application deployment target cluster starts

To verify that the application deployment target cluster can start, you must start all three clusters in your deployment environment. This is an example for a three cluster deployment environment.

Related information



Configuring Business Process Choreographer

Planning for a default stand-alone environment

Use this scenario when your deployment environment must be isolated from other environments. Any applications running in this environment must be self-contained and use limited import protocols, such as Web services SOAP/HTTP. Also use this scenario when ease of installation and set up outweighs any requirement for high availability.

Before you begin

- Design your deployment environment.

- Make sure that you can satisfy all of your business requirements with a single server.
- Familiarize yourself with the concept of a stand-alone profile.

About this task

You have a design that requires you install a default single server environment to meet your needs.

Procedure

1. Determine the hardware and software you need to support your design.
2. Identify or create any user IDs with the authorization you need to complete installation.
3. Optional: Contact your database administrators, if your site policies restrict database creation and access to a centralized department.

Important: If your future plans include federating this environment into a deployment manager cell, make sure that you use a database and database drivers that support remote access. Examples of these types of products are Derby Network and Java toolbox JDBC.

4. Schedule and coordinate the installation of WebSphere Integration Developer and WebSphere Process Server to minimize impacts to the development community.

What to do next

Install the software.

Related concepts

“Stand-alone server” on page 29

A stand-alone server provides an environment for deploying Service Component Architecture (SCA) modules in one server process. This server process includes, but is not limited to, an administrative console, a deployment target, the messaging support, the business rules manager, and a Common Event Infrastructure server.

“Servers” on page 29

Servers provide the core functionality of WebSphere Process Server. Process servers extend, or augment, the ability of an application server to handle Service Component Architecture (SCA) modules. Other servers (deployment managers and node agents) are used for managing process servers.

Related information

 [Configuring Business Process Choreographer](#)

Planning for a custom stand-alone environment

Use this scenario when you need an isolated environment but cannot use a default single server environment because of your business requirements.

Before you begin

- Design your deployment environment.
- Make sure that you can satisfy all of your business requirements with a single server.
- Familiarize yourself with the concept of a stand-alone profile.

About this task

You have a design that requires you install a default single server environment to meet your needs.

Procedure

1. Select the database product to support the deployment environment.

Some systems, such as z/OS and i5/OS do not have automated methods to create databases and tables for messaging engines and Common Event Infrastructure (CEI). When creating databases for these systems, make sure you have sufficient authorization to execute database definition scripts successfully.

Important: If your future plans include federating this environment into a deployment manager cell, make sure that you use a database and database drivers that support remote access. Examples of these types of products are Derby Network and Java toolbox JDBC.

2. Decide how you create the database tables.

Either create the tables during product installation, have the product install process create scripts to create the tables for you, or create scripts yourself to perform this step.

3. Decide how clients are to access the applications in the deployment environment.

Based on your needs there are many ways to access including Web services (SOAP/HTTP and SOAP/JMS), synchronous or asynchronous Service Component Architecture (SCA) requests, Java Message Service (JMS), MQ (either JMS or native), or through adapters. These choices impact what other software and resources you must install.

4. Decide how the applications access any resources they require.

Based on your needs there are many ways to access including Web services (SOAP/HTTP and SOAP/JMS), synchronous or asynchronous Service Component Architecture (SCA) requests, Java Message Service (JMS), MQ (either JMS or native), or through adapters. These choices impact what other software and resources you must install.

5. Decide how to install the software, create and configure the server.

You can create and configure the server while installing the software or you can create and configure the server using the Profile Management Tool. You can also use the administrative console to create and configure the server. Experienced installers can also use scripts to handle these tasks. Understand the benefits and drawbacks of all methods before making a choice.

6. Identify or create any user IDs with the authorization you need to complete installation.
7. Optional: Contact your database administrators, if your site policies restrict database creation and access to a centralized department.

Important: If your future plans include federating this environment into a deployment manager cell, make sure that you use a database and database drivers that support remote access. Examples of these types of products are Derby Network and Java toolbox JDBC.

8. Schedule and coordinate the installation of WebSphere Integration Developer and WebSphere Process Server to minimize impacts to the development community.

What to do next

Install the software.

Related concepts

“Stand-alone server” on page 29

A stand-alone server provides an environment for deploying Service Component Architecture (SCA) modules in one server process. This server process includes, but is not limited to, an administrative console, a deployment target, the messaging support, the business rules manager, and a Common Event Infrastructure server.

“Servers” on page 29

Servers provide the core functionality of WebSphere Process Server. Process servers extend, or augment, the ability of an application server to handle Service Component Architecture (SCA) modules. Other servers (deployment managers and node agents) are used for managing process servers.

Related tasks

“Choosing a database” on page 8

Choosing a database depends on your operating system and on the features that you will use. During the installation procedure, the wizards prompt you to select your databases. Under some circumstances you can operate with only one database containing several tables.

“Determining the products to install” on page 6

The design of your deployment environment includes determining how many and which type of software products you may require. Based on your needs the product requirements may vary among the computer LPAR systems involved in the environment. Not every server in a deployment environment requires a WebSphere Process Server.

“Identifying available resources” on page 4

Identify your assets to make the best use of those resources already available and also to make informed purchasing decisions.

Related information

 [Configuring Business Process Choreographer](#)

Planning for a deployment environment based on one of the supplied patterns

Use this scenario when you have scalability, availability and quality of service requirements for Service Component Architecture (SCA) applications that can be met with one of the IBM-supplied patterns.

Before you begin

Familiarize yourself with the information about these topics and any related topics, if you have not done so already.

- Servers
- Clusters
- Profiles
- Choosing a database
- Deployment environments
- Deployment environment functions
- Deployment environment patterns

Diagram what hardware you are using for your deployment environment and indicate what server each piece of equipment hosts. Also indicate which servers will provide the deployment environment functions so you have a clearer idea of how to cluster the servers together.

About this task

You have analyzed your business needs and determined that a single server is insufficient to meet your needs. You need multiple servers to provide high availability, and failover. Your design fits one of the IBM-supplied deployment environment patterns.

Procedure

1. Determine the hardware and software you need to support your design.
2. Select the database product to support the deployment environment.

Some systems, such as z/OS and i5/OS do not have automated methods to create databases and tables for messaging engines and Common Event Infrastructure (CEI). When creating databases for these systems, make sure you have sufficient authorization to execute database definition scripts successfully.

Important: If your future plans include federating this environment into a deployment manager cell, make sure that you use a database and database drivers that support remote access. Examples of these types of products are Derby Network and Java toolbox JDBC.

3. Decide how you create the database tables.

Either create the tables during product installation, have the product install process create scripts to create the tables for you, or create scripts yourself to perform this step.

4. Decide which IBM-supplied pattern best fits your design.
5. Map the servers as members of the cluster that provides the function you identified in your design.

The pattern you select will map nodes to clusters and will determine the number of members and their distribution.

6. Decide how clients are to access the applications in the deployment environment.

Based on your needs there are many ways to access including Web services (SOAP/HTTP and SOAP/JMS), synchronous or asynchronous Service Component Architecture (SCA) requests, Java Message Service (JMS), MQ (either JMS or native), or through adapters. These choices impact what other software and resources you must install.

7. Decide how the applications access any resources they require.

Based on your needs there are many ways to access including Web services (SOAP/HTTP and SOAP/JMS), synchronous or asynchronous Service Component Architecture (SCA) requests, Java Message Service (JMS), MQ (either JMS or native), or through adapters. These choices impact what other software and resources you must install.

8. Decide how to install the software, create the servers and configure the created servers.

You can create and configure servers while installing the software or you can create and configure servers using the Profile Management Tool. You can also

use the administrative console or scripts to create and configure servers. Understand the benefits and drawbacks of all methods before making a choice.

9. Decide how any servers created on the same hardware will share the resources on that system.
You can either install the software into separate locations, use different profiles, or, on i5/OS use different logical partitions to accomplish the sharing.
10. Identify or create any user IDs with the authorization you need to complete installation.

What to do next

Install your deployment environment.

Related concepts

“Servers” on page 29

Servers provide the core functionality of WebSphere Process Server. Process servers extend, or augment, the ability of an application server to handle Service Component Architecture (SCA) modules. Other servers (deployment managers and node agents) are used for managing process servers.

“Deployment manager” on page 31

A deployment manager is a server that manages operations for a logical group, or cell, of other servers. The deployment manager is the central location for administering the servers and clusters.

“Managed servers” on page 33

A managed server is a server that is configured in a managed node. It provides a resource within the deployment environment that runs your applications.

“Clusters” on page 33

Clusters give your applications more capacity and higher availability than a single server.

“Deployment environment patterns” on page 34

A deployment environment pattern specifies the constraints and requirements of the components and resources involved in a deployment environment. The patterns are designed to meet the needs of most business requirements and are intended to help you create a deployment environment in the most straightforward way.

“Deployment environment functions” on page 37

To design a robust deployment environment, you need to understand the functionality each cluster can provide in a particular IBM-supplied deployment environment pattern or a custom deployment environment. This knowledge can help you make the correct decisions as to which deployment environment pattern best meets your needs.



Overview of custom deployment environment layout configuration

This overview describes two major configuration sections for custom deployment environment.

“Planning for interoperability between WebSphere Process Server and other WebSphere Application Server products” on page 6

When analyzing your software environment, you need to know whether requests can pass between the various software levels that exist in your deployment environment.

Related tasks

Chapter 4, “Planning your deployment environment,” on page 15

Setting up your deployment environment involves many decisions that affect

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

“Choosing a database” on page 8

Choosing a database depends on your operating system and on the features that you will use. During the installation procedure, the wizards prompt you to select your databases. Under some circumstances you can operate with only one database containing several tables.




“Identifying available resources” on page 4

Identify your assets to make the best use of those resources already available and also to make informed purchasing decisions.

“Determining the products to install” on page 6

The design of your deployment environment includes determining how many and which type of software products you may require. Based on your needs the product requirements may vary among the computer LPAR systems involved in the environment. Not every server in a deployment environment requires a WebSphere Process Server.

Related information

-  [Planning to install Network Deployment](#)
-  [Introduction: Clusters](#)
-  [Configuring Business Process Choreographer](#)

Planning for a custom deployment environment

Use this scenario when you have quality of service requirements or need a more complex deployment environment than those defined by the IBM-supplied patterns.

Before you begin

Important: Installing a custom deployment environment is more complicated than installing a default deployment environment and requires a understanding of network deployment, clustering and other WebSphere Process Server features. IBM recommends that you plan and implement each portion of the deployment environment separately and gradually.

Familiarize yourself with the information about these topics and any related topics, if you have not done so already.

- Servers
- Clusters
- Profiles
- Custom deployment environments and their functions
- Business Process Choreographer components and configuration

Diagram what hardware you are using for your deployment environment and indicate what server each piece of equipment hosts. Also indicate which servers will provide the deployment environment functions so you have a clearer idea of how to cluster the servers together.

Your design should specify which clusters provide messaging, Common Event Infrastructure and application support for the deployment environment.

About this task

When your design does not match any of the IBM-supplied patterns or when you want to expand an existing deployment environment use these steps. Consider using an iterative method so that you are only adding, configuring and verifying one portion of the deployment environment at a time to minimize any complexity.

Procedure

1. Select the database product to support the deployment environment.

Some systems, such as z/OS and i5/OS do not have automated methods to create databases and tables for messaging engines and Common Event Infrastructure (CEI). When creating databases for these systems, make sure you have sufficient authorization to execute database definition scripts successfully.

Important: If your future plans include federating this environment into a deployment manager cell, make sure that you use a database and database drivers that support remote access. Examples of these types of products are Derby Network and Java toolbox JDBC.

2. Decide how you create the database tables.

Either create the tables during product installation, have the product install process create scripts to create the tables for you, or create scripts yourself to perform this step.

3. Analyze the applications that you will deploy to this deployment environment to determine the clusters you require to support those applications.

4. Design the physical layout of the deployment environment.

5. Map the servers as members of the cluster that provides the function you identified in your design.

You decide on the functions the deployment environment delivers and which nodes are involved with each cluster.

6. Decide how clients are to access the applications in the deployment environment.

Based on your needs there are many ways to access including Web services (SOAP/HTTP and SOAP/JMS), synchronous or asynchronous Service Component Architecture (SCA) requests, Java Message Service (JMS), MQ (either JMS or native), or through adapters. These choices impact what other software and resources you must install.

7. Decide how the applications access any resources they require.

Based on your needs there are many ways to access including Web services (SOAP/HTTP and SOAP/JMS), synchronous or asynchronous Service Component Architecture (SCA) requests, Java Message Service (JMS), MQ (either JMS or native), or through adapters. These choices impact what other software and resources you must install.

8. Decide how to install the software, create the servers and configure the created servers.

Restriction: For a custom deployment environment in a single cell you cannot use the installer or Profile Management Tool to create servers.

9. Identify or create any user IDs with the authorization you need to complete installation.

10. Optional: Contact your database administrators, if your site policies restrict database creation and access to a centralized department.

Important: If your future plans include federating this environment into a deployment manager cell, make sure that you use a database and database drivers that support remote access. Examples of these types of products are Derby Network and Java toolbox JDBC.

11. Schedule and coordinate the installation of WebSphere Integration Developer and WebSphere Process Server to minimize impacts to the development community.

What to do next

Install your deployment environment.

Related concepts

“Servers” on page 29

Servers provide the core functionality of WebSphere Process Server. Process servers extend, or augment, the ability of an application server to handle Service Component Architecture (SCA) modules. Other servers (deployment managers and node agents) are used for managing process servers.

“Deployment manager” on page 31

A deployment manager is a server that manages operations for a logical group, or cell, of other servers. The deployment manager is the central location for administering the servers and clusters.

“Managed servers” on page 33

A managed server is a server that is configured in a managed node. It provides a resource within the deployment environment that runs your applications.

“Clusters” on page 33

Clusters give your applications more capacity and higher availability than a single server.

“Deployment environment functions” on page 37

To design a robust deployment environment, you need to understand the functionality each cluster can provide in a particular IBM-supplied deployment environment pattern or a custom deployment environment. This knowledge can help you make the correct decisions as to which deployment environment pattern best meets your needs.

 Overview of custom deployment environment layout configuration

This overview describes two major configuration sections for custom deployment environment.

“Planning for interoperability between WebSphere Process Server and other WebSphere Application Server products” on page 6

When analyzing your software environment, you need to know whether requests can pass between the various software levels that exist in your deployment environment.

Related tasks

Chapter 4, “Planning your deployment environment,” on page 15

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

“Choosing a database” on page 8

Choosing a database depends on your operating system and on the features that you will use. During the installation procedure, the wizards prompt you to select your databases. Under some circumstances you can operate with only one database containing several tables.

“Identifying available resources” on page 4

Identify your assets to make the best use of those resources already available and also to make informed purchasing decisions.

“Determining the products to install” on page 6

The design of your deployment environment includes determining how many and which type of software products you may require. Based on your needs the product requirements may vary among the computer LPAR systems involved in the environment. Not every server in a deployment environment requires a WebSphere Process Server.

Related information

 [Planning to install Network Deployment](#)

 [Introduction: Clusters](#)

 [Configuring Business Process Choreographer](#)

Profiles

A profile defines a unique runtime environment, with separate command files, configuration files, and log files. Profiles define three different types of environments: stand-alone server, deployment manager, and managed node.

Using profiles you can have more than one runtime environment on a system, without having to install multiple copies of the WebSphere Process Server binary files.

When you install WebSphere Application Server for z/OS, you configure the product using the ISPF Customization Dialog, which creates a profile. After you have installed WebSphere Process Server, you must augment the profile to define the node either as a deployment manager or stand-alone server for WebSphere Process Server. If you have created a deployment manager, you can then create one or more managed nodes. This automatically creates a profile to define the runtime environment for each node.

Note: On distributed platforms, each profile has a unique name. On z/OS all the profiles are named “default”; you cannot rename, edit, copy, or delete profiles on z/OS.

The profile directory

Every profile in the system has its own directory containing all its files. You specify the location of the profile directory when you create the profile: by default it is in the profiles directory in the directory where WebSphere Process Server was installed, for example, `installation_file_system_root/AppServer/profiles/default`.

Related concepts

“Stand-alone server” on page 29

A stand-alone server provides an environment for deploying Service Component Architecture (SCA) modules in one server process. This server process includes, but is not limited to, an administrative console, a deployment target, the messaging support, the business rules manager, and a Common Event Infrastructure server.

“Deployment manager” on page 31

A deployment manager is a server that manages operations for a logical group, or cell, of other servers. The deployment manager is the central location for administering the servers and clusters.

“Managed nodes” on page 31

A managed node is a node that has been federated into a deployment manager cell. In a managed node, you can configure and run managed servers.

Servers

Servers provide the core functionality of WebSphere Process Server. Process servers extend, or augment, the ability of an application server to handle Service Component Architecture (SCA) modules. Other servers (deployment managers and node agents) are used for managing process servers.

A process server can be either a *stand-alone server* or a *managed server*. A managed server can optionally be a member of a *cluster*. A collection of managed servers, clusters of servers, and other middleware is called a *deployment environment*. In a deployment environment, each of the managed servers or clusters is configured for a specific function within the deployment environment (for example, destination host, application module host, or Common Event Infrastructure server). A stand-alone server is configured to provide all of the required functions.

Servers provide the runtime environment for Service Component Architecture (SCA) modules, for the resources that are used by those modules (data sources, activation specifications, and JMS destinations), and for IBM-supplied resources (message destinations, Business Process Choreographer containers, and Common Event Infrastructure servers).

A *node agent* is an administrative agent that represents a node to your system and manages the servers on that node. Node agents monitor servers on a host system and route administrative requests to servers. The node agent is created when a node is federated to a deployment manager.

A *deployment manager* is an administrative agent that provides a centralized management view for multiple servers and clusters.

A stand-alone server is defined by a stand-alone profile; a deployment manager is defined by a deployment manager profile; managed servers are created within a *managed node*, which is defined by a custom profile.

Stand-alone server

A stand-alone server provides an environment for deploying Service Component Architecture (SCA) modules in one server process. This server process includes, but is not limited to, an administrative console, a deployment target, the messaging support, the business rules manager, and a Common Event Infrastructure server.

You can deploy your own solutions to a stand-alone server, but a stand-alone server cannot provide the capacity, scalability, or robustness that is generally required of a production environment. For your production environment, it is better to use a network deployment environment.

It is possible to start off with a stand-alone server and later include it in a network deployment environment, by federating it to a deployment manager cell, *provided that no other nodes have been federated to that cell*. It is not possible to federate

multiple stand-alone servers into one cell. To federate the stand-alone server, use the administrative console of the deployment manager or the **addNode** command. The stand-alone server must not be running when you federate it using the **addNode** command.

A stand-alone server is defined by a stand-alone server profile.

Related concepts

“Profiles” on page 28

A profile defines a unique runtime environment, with separate command files, configuration files, and log files. Profiles define three different types of environments: stand-alone server, deployment manager, and managed node.

Messaging engines

A messaging or queue destination host provides the messaging function within a server. A server becomes the messaging destination host when you configure it as the messaging target.

Data sources

Data sources provide the link between applications and relational databases.

Service integration buses

A service integration bus is a managed communication mechanism that supports service integration through synchronous and asynchronous messaging. A bus consists of interconnecting messaging engines that manage bus resources. It is one of the WebSphere Application Server technologies on which WebSphere Process Server is based.

Network deployment

Network deployment provides the capacity, scalability, and robustness that is generally required of a production environment. In network deployment, a group of servers can be used collaboratively to provide workload balancing and failover. The servers are managed centrally, using a single administrative console.

Network deployment in WebSphere Process Server builds upon network deployment functions implemented in WebSphere Application Server Network Deployment. If you are familiar with network deployment in WebSphere Application Server Network Deployment, the concepts are the same. WebSphere Process Server adds the concept of deployment environments to network deployment.

What you need to read about network deployment depends on whether you are upgrading WebSphere Application Server Network Deployment or implementing WebSphere Process Server with no previous experience of WebSphere Application Server Network Deployment.

Upgrading WebSphere Application Server Network Deployment

WebSphere Application Server Network Deployment, as its name implies, supports network deployment of applications. If you already have a WebSphere Application Server Network Deployment installation, which you are upgrading with WebSphere Process Server, you are familiar with the concept of network deployment. You probably have one or more network deployment cells each with its deployment manager and managed nodes. You can *augment* their profiles to support WebSphere Process Server, using the WebSphere Process Server Profile

Management tool. After augmentation, the servers still continue to function as application servers, but they are also capable of supporting Service Component Architecture (SCA) modules.

Implementing WebSphere Process Server network deployment

In network deployment, you install WebSphere Process Server on one or more host systems and then create a *deployment environment*. IBM supplies a number of deployment environment *patterns* to help you configure the *clusters*, *servers* and middleware that you need to host Service Component Architecture (SCA) modules.

Related information

 WebSphere Application Server Network deployment and single server (all operating systems) information center

Deployment manager

A deployment manager is a server that manages operations for a logical group, or cell, of other servers. The deployment manager is the central location for administering the servers and clusters.

You use the administrative console of the deployment manager to manage the servers and clusters in the cell. This includes configuring servers and clusters, adding servers to clusters, starting and stopping servers and clusters, and deploying Service Component Architecture (SCA) modules to them.

Although the deployment manager is a type of server, you cannot deploy modules to the deployment manager itself.

Related concepts

Chapter 3, “Using multiple platforms within a cell,” on page 13

With careful planning, you can create a deployment manager cell that includes nodes on both distributed, i5/OS and z/OS operating system platforms.

“Profiles” on page 28

A profile defines a unique runtime environment, with separate command files, configuration files, and log files. Profiles define three different types of environments: stand-alone server, deployment manager, and managed node.

Managed nodes

A managed node is a node that has been federated into a deployment manager cell. In a managed node, you can configure and run managed servers.

The servers that are configured on a managed node make up the resources of your deployment environment. These servers are created, configured, started, stopped, managed and deleted using the administrative console of the deployment manager. When a node is federated, a node agent process is automatically created. This node agent must be running to be able to manage the configuration of the profile. For example, when you do the following tasks:

- Start and stop server processes.
- Synchronize configuration data on the deployment manager with the copy on the node.

However, the node agent does not need to be running in order for the applications to run or to configure resources in the node.

A managed node can contain one or more servers, which are managed by a deployment manager. You can deploy solutions to the servers in a managed node, but the managed node does not contain a sample applications gallery.

Related concepts

Chapter 3, “Using multiple platforms within a cell,” on page 13

With careful planning, you can create a deployment manager cell that includes nodes on both distributed, i5/OS and z/OS operating system platforms.

“Profiles” on page 28

A profile defines a unique runtime environment, with separate command files, configuration files, and log files. Profiles define three different types of environments: stand-alone server, deployment manager, and managed node.

Deployment environments

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

Planning deployment environments requires that you design the physical layout (topology) of the deployment environment so you can meet your business needs for capacity, availability, scalability and failover support. Some key aspects of the design involve the number and relative placement of the servers on the hardware that comprises your deployment environment.

Stand-alone environment

It is possible to deploy Service Component Architecture (SCA) modules to a *stand-alone server*. This is the easiest environment to set up, but a stand-alone server does not connect to other servers, its capacity is limited to the resources on the same computer system, and it does not include failover support.

If you require more capacity, scalability, availability or failover support than a stand-alone server provides, you need to consider a deployment environment of interconnected servers.

Interconnected servers

A deployment environment is a collection of interconnected servers that supports WebSphere Process Server application components such as:

- Business Process Choreographer.
- Business rules.
- Mediations.
- Relationships.

The environment also supports WebSphere Enterprise Service Bus and WebSphere Application Server based servers.

The servers in a deployment environment can run on one or more host systems. Servers can be grouped into *clusters* to support load-balancing and failover.

In addition to the performance, availability, scalability, isolation, security, and stability characteristics that cannot be provided by a stand-alone server, a deployment environment of interconnected servers or clusters has the additional

advantage that you can manage all the servers or clusters from a centralized *deployment manager*.

Deployment environment patterns

Creating a deployment environment is straightforward if you use one of the supplied deployment environment patterns, provided you know your requirements and plan accordingly. There are three patterns:

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

If none of the patterns meet your requirements, you can plan and create your own customized deployment environment.

Deciding when to create the deployment environment

In addition to planning the deployment environment, you must also decide when to create it. You can choose one of the following options:

1. Create the deployment environment when you install the software, using the installation wizard or silent installation.
2. Install the software on the host systems that you intend to use. Then use the Profile Management Tool or `manageprofiles` command to create the deployment environment.
3. Install the software on the host systems that you intend to use. Use the Profile Management Tool or `manageprofiles` command to create deployment manager and custom profiles. Then create the deployment environment using the administrative console of the deployment manager.

The option you choose depends on the complexity of the deployment environment. If one of the supplied deployment environment patterns meets your requirements, choose option 1 or 2; if none of the supplied patterns meet your requirements, choose option 3.

Regardless which method you use to create the deployment environment, you can still manage some aspects of the deployment environment using the administration console. (For example, you can add more nodes to the deployment environment.) However some aspects you cannot change if you created the deployment environment using options 1 or 2. (For example, you cannot change database types.)

Managed servers

A managed server is a server that is configured in a managed node. It provides a resource within the deployment environment that runs your applications.

A managed server can optionally be a member of a cluster. To provide a robust, production-scale process server, configure a deployment environment containing clusters of managed servers.

You configure and manage the servers and clusters using the administrative console of the deployment manager.

Clusters

Clusters give your applications more capacity and higher availability than a single server.

A *Cluster* is a set of managed servers that provide high availability and workload balancing for applications. Members of a cluster can be servers located on various hosts or servers located on the same host (the same node). To best achieve high availability and workload balancing, place each cluster member on different host machines.

A clustered environment provides the following benefits:

- **Workload balancing:** By running application images on multiple servers, a cluster balances an application workload across the servers in the cluster.
- **Processing power for the application:** You can add processing power to your application by configuring server hardware as cluster members to support the application.
- **Application availability:** When a server fails, the application continues to process work on other servers in the cluster. This allows recovery efforts to proceed without affecting the application users.
- **Maintainability:** You can stop a server for planned maintenance without stopping application processing.
- **Flexibility:** You can add or remove capacity as needed by using the administrative console of the deployment manager.

By nature, the z/OS environment is clustered and can provide a highly scalable environment without all the complexities of separate clustered servers. Separate clustered servers can be used for multisystem nodes to provide additional availability.

Deployment environment patterns

A deployment environment pattern specifies the constraints and requirements of the components and resources involved in a deployment environment. The patterns are designed to meet the needs of most business requirements and are intended to help you create a deployment environment in the most straightforward way.

Each of the three deployment environment patterns addresses a specific set of requirements. Most requirement sets can be met using one of these patterns.

These descriptions are not intended as installation instructions. To create a deployment environment that fits one of the patterns, you make selections during installation, profile creation, or in the administrative console.

Single cluster pattern

The single cluster pattern is suitable for scenarios that are focused on running applications and on synchronous invocations. Any messaging requirements should be kept to a minimum with this pattern. Service Component Architecture (SCA) internal asynchronous invocations, the Java Message Service (JMS) and MQ messaging bindings do not support multiple messaging engines in the same cluster. If your modules require any of these, choose one of the other patterns, in which the messaging infrastructure is in a separate cluster from the application deployment target.

All components are run on a single cluster:

- Service Component Architecture (SCA) application bus members
- SCA system bus members
- Business Process Choreographer bus members

- Business Process Choreographer components such as the explorer and observer
- Business Process Choreographer container
- Common Event Interface (CEI) bus members
- CEI server
- Business Rules manager
- Application deployment target

You configure the application deployment target to support SCA applications and Business Process Choreographer components.

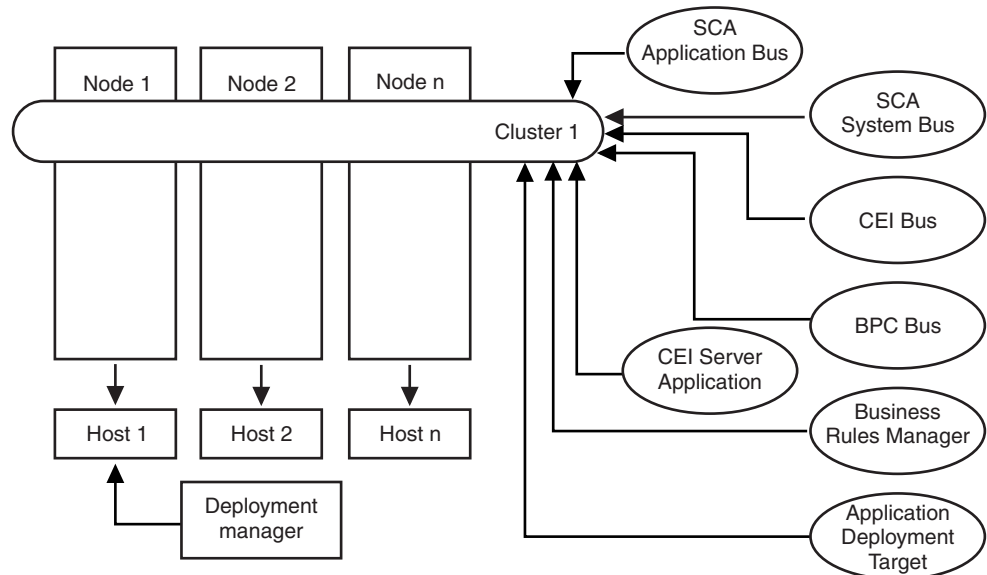


Figure 1. Single cluster pattern

Remote messaging pattern

The remote messaging pattern provides a separate cluster for the messaging role. This pattern is suitable for scenarios involving asynchronous invocations, because the cluster can be scaled for this load. The components are divided between the two clusters.

Remote messaging cluster:

- Service Component Architecture (SCA) application bus members
- SCA system bus members
- Business Process Choreographer (BPC) bus members
- Common Event Interface (CEI) bus members

Support infrastructure and application deployment target cluster:

- CEI server application
- Business Rules manager
- Business Process Choreographer components such as the explorer and observer
- Application deployment target

You configure the application deployment target to support SCA applications and Business Process Choreographer components.

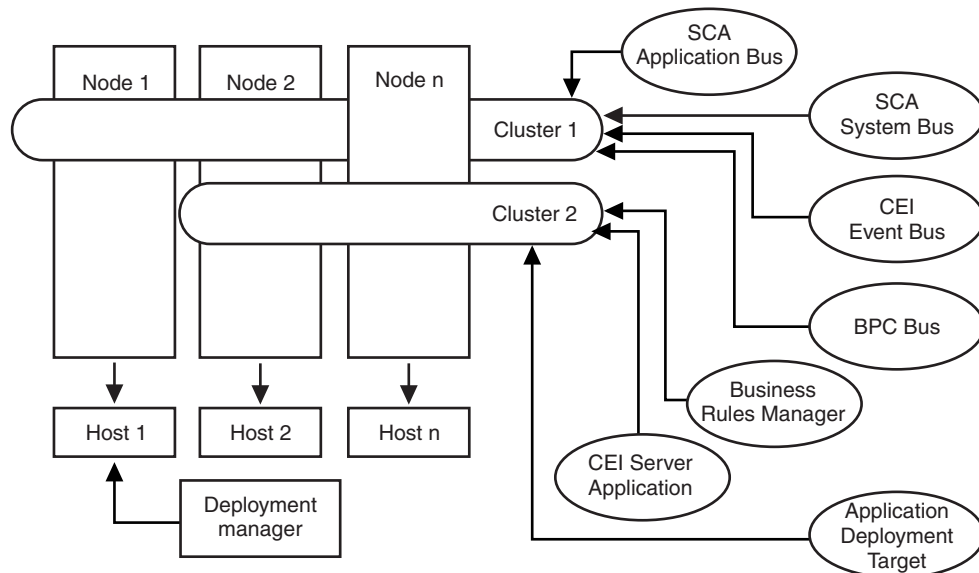


Figure 2. Remote messaging pattern

Remote messaging and remote support pattern

With this three-cluster pattern, resources are allocated to the cluster that handles the highest loads. This pattern is the most flexible and versatile, and is preferred by most users. The components are divided between the three clusters.

Remote messaging infrastructure cluster:

- Service Component Architecture (SCA) application bus members
- SCA system bus members
- Business Process Choreographer (BPC) bus members
- Common Event Interface (CEI) bus members

Remote support infrastructure cluster:

- CEI server application
- Business Rules manager
- Business Process Choreographer components such as the explorer and observer

Application deployment cluster:

- Application deployment target
- Business Process Choreographer container

You configure the application deployment target to support SCA applications and Business Process Choreographer components.

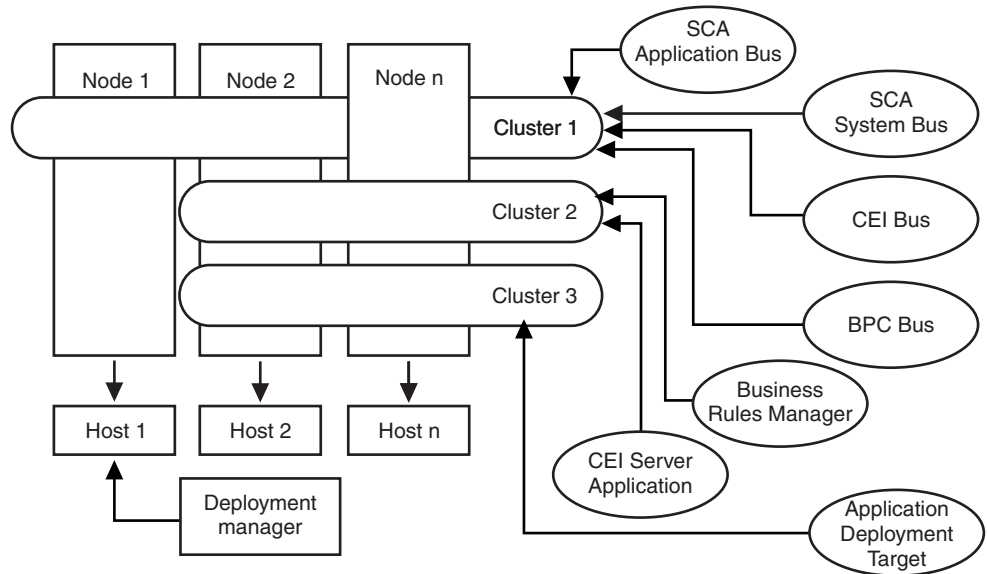


Figure 3. Remote messaging and support pattern

Resource allocation example

The following figure shows one way in which resources might be allocated using the remote messaging and remote support pattern. Because the heaviest load for this installation is for application use, there are more resources allocated (server1, server2 and server6) for the application deployment target cluster (Cluster 3) than the other functions.

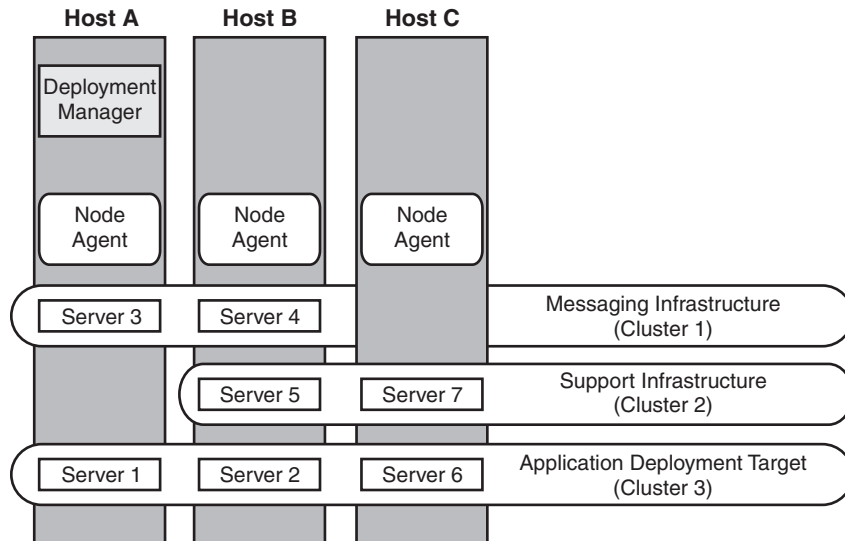


Figure 4. Resource allocation example

Deployment environment functions

To design a robust deployment environment, you need to understand the functionality each cluster can provide in a particular IBM-supplied deployment

environment pattern or a custom deployment environment. This knowledge can help you make the correct decisions as to which deployment environment pattern best meets your needs.

For network deployment, clusters can collaborate to provide specific functionality to the environment. Depending on your requirements, you assign specific functions to each cluster within the deployment environment, to provide performance, failover, and capacity.

IBM-supplied deployment environment patterns

The clusters configured in a deployment environment pattern provide these functions:

Application deployment target

Consists of a cluster to which you install applications. Depending on which deployment environment pattern you choose, the application deployment target may also provide messaging infrastructure and supporting infrastructure functions. Choose the appropriate product depending on the type of applications you intend to deploy.

- Install WebSphere Process Server, if the applications contain human task or business process artifacts.
- Install WebSphere Enterprise Service Bus, if the applications contain mediation modules only.

In a single cluster pattern, the application deployment target provides the entire functionality of the deployment environment.

Supporting infrastructure

Consists of a cluster that hosts the Common Event Infrastructure (CEI) server and other infrastructure services used to manage your system. These services include:

- Business rules
- Selectors
- human tasks
- Business processes

Important: You must use a custom profile with the same product functionality for this node as you did for the application deployment target cluster.

Messaging infrastructure

Consists of a cluster where the messaging engines are located. The messaging engines enable communication amongst the nodes in the deployment environment. Your cluster can consist of members on nodes created with WebSphere Application Server instead of WebSphere Process Server if the cluster solely provides the messaging function.

Custom deployment environments

Custom deployment environments allow for more varying topologies. If you need more processing capabilities for applications, if you need to spread the supporting infrastructure functions over more clusters, or if you need to consolidate supporting infrastructure for several servers or clusters onto one cluster, you can achieve this with custom deployment environments.

You divide the function amongst clusters using *collaborative units*. Collaborative units allow functions to be spread depending on your needs onto different clusters and servers that work together as a unit to further increase isolation, function consolidation, throughput capabilities and failover.

The administrative console groups collaborative units as follows:

Messaging

Messaging units provide the same support as the messaging infrastructure for an IBM-supplied deployment environment pattern. There is a server within the cluster that contains a local messaging engine and the other servers and clusters within the unit use that messaging engine as a destination for messages.

Common Event Infrastructure

Common Event Infrastructure units consist of the server hosting the CEI server and other clusters and servers that support the CEI functions. Common base events received at each cluster or server in the unit are routed to the server hosting the CEI server. Use as many collaborative units as your deployment environment needs to host more CEI servers to isolate events from different event sources

Application support

Application support units are similar to the supporting infrastructure for an IBM-supplied deployment environment pattern. They group clusters and servers onto which you are deploying your applications. They differ in that they allow for more than one business container or Service Component Architecture (SCA) support cluster to be defined in a deployment environment by defining more collaborative units. One unit defines a business process cluster and one or more SCA support clusters and support applications on the same or different clusters in that unit.

Choosing your deployment environment pattern

You can configure your deployment environment by choosing one of the IBM-supplied patterns or by creating your own custom deployment environment. This topic lists the features supported by each IBM-supplied pattern.

Before you begin

You should familiarize yourself with the information in:

- Assessing your business requirements
- Identifying available resources

About this task

You have completed designing your deployment environment and you need to determine whether one of the IBM-supplied patterns that are supported through the various product wizards would meet your needs.

Important: If you intend to use a z/OS system or cluster in your deployment environment, make sure you determine which function that server or cluster will provide. You cannot mix z/OS systems with other systems in the same cluster so your design must take this fact into account.

Procedure

1. Determine which IBM-supplied pattern best meets your business needs

Deployment Environment Pattern	Features
Single cluster	Messaging, application deployment target, and application support functions are contained in a single cluster. This pattern is useful for synchronous messaging, proof of concept, or application testing environments.
Remote messaging	This pattern separates the messaging environment from the application deployment target and the application support functions. Use this pattern when message throughput is a critical requirement for your daily operation. This pattern is highly recommended for asynchronous messaging and transactional systems.
Remote messaging and remote support	This pattern separates messaging, Common Event Infrastructure (CEI), application deployment target and application support functions into distinct clusters. Most businesses can use this pattern to support their deployment environments as it is designed for performance and isolation of transactional processing from messaging and other support functions.

2. Optional: If you need to provide only mediation services, you will install Enterprise Service Bus instead of WebSphere Process Server.
3. If none of the IBM-supplied patterns meets your business needs, you can implement a custom deployment environment.

Note: To implement a custom pattern requires a good knowledge of how deployment environments work and an understanding of how to correctly configure servers and clusters.

What to do next

Install and configure the product.

Related concepts

“Deployment environment patterns” on page 34

A deployment environment pattern specifies the constraints and requirements of the components and resources involved in a deployment environment. The patterns are designed to meet the needs of most business requirements and are intended to help you create a deployment environment in the most straightforward way.

Chapter 5. Implementing a deployment environment

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

Before you begin

- Plan your topology and record the decisions you make about:
 - The servers and clusters involved.
 - The number of databases required.
 - Which database tables belong in which databases
 - Any required userids and authentication roles
 - What function each cluster involved in the deployment environment provides
 - Which method you are using to implement the deployment environment
- Make sure the systems on which you are installing the product meet the hardware and software requirements.
- Prepare the operating system for installation.
- Install and configure your database product following the product documentation. You must:
 - Configure the product as a server.
 - Define a userid for WebSphere Process Server to use to access the data and tables within the database.
 - **Optional:** Create the WebSphere common database, called WPRCSDB by default.
If you created this database during product installation or through the Profile Management Tool, bypass this step.
 - Create all other databases you need for your configuration. If you do not create a database for a specific function, the system uses the WebSphere common database.
- Synchronize the system clocks on all the servers. When adjusted for the same time zone, the clocks must be within five minutes of each other.
- Make sure all servers involved in the topology can be located by both IP address and Domain Name Server (DNS) name.
- Make sure you have a user ID that has the appropriate authority to create directories and files on all systems.
- Make sure you perform any other preparation that might be needed to coexist with other products and provide any needed redundancy.

About this task

Now that you have completed planning your deployment environment and performed all the prerequisite tasks, install and configure the servers and clusters involved in your design. Regardless of the method you choose to implement the deployment environment, the following steps outline creating a single cell of that design.

Note: This procedure covers all of the steps required to implement a deployment environment and the order might differ slightly depending on your installation method.

Procedure

1. Install the product binaries on all systems involved in the deployment environment and verify that the software is correctly installed.
2. Create the deployment manager.
3. Start the deployment manager.
4. Create as many managed nodes as you need.
5. Federate the nodes from step 4 to the deployment manager created in step 2.
6. Configure the cell.

Important: The configuration can take a long time depending on your deployment environment. To prevent the process from timing out, set the SOAP request timeout on the deployment manager to a large value, for example 1800 seconds. See “Timeout properties summary” in the WebSphere Application Server information center.

This involves creating the clusters to perform the functions you defined to them in your design and then adding members to those clusters.

If your design implements a patterned deployment environment, the system creates all needed clusters and defines cluster members to provide all necessary functions. Depending on the pattern you selected, this includes clusters for application deployment, messaging support and infrastructure support.

If your design implements a custom deployment environment, you must create all the clusters needed to provide the necessary functions. These functions include messaging support for application deployment, application support and Common Event Infrastructure support.

7. Configure the databases or database tables required for your topology, if you chose deferred table creation.

Configuration consists of running the scripts generated by the deferred option.

- a. Configure the common database tables. This table is in the common database.
- b. Configure the messaging engine database tables. This table is in the common database.
- c. Optional: Configure the Business Process Choreographer database tables.

If your system is not using business processes or human tasks, bypass this step. This table resides in whichever database you configured for use by the Business Process Choreographer, which is named BPEDB by default.

You may also need to configure the Business Process Choreographer Observer database tables, if you are using the Business Process Choreographer Observer. This table resides in the Business Process Choreographer Observer database, which is named BPEODB by default.

- d. Create the enterprise service bus logging mediation database table. This table is in the common database.
 - e. Configure the Common Event Infrastructure database.
8. Install and configure a routing server. This could be an IBM HTTP server or another server of your choosing. This server allows clients to access the applications within this topology.
 9. Verify the installation by installing and running test applications.

What to do next

- Create another cell, if desired.
- Deploy the applications that are to run in this deployment environment.

Related concepts

“Deployment environment functions” on page 37

To design a robust deployment environment, you need to understand the functionality each cluster can provide in a particular IBM-supplied deployment environment pattern or a custom deployment environment. This knowledge can help you make the correct decisions as to which deployment environment pattern best meets your needs.

“Deployment environment patterns” on page 34

A deployment environment pattern specifies the constraints and requirements of the components and resources involved in a deployment environment. The patterns are designed to meet the needs of most business requirements and are intended to help you create a deployment environment in the most straightforward way.

“Deployment environments” on page 32

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

“Clusters” on page 33

Clusters give your applications more capacity and higher availability than a single server.

“Servers” on page 29

Servers provide the core functionality of WebSphere Process Server. Process servers extend, or augment, the ability of an application server to handle Service Component Architecture (SCA) modules. Other servers (deployment managers and node agents) are used for managing process servers.

“Deployment manager” on page 31

A deployment manager is a server that manages operations for a logical group, or cell, of other servers. The deployment manager is the central location for administering the servers and clusters.

Messaging engines

A messaging or queue destination host provides the messaging function within a server. A server becomes the messaging destination host when you configure it as the messaging target.

Related tasks

Stopping and restarting the deployment manager

After any configuration changes to the deployment manager, you must stop and restart the deployment manager before those changes take effect.






Federating custom nodes to a deployment manager

Learn about how to use the **addNode** command to federate a custom node into a deployment manager cell.

Related information

Configuring Business Process Choreographer

Communicating with Web servers

-  Installing IBM HTTP server
-  wsadmin tool
-  Managing node agents
-  Starting clusters
-  Stopping clusters

Notices

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows: (c) (your company name) (year). Portions of

this code are derived from IBM Corp. Sample Programs. (c) Copyright IBM Corp. _enter the year or years_. All rights reserved.

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

Programming interface information

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

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

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

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

Trademarks and service marks

IBM, the IBM logo, DB2, i5/OS, WebSphere, and z/OS are registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

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

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

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

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

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



IBM WebSphere Process Server for z/OS, Version 6.1.0



Printed in USA