WebSphere Process Server for z/OS

Planning the Installation

Version 7.0.0

30 April 2010

This edition applies to version 7, release 0, modification 0 of WebSphere Process Server for z/OS (product number 5655-N53) and to all subsequent releases and modifications until otherwise indicated in new editions.

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

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

Contents

Planning overview

The activities associated with implementing WebSphere® Process Server as part of a BPM solution vary depending on the planned usage.

How planned usage affects planning activities

The following table shows how the intended use of WebSphere Process Server affects the amount of user planning activities:

Table 1. Intended use of WebSphere Process Server

Intended use	Configuration path and planning activities
A single server UTE	The stand-alone profile configuration path - with little planning required.
A clustered test environment	Standard Remote Messaging Remote Support topology of Deployment Environment - with little planning required.
A production environment, with good flexibility	Standard Remote Messaging Remote Support topology of Deployment Environment - with little planning required.
A highly-optimized production environment	Detailed planning required as described in Planning activities for a highly optimized production environment.

Planning activities for a highly optimized production environment

Planning for a highly optimized production environment involves assessment, and design activities that affect the way you install and configure the product, and how you create and generate the environment.

Successful planning requires input from multiple sources as follows:

- Solution architects

  The persons responsible for overseeing the architecture of the BPM solution, and for making recommendations on the products needed to support the business goals and objectives, and how to best use the products.
- Business analysts and business leaders

  The persons who discover the business intent of proposed BPM solutions and map that intent to business capabilities and process maps. The business analysts and business leader identify and prioritize options regarding business processes. The business analyst captures and defines the "as-is" processes (current process) and the to-do processes (improvements or changes to existing processes). The business analyst creates mock-up forms to validate and visualize human interaction and works closely with the solution architect by providing key input. This enables the solution architect to visualize a BPM solution that can address the needs of the business.
- Systems personnel

  The persons responsible for maintaining the workstations on which BPM run time products are installed and on which SCA applications are deployed.

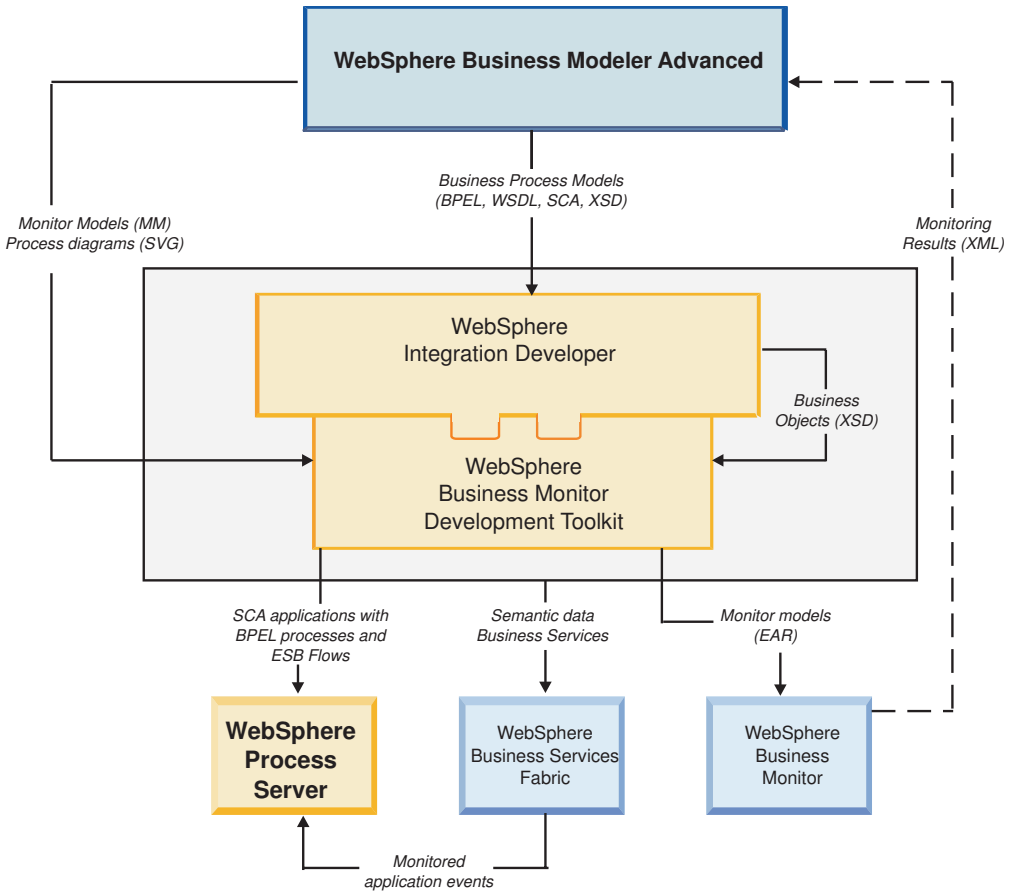
Systems personnel set up the IT environments and prepare and deploy production artifacts. The system personnel also monitor the health of the BPM solution to ensure process integrity.

- Database administrator

  The persons responsible for installing and maintaining the database system.
- Application developers

  The persons who create the applications using BPM application development tools such as WebSphere Integration Developer, WebSphere Business Monitor Development Toolkit and WebSphere Business Modeler.
- IBM services personnel with expertise in BPM solutions

The following diagram shows how IBM WebSphere products (development tools and associated runtimes) might be used to implement a business process management (BPM) solution.



See the *Product family overview* for more information about the products listed in the diagram.

A key aspect of the WebSphere Process Server topology design involves the number of physical workstations (in distributed environments), the number of servers on those workstations, and the number of clusters needed to provide your production environment with the processing capabilities required by your business. In addition, a production topology includes other non-WebSphere Process Server supporting resources such as a user registry (for security), one or more HTTP servers (for Web content), necessary firewalls, load balancers, and so on.

You should carefully plan any WebSphere Process Server production deployment topology, with a goal of meeting the requirements of the work to be performed on it. There are multiple aspects to consider, including the following:

- Number of physical workstations and hardware resources that you require
- Number of clusters and cluster members required to support your business
- Number of databases required
- Authentication roles and security considerations
- Method that you will use to implement the deployment environment

The following diagram provides a high-level overview of the tasks associated with planning, installing, and configuring WebSphere Process Server. The decisions that you make during the "Planning" phase have an effect on the tasks listed under the "Installing" and "Configuring" phase.

Note that the configuring phase consists of two types of tasks - product configuration tasks and environment configuration tasks. Product configuration tasks pertain to setting up the product profiles and configuring the database, while environment configuration pertains to tasks for setting up and generating the deployment environment. A *deployment environment* is the collection of configured clusters, servers, and middleware that hosts the SCA modules and applications that support the business and the required messaging infrastructure.

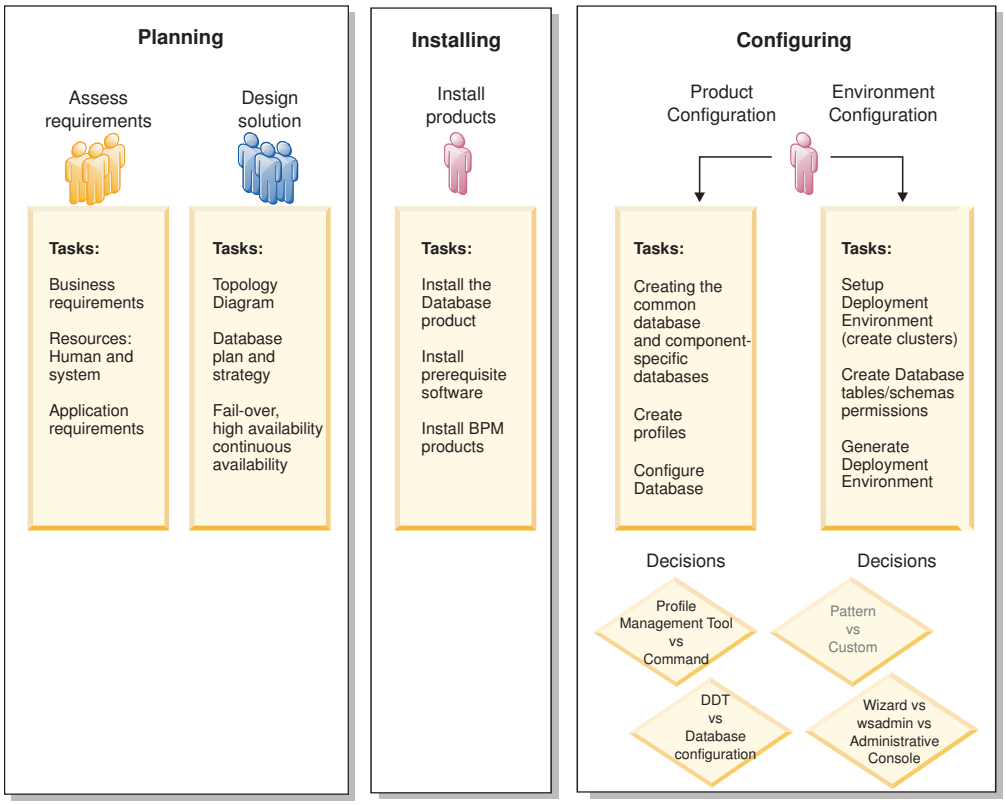
Installing the database product is part of the *installing product* phase.

Making the database available for use by WebSphere Process Server can be part of the product configuration phase.

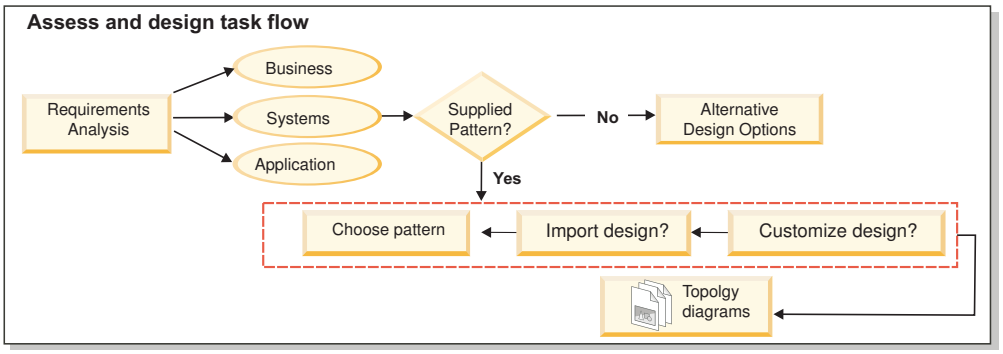
You have some flexibility on when to create the WebSphere Process Server databases. You can create databases before installing WebSphere Process Server, after installing WebSphere Process Server (but before profile creation), or after profile creation. The important thing to remember is that the databases (that is, databases, tables, schemas, and so on) must exist before the WebSphere Process Server servers try to use them.

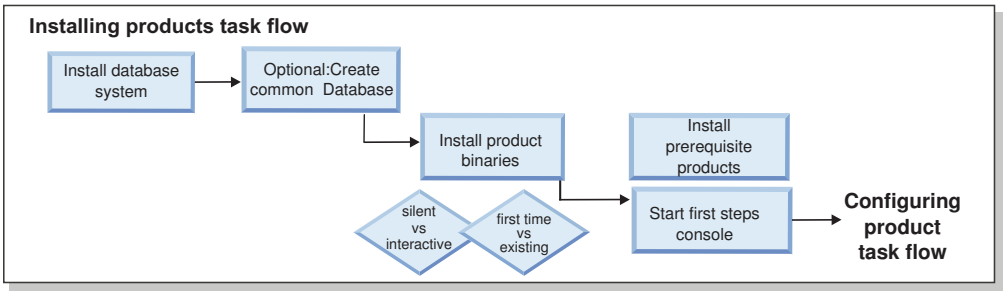
The Deployment Manager process uses the common database. Because you create the deployment manager "early-on" in the cell creation process, creating the common database that is needed for the deployment manager can be considered a *product configuration* task.

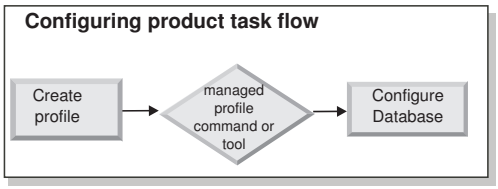
Creating the clusters of the deployment environment is part of the *environment configuration* tasks. Clusters in a deployment environment require specific tables, schemas, and user permissions based on the functional purpose of the cluster. For example, a deployment environment could include both a messaging infrastructure cluster and an application deployment target cluster. The messaging infrastructure cluster will access messaging engine database tables and the application deployment target cluster will access Business Process Choreographer database tables.



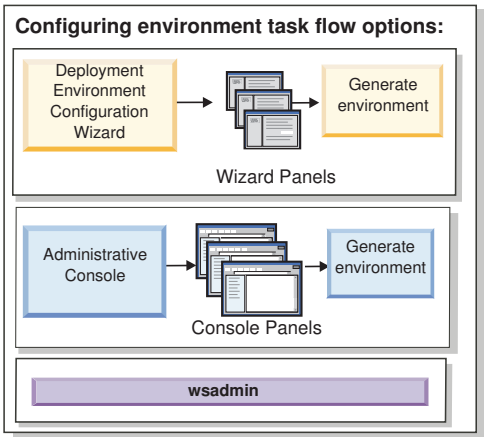
The following diagram provides a task-flow as it pertains to planning, installing, and configuring the product and the environment.







Note: You can create the product databases before installing the product, after installing the product (but before profile creation,) or after you create the profiles



Related concepts

Product family overview

Service Component Architecture

"Understanding topologies" on page 63
Your choice of a topology depends on your business goals and objectives and on the design and purpose of the applications that will be deployed to your environment.

Related information

Determining your software needs
To minimize rework and outages, take the time to study your current environment before moving forward and making installation and configuration decisions. Your current business requirements 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 requirements.

 BPM Solution Implementation Guide

 z/OS: WebSphere Business Process Management V6.2 Production Topologies

Determining your software needs

To minimize rework and outages, take the time to study your current environment before moving forward and making installation and configuration decisions. Your current business requirements 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 requirements.

There are several factors that go into determining your software needs.

These factors can be organized into the following categories.

- Resource availability and constraints

  This category includes understanding the product hardware and software requirements as well as understanding your own system resource constraints and the availability of resources to administer and maintain your system.
- Business requirements

  Include an understanding of the applications to be deployed to the run time environment, as well as the intended use of the configured environment.
- Which products (and the versions of these products) to install to meet your requirements

In order to make educated choices pertaining to all these factors, you must understand the following:

- Terminology as it applies to environment configuration
- The administrative architecture of the product (or group of BPM products) that you will install, configure, administer, and maintain.
- The available configuration options (by way of supplied patterns) and how to determine if a pattern addresses your intended use of the product.
- Supported methods of implementation, including an understanding of the different task flows for installing the product and configuring the environment.

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

Note: For the latest information about platform-specific disk space requirements, supported operating systems, database versions that are supported and the operating system fixes and patches that you must install to have a compliant operating system, see WebSphere Process Server detailed system requirements at http://www.ibm.com/support/docview.wss?uid=swg27006205 and select the link to your version of WebSphere Process Server.

Assessing requirements

Your current 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.

WebSphere Process Server comes with deployment environment patterns designed to meet the requirements of both production and test environments.

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. Consider how processes and applications interact with existing services and back-end systems.
2. Consider how applications process data and how data flows through your system to address a specific business need

   An understanding of how data persists across retrievals, sessions, processes, and other boundaries are all things you should consider when developing a solution and when configuring the environment that will support that solution.

   Consider the following items regarding the applications to be deployed to your environment:

   - Application invocation patterns

     Understanding the how the runtime handles asynchronous invocations and how the SCA run time leverages the underlying message system to implement asynchronous invocations.

     Different applications have different needs, determined by such factors as the application modules' export types, component types, interactions between components, import types, resources needed such as databases or JMS resources, the need for business events, and their transmission mechanism.
   - Types of business processes that you plan to implement (interruptible versus non-interruptible)

     Non-Interruptible business processes, or *microflows* are short running business processes that run in one transaction or without a transaction. Non-Interruptible business processes are fast with little performance overhead; all activities within it are processed in a single thread.

     Interruptible business processes, or *macroflows* are long running business processes that contain a set of activities each of which is performed in it's own transaction.

     Interruptible business processes can include activities that require human intervention and/or calls to remote systems. Asynchronous activities such as those that require human intervention (Staff) or require an outside event to occur will cause a business process to be interruptible in execution as these activities may take minutes, hours, or even days to occur and complete.
3. Plan for the future.
   - If the project or company expands, how will the company's IT department adapt to the expansion?
   - 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 available resources.

Related concepts

Business process types

Related information

SCA asynchronous invocation patterns in depth

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. Determine the number of physical computer systems you will use and itemize each piece of physical hardware.

   Note the following:

   - 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 information

Hardware and software requirements

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 7.0 and WebSphere Integration Developer version 7.0 are compatible with prior releases as follows:

- Deployment from WebSphere Integration Developer version 6.0.2, 6.1.0, 6.1.2 or 6.2.0 to WebSphere Process Server 7.0 is supported.
  - Applications authored and generated using WebSphere Integration Developer 6.0.2, 6.1.0, 6.1.2 or 6.2.0 can be published to WebSphere Process Server 7.0 servers.
  - Applications authored in, generated in, and exported from WebSphere Integration Developer 6.0.2, 6.1.0, 6.1.2 or 6.2.0 can be installed on WebSphere Process Server 7.0 servers.

  Note: For WebSphere Adapters (versions 6.1.0, 6.1.2 and 6.2.0) you must install the interim fix titled *Mandatory adapter fix for running 6.1 and 6.2 Adapters on WPS v7.0* . You must apply this interim fix on the source environment if you DO NOT plan to update the WebSphere Adapter to a version 7.0 level, and you plan to continue using the application with WebSphere Adapter versions 6.1.0, 6.1.2 or 6.2.0.

  Note: Websphere Adapter for SAP (versions 6.0.2, 6.1.0, 6.1.2 and 6.2.0) is not supported on WebSphere Process Server version 7.0. You must update Websphere Adapter for SAP to version 7.0 before deploying any applications that use Websphere Adapter for SAP on WebSphere Process Server version 7.0. For more information specific to WebSphere Adapter for SAP, see *Postmigration tasks for WebSphere Process Server* .
- Running WebSphere Process Server 7.0 artifacts on WebSphere Process Server 6.0.2, 6.1.0, 6.1.2 or 6.2.0 is *not* supported.
  - Applications authored with WebSphere Integration Developer 7.0 cannot be published to or installed on WebSphere Process Server 6.0.2, 6.1.0, 6.1.2 or 6.2.0 (any prior release) servers. Such content will not run correctly on WebSphere Process Server 6.0.2, 6.1.0, 6.1.2 or 6.2.0, and changes in code generation will cause the applications to not run correctly on WebSphere Process Server 6.0.2, 6.1.0, 6.1.2 or 6.2.0.
  - Applications authored with WebSphere Integration Developer 6.0.2, 6.1.0, 6.1.2 or 6.2.0 and generated in WebSphere Integration Developer 7.0 cannot be published to or installed on WebSphere Process Server 6.0.2, 6.1.0, 6.1.2 or 6.2.0 servers. Changes in code generation will cause the applications to not run correctly on WebSphere Process Server 6.0.2, 6.1.0, 6.1.2 or 6.2.0.
  - Applications generated using serviceDeploy from a WebSphere Process Server 7.0 server cannot be installed on a WebSphere Process Server 6.0.2, 6.1.0, 6.1.2 or 6.2.0 server. Changes in code generation will cause the applications to not run correctly on WebSphere Process Server 6.0.2, 6.1.0, 6.1.2 or 6.2.0.

Related concepts

"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.

Related information

Migrating to WebSphere Process Server

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 9
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 multiple application server versions

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 100
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 in deployment environments" on page 101
Clusters give your applications more capacity and higher availability than a single server.

"Servers" on page 96
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

Messaging engines

Planning to configure Business Process Choreographer

Plan your Business Process Choreographer setup and configuration parameters.

Procedure

1. Perform "Planning the topology, setup, and configuration path."
2. Depending on your chosen configuration path, perform one of the following:
   - For "Basic sample", perform "Planning to create a basic sample Business Process Choreographer configuration" on page 19.
   - For "Sample with organization", perform "Planning to create a sample Business Process Choreographer configuration including a sample organization" on page 19.
   - For "Non-production deployment environment", perform "Planning a non-production deployment environment configuration" on page 20.
   - For "Production deployment environment", perform "Planning to use the administrative console's deployment environment wizard" on page 22.
   - For "Flexible custom configuration", perform "Planning for a custom Business Process Choreographer configuration" on page 26.

Results

You have planned everything you need in order to configure Business Process Choreographer.

Planning the topology, setup, and configuration path

Your choice of topology and setup affects which Business Process Choreographer configuration paths you can use.

About this task

The different configuration paths vary in complexity, flexibility, and their support for different topologies and databases.

Procedure

1. Be aware that you must choose between five different configuration paths.
   - "Basic sample"
   - "Sample with organization"
   - "Non-production deployment environment"
   - "Production deployment environment"
   - "Flexible custom configuration"

   For most configuration paths, you have a choice of configuration tools.
2. Be aware of the different configuration tools that you can use to configure Business Process Choreographer.

   Installer or Profile Management Tool
   Provide the easiest ways to create a non-production system, and they require the least planning.
   - The "Basic sample" configuration includes the following Business Process Choreographer components:
     - Business Process Choreographer
     - Business Process Choreographer Explorer with reporting function
     - A Business Process Choreographer event collector for the reporting function
   - The "Sample with organization" configuration also includes a people directory that is preconfigured with 15 users in a sample organization, and has substitution and group work items enabled.
   - The "Non-production deployment environment" configuration provides an easy way to configure Business Process Choreographer on a cluster, but Business Process Choreographer cannot have its own database, instead, it uses the common WPRCSDB database.

   Administrative console's deployment environment wizard
   Can be used to create a "Production deployment environment" Business Process Choreographer configuration, based on a deployment environment pattern.

   Administrative console's Business Process Choreographer configuration page
   You can use this administrative console page to configure a "Flexible custom configuration" Business Process Choreographer production system on a server or cluster. It provides the opportunity to set many configuration parameters, which need detailed planning. This page does not configure the Business Process Choreographer Explorer, which you can configure using its own configuration page in the administrative console, or by running a script. This configuration path is most suitable for creating production systems.

   bpeconfig.jacl configuration script
   You can use this script to configure a "Flexible custom configuration" Business Process Choreographer production system and all the necessary resources on a given server or cluster. You can run the script interactively, or if you provide all the necessary parameters, it can run in batch mode for repeatable automation. It can create a local database, the necessary messaging resources, and can optionally configure the Business Process Choreographer Explorer, which includes the Business Process Choreographer Explorer reporting function. For some database systems, it can also create a remote database. This configuration path is most suitable for creating production systems.

clientconfig.jacl configuration script

You can only use this script to configure a Business Process Choreographer Explorer, with or without the optional reporting function.

3. Be aware that some of the configuration paths have restrictions that limit their suitability for production systems: For example:
   - After experimenting with one of the sample configurations, it must be removed before you can create a configuration that is suitable for a production system.
   - If you create a configuration that uses a Derby Embedded database, or the common WPRCSDB database, it will not be suitable for a high-performance system. You must remove the configuration before you can create a new configuration that uses a separate high-performance database.
   - If your message store uses either a file store or Derby Embedded data store, you cannot federate the profile into a network deployment environment. To be able to federate the profile, you would have to completely remove your Business Process Choreographer configuration and create a new configuration that uses a remotely accessible database for the message store.
4. If you were familiar with the Business Process Choreographer Observer up to version 6.1.2, be aware that it is now integrated in the Business Process Choreographer Explorer. It is now referred to as the Business Process Choreographer Explorer reporting function, and it can be accessed using the **Reports** tab in the Business Process Choreographer Explorer client. The reporting function uses the same URL as the Business Process Choreographer Explorer .

   When configuring the Business Process Choreographer Explorer in the administrative console, or using the bpeconfig.jacl configuration script or clientconfig.jacl configuration script there is an option to configure the Business Process Choreographer Explorer reporting function.

   If you migrated an existing Business Process Choreographer configuration, any Business Process Choreographer Observer configuration is not migrated. To use the Business Process Choreographer Explorer reporting function you must enable it, as described in Enabling the Business Process Choreographer Explorer reporting function after migration.
5. Identify the main criteria for deciding which configuration path to use. Use the following table to identify choices and constraints:

Table 2. Criteria for selecting a configuration path

Choices			Restrictions		Suitable configuration path
Are you planning a production system?	**What is the deploy-ment target?**	**Type of Business Process Choreographer configuration**	**Can use a separate BPEDB database?**	**Which message stores are supported for the messaging engine**	**Configuration path name, tools, and options**
No	Stand-alone server	Basic sample (without the sample organization)	Yes, but only Derby Embedded	Only Derby Embedded	**"Basic sample"** using one of: • Installer • Profile Management Tool Select the options: • **Stand-alone server profile** • **Typical** • **Enable administrative security**
		Sample including a 15 person organization, and substitution is enabled. This sample is identical to the sample that is available in WebSphere Integration Developer when you include the WebSphere Test Environment.		Derby Embedded, File Store, or WPRCSDB	**"Sample with organization"** using: • Profile Management Tool Select the options: • **Stand-alone server profile** • **Advanced** • **Create server from development template** • **Enable administrative security** • **Configure a sample Business Process Choreographer**
	Cluster	Choice of deployment environment patterns: • Remote Messaging and Remote Support • Remote Messaging • Single Cluster	No, it shares WPRCSDB, which can be any database except Derby Embedded and Microsoft® SQL Server	Shares WPRCSDB, which can be any supported database except File Store and Derby Embedded	**"Non-production deployment environment"** using one of: • Installer • Profile Management Tool Select: **Deployment environment**

Table 2. Criteria for selecting a configuration path (continued)

Choices			Restrictions		Suitable configuration path
Are you planning a production system?	**What is the deploy-ment target?**	**Type of Business Process Choreographer configuration**	**Can use a separate BPEDB database?**	**Which message stores are supported for the messaging engine**	**Configuration path name, tools, and options**
Yes	Cluster	Choice of deployment environment patterns: • Remote Messaging and Remote Support • Remote Messaging • Single Cluster • Custom	Yes, any supported database except Derby Embedded	Any supported database except File Store and Derby Embedded	**"Production deployment environment"** using: • Administrative console Select: **Deployment environment**
		Flexible custom configuration	Yes, any supported database	Any supported database except File Store and Derby Embedded	**"Flexible custom configuration"** using one of: • bpeconfig.jacl script • Administrative console Business Process Choreographer configuration page
	Stand-alone server			Any supported database, or File Store	

Note: It is also possible to use any of the configuration paths that are recommended for creating a production system to create a configuration that is not suitable for a production system.
Consider the following options:

a. Decide whether you are configuring a production system. Typically a production system requires high-performance, scalability, and security. For Business Process Choreographer, a production system should have its own non-Derby BPEDB database.

b. Decide whether the deployment target for the Business Process Choreographer will be a stand-alone server or a cluster.

c. If you do not want to create a production system, decide whether a sample configuration on a stand-alone server will meet your needs. If so, decide whether you want the sample to include a sample people directory (populated with a sample organization) for people assignment and substitution enabled.

   Note: The sample people directory uses the default file registry configured for the federated repositories, and includes all sample people with the same password "wid". The WebSphere administration user ID is also added to the directory, using the password that was specified during profile creation. After the sample configuration has been created, you can use the administrative console to view which users and groups are available by clicking **Users and Groups**, then either **Manage Users** or **Manage Groups**.

d. If you want to configure Business Process Choreographer on a cluster, depending on your performance requirements, decide whether the messaging engines and supporting applications, (such as the Business

Process Choreographer Explorer and Common Event Infrastructure) will have their own cluster, or share one. The standard deployment environment patterns are:

Remote Messaging and Remote Support
Three clusters are used. One each for the applications, messaging engines, and support applications.

Remote Messaging
One cluster is used for the applications and support functions. A second cluster is used for the messaging engines.

Single cluster
Only one cluster is used for applications, messaging engines, and support applications.

Custom
More flexible setup.

e. Decide whether you want a dedicated BPEDB database for Business Process Choreographer.

f. Business Process Choreographer will use the same type of messages store that is used by SCA:
   - If SCA uses a FILESTORE, then Business Process Choreographer will also use a FILESTORE.
   - If SCA uses a Derby Embedded database, then Business Process Choreographer will use its own Derby Embedded database.
   - If SCA uses any other database, then Business Process Choreographer will use its own schema in the same database.

6. If you want to use the Business Process Choreographer Explorer reporting function, which is integrated in the Business Process Choreographer Explorer, you can either configure it at the same time as you create a Business Process Choreographer configuration, or you can create it later. Decide whether Business Process Choreographer Explorer reporting function will also use the BPEDB database, or whether it will have its own, OBSRVRDB, database. Also plan the topology for the Business Process Choreographer Explorer reporting function components. To perform the detailed planning now, perform "Planning for the Business Process Choreographer Explorer reporting function" on page 48.
7. If you want WebSphere Portal Server or any custom WebSphere Process Server client to access Business Process Choreographer, perform "Planning for a remote client application" on page 51.
8. If you have application security enabled and you have a long-running process that calls a remote EJB method, Common Secure Interoperability Version 2 (CSIv2) identity assertion must be enabled when you configure CSIv2 inbound authentication.
9. If you will use human tasks, then WebSphere administrative security and application security must both be enabled.

Results

You have planned the topology and know which configuration path and configuration tool you will use.

Planning to create a basic sample Business Process Choreographer configuration

This basic sample, for a stand-alone server, does not include a sample organization.

Before you begin

You have performed "Planning the topology, setup, and configuration path" on page 13, and have selected the "Basic sample" configuration path.

Procedure

1. Decide whether you will create the sample using the Installer or Profile Management Tool.
2. If you decided to use the Profile Management Tool, decide whether the Business Process Choreographer messaging engine will use file store, an embedded Derby database, or the common, WPRCSDB, database.
3. If you want the Human Task Manager to be able to send escalation e-mails, plan the following:
   - If there will not be a local Simple Mail Transfer Protocol (SMTP) mail server available, plan to change the mail session later to point to a suitable mail server.
   - Plan to change the sender address for the e-mails. Otherwise, it will use a dummy sender address.
4. Be aware that this sample configuration uses the WebSphere administrator user ID and password for the various Business Process Choreographer user IDs.

Results

You have planned to create a basic sample Business Process Choreographer configuration.

Planning to create a sample Business Process Choreographer configuration including a sample organization

This sample includes a 15 person sample organization, which is suitable for experimenting with people assignment and substitution on a stand-alone server. This sample is identical to the sample that is available in WebSphere Integration Developer when you include the WebSphere Test Environment.

Before you begin

You have performed "Planning the topology, setup, and configuration path" on page 13, and have selected the "Sample with organization" configuration path.

About this task

This sample Business Process Choreographer configuration requires minimal planning.

Procedure

1. Decide whether the Business Process Choreographer messaging engine will use file store, an embedded Derby database, or the common, WPRCSDB, database.
2. Be aware that this sample can only be created using the Profile Management Tool. To get this sample, you must select the following options:

- **Stand-alone server profile**
- **Advanced**
- **Create server from development template**
- **Enable administrative security**
- **Configure a sample Business Process Choreographer**

If for example, you do not enable administrative security, the sample Business Process Choreographer configuration will not be created.

Note: The sample people directory uses the default file registry configured for the federated repositories, and includes all sample people with the same password "wid". The WebSphere administration user ID is also added to the directory, using the password that was specified during profile creation. After the sample configuration has been created, you can use the administrative console to view which users and groups are available by clicking **Users and Groups**, then either **Manage Users** or **Manage Groups**.

3. If you want the Human Task Manager to be able to send escalation e-mails, plan the following:
   - If there will not be a local Simple Mail Transfer Protocol (SMTP) mail server available, plan to change the mail session later to point to a suitable mail server.
   - Plan to change the sender address for the e-mails. Otherwise, it will use a dummy sender address.
4. Be aware that this sample configuration uses the WebSphere administrator user ID and password for the various Business Process Choreographer user IDs.

Results

You have planned to create a sample Business Process Choreographer configuration including a sample organization.

Planning a non-production deployment environment configuration

Planning to use the Installer or Profile Management Tool to create a Business Process Choreographer configuration that is based on a deployment environment pattern.

Before you begin

You have performed "Planning the topology, setup, and configuration path" on page 13, and have selected the "Non-production deployment environment" configuration path.

About this task

When using the deployment environment wizard, you must select the deployment environment pattern, then you have the opportunity to change the default database parameters and authentication aliases for the WBI_BPC component, and enter other parameters for Business Process Choreographer.

Procedure

1. Decide which deployment environment pattern you will use:
   - **Remote Messaging and Remote Support**
   - **Remote Messaging**

- **Single Cluster**

2. Plan the user name for the Business Process Choreographer JMS authentication alias that you will enter during the Security step.
3. Plan the **Business Process Choreographer Explorer context root**, which defines part of the URL that browsers must use to reach the Business Process Choreographer Explorer.
4. Plan the security parameters for the Business Process Choreographer step. These user IDs and groups will be used for the Business Flow Manager and Human Task Manager:

   Administrator User and Group
   Plan a list of user IDs or a list or groups, or both, onto which the business administrator role is mapped.

   Monitor User and Group
   Plan a list of user IDs or a list or groups, or both, onto which the business monitor role is mapped.

   JMS API Authentication User and Password
   The run-as user ID for the Business Flow Manager message driven bean.

   Escalation User Authentication User and Password
   The run-as user ID for the Human Task Manager message driven bean.

   Cleanup User Authentication User and Password
   The run-as user ID for the Business Flow Manager and Human Task Manager cleanup service. This user must be in the business administrator role.

5. If you want to configure an e-mail session for the Human Task Manager escalations, plan the following parameters for the Business Process Choreographer step:

   Mail transport host
   The host name or IP address where the Simple Mail Transfer Protocol (SMTP) e-mail service is located.

   Mail transport user and Mail transport password
   If the mail server does not require authentication, you can leave these fields empty.

   Business Process Choreographer Explorer URL
   This URL is used to provide a link in generated e-mails so that a business administrator who receives an e-mail notification can click on the link to view the related business process or human task in their Web browser.

6. If you are going to use the Business Process Choreographer Explorer, the Business Space, or a client that uses the Representational State Transfer (REST) API or the JAX Web Services API, decide on the context roots for the REST API and the JAX Web Services API.
   - The defaults for the Business Flow Manager are `/rest/bpm/bfm` and `/BFMJAXWSAPI`.
   - The defaults for the Human Task Manager are `/rest/bpm/htm` and `/HTMJAXWSAPI`.
   - When configured on a server, or on a single cluster, or on multiple clusters that are mapped to different Web servers, you can use the default values.
   - When configured in a network deployment environment on multiple deployment targets that are mapped to the same Web server, do not use the

default values. The context root for each Business Process Choreographer configuration must be unique for each combination of host name and port. You will have to set these values manually using the administrative console after configuring Business Process Choreographer.

7. If you want to use people assignment, perform "Planning for the people directory provider" on page 45.

Results

You have planned to create a non-production deployment environment configuration.

Planning to use the administrative console's deployment environment wizard

For a production system plan all the configuration parameters for Business Process Choreographer, including a separate database. For a non-production system you can use a shared database.

Before you begin

You have performed "Planning the topology, setup, and configuration path" on page 13, and have selected the "Production deployment environment" configuration path.

About this task

When using the deployment environment wizard, you must select the deployment environment pattern, then you have the opportunity to change the default database parameters and authentication aliases for the WBI_BPC component, and enter other parameters for Business Process Choreographer.

Procedure

1. If you do not have enough information or authority to create the whole configuration on your own, consult and plan with the people who are responsible for other parts of the system. For example:
   - You might need to request information about your organization's LDAP server, if it uses authentication you will need to request a user ID, and authorization.
   - If you are not authorized to create the database, your database administrator (DBA) must be included in planning the databases. Your DBA will need a copy of the database scripts to customize and run.
2. Perform Planning security, user IDs, and authorizations.
3. Decide which deployment environment pattern you will use:
   - **Remote Messaging and Remote Support**
   - **Remote Messaging**
   - **Single Cluster**
   - **Custom**
4. If you chose the **Custom** deployment environment pattern:
   a. Decide if you want to install the Business Process Choreographer Explorer. If so, plan the following:
      - Where you will deploy it.

- If you want to use the Business Process Choreographer Explorer reporting function, also plan where you will deploy the Business Process Choreographer event collector.

b. Plan the context root for the SCA bindings.

c. Plan whether you want to enable or disable the state observers and audit logging.

5. If you are planning to have dedicated databases for the following:
   - The BPEDB database for Business Process Choreographer, which can be changed in the wizard in a table row for the component WBI_BPC.
   - The BPEME database for the Business Process Choreographer messaging engine, which can be changed in the wizard in a table row for the component WBI_BPC_ME.
   - The OBSRVRDB database for the Business Process Choreographer Explorer reporting function, which can be changed in the wizard in a table row for the component WBI_BPCEventCollector.

   Plan the following parameters for each database, to enter on the wizard's database page:

   Database name
   The name of the database, for example, `BPEDB`, `BPEME`, or `OBSRVRDB` instead of the default value, `WPRCSDB`, which results in sharing the common database. The default value is only suitable for lower performance setups.

   Schema
   The schema qualifier to be used for each database.

   Create Tables
   If selected, the tables will be created automatically the first time that the database is accessed. For this option to work, the database must already exist, and the user name provided for creating the data source must have the authority to create tables and indexes in the database. If not selected, the tables will not be created automatically, and you must create the tables manually by running scripts. For a production system, clear this option, and plan to use the provided SQL scripts to setup the database.

   User Name and Password
   A user ID that has the authority to connect to the database and to modify the data. If the user ID has the authority to create tables and indexes in the database, then the option to create the tables automatically can be used, and when necessary, the database schema will be updated automatically after applying a service or fix pack.

   Server The address of the database server. Specify either the host name or the IP address.

   Provider
   The JDBC provider.

   Also plan the database-specific settings, which you can set using the **Edit** button for the JDBC provider.

Table 3. Database-specific settings

Database / JDBC driver type	Database-specific settings
DB2® UDB – Universal driver	• User name • Password • Database name • Schema name • Server name • Server port number • Driver type • Description • Create tables
DB2 for i5/OS® – Toolbox driver	• User name • Password • Database name • Collection name • Server name • Description • Create tables
DB2 for z/OS® V8 and V9	• Implementation type – Connection pool data source or XA data source • User name • Password • Database name • Schema name • Server name • Server port number • Storage group • Description
Derby Network Server or Derby Network Server 40	• User name • Password • Description • Create tables • Server name • Server port number
Derby Embedded or Derby Embedded 40	• Description • Create tables
Microsoft SQL Server – Datadirect, and Microsoft drivers	• User name • Password • Database name • Server name • Server port number • Description • Create tables

Table 3. Database-specific settings (continued)

Database / JDBC driver type	Database-specific settings
Informix® Dynamic Server – Universal and DataServer drivers	• User name • Password • Server name • Server port number • Description • Create tables
Oracle – oci driver	• User name • Password • Database name • Schema name • Driver type – oci • Description • Create tables
Oracle – thin driver	• User name • Password • Database name • Schema name • Server name • Server port number • Driver type – thin • Description • Create tables

For more details about planning the databases, see "Planning the databases for Business Process Choreographer" on page 33.

6. Plan the user name for the Business Process Choreographer JMS authentication alias that you will enter during the Security step.
7. Plan the **Business Process Choreographer Explorer context root**, which defines part of the URL that browsers must use to reach the Business Process Choreographer Explorer.
8. Plan the security parameters for the Business Process Choreographer step. These user IDs and groups will be used for the Business Flow Manager and Human Task Manager:

   Administrator User and Group
   Plan a list of user IDs or a list or groups, or both, onto which the business administrator role is mapped.

   Monitor User and Group
   Plan a list of user IDs or a list or groups, or both, onto which the business monitor role is mapped.

   JMS API Authentication User and Password
   The run-as user ID for the Business Flow Manager message driven bean.

   Escalation User Authentication User and Password
   The run-as user ID for the Human Task Manager message driven bean.

Cleanup User Authentication User and Password
The run-as user ID for the Business Flow Manager and Human Task Manager cleanup service. This user must be in the business administrator role.

9. If you want to configure an e-mail session for the Human Task Manager escalations, plan the following parameters for the Business Process Choreographer step:

   Mail transport host
   The host name or IP address where the Simple Mail Transfer Protocol (SMTP) e-mail service is located.

   Mail transport user and Mail transport password
   If the mail server does not require authentication, you can leave these fields empty.

   Business Process Choreographer Explorer URL
   This URL is used to provide a link in generated e-mails so that a business administrator who receives an e-mail notification can click on the link to view the related business process or human task in their Web browser.

10. If you are going to use the Business Process Choreographer Explorer, the Business Space, or a client that uses the Representational State Transfer (REST) API or the JAX Web Services API, decide on the context roots for the REST API and the JAX Web Services API.
    - The defaults for the Business Flow Manager are `/rest/bpm/bfm` and `/BFMJAXWSAPI`.
    - The defaults for the Human Task Manager are `/rest/bpm/htm` and `/HTMJAXWSAPI`.
    - When configured on a server, or on a single cluster, or on multiple clusters that are mapped to different Web servers, you can use the default values.
    - When configured in a network deployment environment on multiple deployment targets that are mapped to the same Web server, do not use the default values. The context root for each Business Process Choreographer configuration must be unique for each combination of host name and port. You will have to set these values manually using the administrative console after configuring Business Process Choreographer.
11. If you want to use people assignment, perform "Planning for the people directory provider" on page 45.

Results

You have planned to use the administrative console's deployment environment wizard.

Planning for a custom Business Process Choreographer configuration

Plan the configuration parameters and options for creating a custom configuration, using either the Administrative console's Business Process Choreographer configuration page or the bpeconfig.jacl configuration script.

Before you begin

You have performed "Planning the topology, setup, and configuration path" on page 13, and have selected the "Flexible custom configuration" configuration path.

Procedure

1. Know which of the following you will use to configure Business Process Choreographer:
   - Administrative console's Business Process Choreographer configuration page
   - The bpeconfig.jacl configuration script
2. If you do not have enough information or authority to create the whole configuration on your own, consult and plan with the people who are responsible for other parts of the system. For example:
   - You might need to request information about your organization's LDAP server, if it uses authentication you will need to request a user ID, and authorization.
   - If you are not authorized to create the database, your database administrator (DBA) must be included in planning the databases. Your DBA will need a copy of the database scripts to customize and run.
3. Planning security, user IDs, and authorizations
4. "Planning the databases for Business Process Choreographer" on page 33
5. "Planning for the Business Flow Manager and Human Task Manager" on page 44
6. "Planning for the people directory provider" on page 45
7. "Planning for the Business Process Choreographer Explorer" on page 47
8. If you will use the Administrative console's Business Process Choreographer configuration page, make sure that you have planned all the values that you will enter on the configuration page.
9. If you will use the bpeconfig.jacl configuration script:
   a. Make sure that you have planned all the options and parameter values that you must specify on the command-line, or in a batch file. The options and parameters are summarized in Using the bpeconfig.jacl script to configure Business Process Choreographer, and are described in detail in bpeconfig.jacl script.
   b. If you will use a batch file to run the bpeconfig.jacl configuration script, create the batch file or shell script.

Results

You have planned everything you need to be able to create a custom Business Process Choreographer configuration.

What to do next

Perform Configuring Business Process Choreographer.

Planning security, user IDs, and authorizations

Plan the user IDs and authorizations for configuring Business Process Choreographer.

About this task

During configuration, you need to use various user IDs and you must specify other user IDs that will be used at runtime. Make sure that you plan and create all user IDs before you start configuring Business Process Choreographer.

For a sample Business Process Choreographer configuration:

You only need the authority to create a new profile. In the Profile

Management Tool, using the option to create a typical profile, when you enable administrative security, the Business Process Choreographer sample will also be configured. No other planning or user IDs are required, and you can skip this task.

For a high security configuration:
You must plan all user IDs in detail as described in this task.

For a low security configuration:
If you do not require full security, for example for a non-production system, you can reduce the number of user IDs that are used. You must plan all user IDs in detail, but you can use certain user IDs for multiple purposes. For example, the database user ID used to create the database schema can also be used as the data source user name to connect to the database at runtime.

If you will use the bpeconfig.jacl script to configure Business Process Choreographer:

The user ID used to run the bpeconfig.jacl script must have the necessary rights for the configuration actions that the script will perform. Otherwise, you must specify user IDs as parameters for the script that have the necessary rights, in which case you must plan all user IDs in detail. For user IDs that can be specified as parameters to the bpeconfig.jacl script, the parameter names are included in the table.The profile must already exist. If WebSphere administrative security is enabled, you need a WebSphere administrator user ID in the `configurator` role that you can use to invoke the wsadmin tool.

If you will use human tasks:
WebSphere administrative security and application security must both be enabled.

Procedure

1. Print a hardcopy of this page so that you can write your planned values in the last column. Keep it for reference when you are configuring Business Process Choreographer, and keep it in your records for future reference.
2. Plan the user ID you will use on the WebSphere Process Server to configure Business Process Choreographer.

Table 4. Planning user IDs for WebSphere Process Server

User ID or role	When the user ID is used	What the user ID is used for	Which rights the user ID must have	Planned user ID
The user who configures Business Process Choreographer	Configuring	Logging onto the administrative console and running administrative scripts.	WebSphere administrator or configurator role, if WebSphere administrative security is enabled.	
		If you are going to run the bpeconfig.jacl script to configure the Business Process Choreographer.	When running the script, you must also provide any user IDs that are necessary for the options that you select. For more information see bpeconfig.jacl script.	

3. Plan which people need access to subdirectories of *install_root*. If your security policy does not allow these people to be granted this access, they will need to be given copies of the files in the directories.

Table 5. Planning access to the subdirectories of install_root

User ID or role	When the user ID is used	What the user ID is used for	Which rights the user ID must have	Planned user ID
Database administrator	Configuring	Running the scripts to setup the following databases: **BPEDB**: This is the default name for the database for Business Process Choreographer. **OBSRVDB**: This is the default name for the database for the Business Process Choreographer Explorer reporting function.	If you use the bpeconfig.jacl script to configure Business Process Choreographer: Read access to (or a copy of) the createSchema.sql script that bpeconfig.jacl generates in a subdirectory of the directory: *profile_root*/dbscripts/ProcessChoreographer/	
			If you want to review the database script files: Read access to (or a copy of the files in) the database scripts provided in the directory: *install_root*/dbscripts/ProcessChoreographer/*database_type*Where *database_type* is one of the following: • DB2zOSV8 • DB2zOSV9	
Integration developer	Customizing	To use people assignment with a Lightweight Directory Access Protocol (LDAP) or Virtual Member Manager (VMM) people directory provider, you will have to customize a copy of the sample XSL transformation file.	Either read access to the Staff directory, or a copy of the files in the directory: *install_root*/ProcessChoreographer/StaffThe integration developer will also need write access to a suitable directory to make the customized XSL transformation file available to the server.	

4. Plan the user IDs that will be used to create, configure, and access the database that is used by Business Process Choreographer.

Table 6. Planning user IDs for the BPEDB database

User ID or role	When the user ID is used	What the user ID is used for	Which rights the user ID must have	Planned user ID
Database administrator	Before configuring	To create the BPEDB database. For Oracle: To create the BPEDB database.	Create the database.	
Database administrator or an administrator who will run the bpeconfig.jacl script	Configuring	You or your database administrator must run Business Process Choreographer database scripts, unless you are using the embedded Derby database.	For the BPEDB database: Alter tables, connect, insert tables, and create indexes, schemas, tables, table spaces, and views.	

Table 6. Planning user IDs for the BPEDB database (continued)

User ID or role	When the user ID is used	What the user ID is used for	Which rights the user ID must have	Planned user ID
Data source user name If you use the bpeconfig.jacl script, this is the -dbUser parameter.	Configuring	If you select the **Create Tables** option, this user ID is used to create the database tables.	To use the **Create Tables** configuration option, this user ID must also be authorized to perform the following actions on the BPEDB database: Alter tables, connect, insert tables, and create indexes, tables, and views.	
	Runtime	The Business Flow Manager and Human Task Manager use this user ID to connect to the BPEDB database.	This user ID must be authorized to perform the following actions on the BPEDB database: Connect, delete tables, insert tables, select tables and views, and update tables.	
	After applying service or a fix pack	When necessary, the database schema is updated automatically after applying service. This only works if this user ID has the necessary database rights, otherwise schema updates must be performed manually.	This user ID must be authorized to perform the following actions on the BPEDB database: Alter, create, insert and select tables, connect to the database, create and drop indexes and views.	

5. If you will configure the Business Process Choreographer Explorer reporting function, plan the user IDs to use to create, configure, and access the reporting database.

Table 7. Planning user IDs for the reporting database

User ID or role	When the user ID is used	What the user ID is used for	Which rights the user ID must have	Planned user ID
Database administrator	Before configuring	To create the reporting database. For Oracle, to create the reporting database.	Create the database.	
Database administrator or an administrator	Configuring	Running the setupEventCollector tool, or SQL scripts to create the schema.	For the reporting database: Alter tables, connect, create procedure, insert tables, and create tables, table spaces, and views. If you are going to use the Java implementation of the user-defined functions, the user ID must also be authorized to install the JAR file.	
Event collector data source user name	Runtime	Connecting to the reporting database. If you are using the reporting database and it uses the BPEDB database, use the same user name as for the Business Process Choreographer data source.	Connect to the database.	

6. If you will have a separate database for the Business Process Choreographer's messaging engine message store (not Derby Embedded nor file store), plan the user ID that will be used to access the database.

Table 8. Planning user ID for the preconfigured BPEME messaging engine database

User ID	When the user ID is used	What the user ID is used for	Which rights the user ID must have	Planned user ID
Bus data source user name If you use the bpeconfig.jacl script, this is the -medbUser parameter.	Configuring and runtime	This user name is used to connect to the BPEME database, and to create the necessary tables and index.	This user ID must be authorized to perform the following actions on the BPEME database: Connect, delete tables, insert tables, select tables and views, and update tables.	

7. Plan the Business Process Choreographer user IDs for the Java Message Service (JMS).

Table 9. Planning user IDs for JMS

User ID	When the user ID is used	What the user ID is used for	Which rights the user ID must have	Planned user ID
JMS authentication user	Runtime	The authentication alias for the system integration bus. You must specify it when configuring Business Process Choreographer. If you use the bpeconfig.jacl script, this user IDs and its password are the parameters -mqUser and -mqPwd.	It must be a user name that exists in the WebSphere user registry. It is automatically added to the Bus Connector role for the Business Process Choreographer bus.	
JMS API authentication user	Runtime	Any Business Flow Manager JMS API requests will be processed on using this user ID. If you use the bpeconfig.jacl script, this user IDs and its password are the parameters -jmsBFMRunAsUser and -jmsBFMRunAsPwd.	The user name must exist in the WebSphere user registry.	
Escalation authentication user	Runtime	Any Human Task Manager escalations will be processed using this user ID. If you use the bpeconfig.jacl script, this user ID and its password are the parameters -jmsHTMRunAsUser and -jmsHTMRunAsPwd.	The user name must exist in the WebSphere user registry.	

8. Plan which groups or user IDs, the Java EE roles for the Business Flow Manager and Human Task Manager will be mapped onto.

Table 10. Planning the security roles for the Business Flow Manager and Human Task Manager

User ID or role	When the user ID is used	What the user ID is used for	Planned list of user IDs, groups, or both
Administrator user	Runtime	The system administrator and monitor security roles for both the Business Flow Manager and Human Task Manager are each mapped to a list of user IDs, groups, or both. The values defined here create the mapping that gives users in this role the access rights that they need. If you use the bpeconfig.jacl script, these users and groups correspond to the following parameters: • -adminUsers • -adminGroups • -monitorUsers • -monitorGroups	
Administrator group	Runtime		
Monitor user	Runtime		
Monitor group	Runtime		

9. Plan the user ID to use as the Java EE run-as role for administration jobs like the Business Flow Manager and Human Task Manager cleanup services and the process instance migration tool. This user ID must be a member of the administrator role user or group planned in Table 10.

Table 11. Planning the user ID for running administration jobs

User ID	When the user ID is used	What the user ID is used for	Planned user ID
Administration job user ID	Runtime administration	This user ID is used to run administration jobs. If you use the bpeconfig.jacl script, this user ID and its password correspond to the -adminJobUser and -adminJobPwd parameters.	

10. If you want human task escalations to send notification e-mails for specific business events, and your Simple Mail Transfer Protocol (SMTP) server requires authentication, decide which user ID will be used to connect to the e-mail server.

Table 12. Planning the user ID for the e-mail server

User ID or role	When the user ID is used	What the user ID is used for	Which rights the user ID must have	Planned user ID
Mail transport user	Runtime	The Human Task Manager uses this user ID to authenticate against the configured mail server to send escalation e-mails. If you use the bpeconfig.jacl script, this is the -mailUser parameter. The password is the -mailPwd parameter.	Send e-mails.	

11. If you will use people assignment for human tasks, and you will use a Lightweight Directory Access Protocol (LDAP) people directory provider that uses simple authentication, plan a Java Authentication and Authorization Service (JAAS) alias and an associated user ID that will be used to connect to the LDAP server. If the LDAP server uses anonymous authentication this alias and user ID are not required.

Table 13. Planning the alias and user ID for the LDAP server

User ID or role	When the user ID is used	What the alias and user ID are used for	Which rights the user ID must have	Planned alias and user ID
LDAP plug-in property: Authentication Alias	Runtime	The alias is used to retrieve the user ID that is used to connect to the LDAP server. You specify this alias ID when customizing the properties for the LDAP plug-in, for example `mycomputer/My LDAP Alias`.	The JAAS alias must be associated with the LDAP user ID.	
LDAP user ID	Runtime	This user ID is used to connect to the LDAP server.	If the LDAP server uses simple authentication, this user ID must be able to connect to the LDAP server. This user ID is either a short name or a distinguished name (DN). If the LDAP server requires a DN you cannot use the short name.	

12. Create the user IDs that you have planned with the necessary authorizations. If you do not have the authority to create them all yourself, submit a request to the appropriate administrators, and enter the names of the user IDs that they create for you in this table.

Results

You know which user IDs will be required when configuring Business Process Choreographer.

Planning the databases for Business Process Choreographer

Plan the databases for Business Process Choreographer. Depending on your setup, you might need to plan to create up to three databases, or none.

About this task

Business Process Choreographer can share a database with other process server components or other Business Process Choreographer configurations. The BPEDB database is used by the Business Flow Manager and the Human Task Manager. For a production system plan to have a dedicated database for each deployment target where Business Process Choreographer is configured.

If you have multiple Business Process Choreographer configurations, then each of these needs its own database or database schema. The Business Process Choreographer database tables cannot be shared between multiple Business Process Choreographer configurations.

Restriction: If you are using Informix, you cannot have multiple Business Process Choreographer configurations share the same database.

If you use the Business Process Choreographer Explorer reporting function, which until Version 6.1.2 was known as the Business Process Choreographer Observer, it can use the same BPEDB database, but using an additional database, gives better performance. Some of the scripts for setting up the reporting database already contain the suggested name OBSRVDB, though you are free to choose a different name.

The Business Process Choreographer messaging engines can either share the database used by the SCA messaging engines, or have their own BPEMEDB database. For more information about which databases are supported for your selected configuration path, see Table 2 on page 16.

Procedure

1. For a production system:
   a. If performance is important, plan to use a separate database for Business Process Choreographer, as described in "Planning the BPEDB database" on page 35, otherwise, plan to use the WPRCSDB common database.
   b. If you will use the Business Process Choreographer Explorer reporting function:
      - If you want to minimize the impact that its queries have on the performance of your business processes, plan to use a separate database (OBSRVDB) as described in "Planning the reporting database" on page 39.
      - Otherwise, plan to configure it to use the BPEDB database.
   c. For high-load setups, for example, a large cluster with very high messaging rates, consider improving the performance by using a separate database for the Business Process Choreographer messaging engine. This allows the database logging to be parallelized, which can help to prevent it from becoming a bottleneck.
      - If you use the administrative console to configure Business Process Choreographer, and you want a separate database for the Business Process Choreographer messaging engine, perform "Planning the messaging engine database" on page 43, otherwise plan to use the default database that is used by the Service Component Architecture (SCA).
      - If you use the bpeconfig.jacl configuration script to configure Business Process Choreographer, Business Process Choreographer will use the same type of messages store that is used by SCA.
   d. Optional: Use the database design tool to interactively create the database design file and the SQL script files that the database administrator can use to create all three databases that you planned in the previous steps. There are significant advantages to using this tool:
      - You can run the tool as often as necessary to refine the database design parameters, without the risk of breaking them, rather than editing the provided template SQL files manually.
      - If you have used a database design file, the next time that you migrate to a newer version of WebSphere Process Server, you can generate the schema update SQL scripts.
      - If you create a database design file for a test configuration, it is convenient to be able to make of copy of the design file and make minor changes to it for the databases for your production system.
      - Using the tool, you can also define the data sources for all three databases. Though you must configure the data source for the reporting database manually.

      Important: When using the database design tool to create a deployment environment, after you have configured the common database, Business Process Choreographer is shown as being "complete". This is because there is a valid default, that causes the tables for Business Process Choreographer to be created in the common database. However, this default is not suitable

for production systems. For a production system, make sure that you configure a dedicated database for each deployment target where Business Process Choreographer is configured.

2. For a non-production system where simplicity of setup is more important than performance, your options depend on the configuration path that you have chosen:
   - If you will use the Installer or Profile Management Tool to create the "Basic sample" or the "Sample with organization" Business Process Choreographer configuration, a separate Derby Embedded BPEDB database is created, which is also used by the Business Process Choreographer Explorer reporting function. For the Business Process Choreographer messaging engine, the default is to have a separate Derby Embedded database (BPEME). If you use the Profile Management Tool, you can also select to use a **File Store** or to share the WPRCSDB database.
   - If you will use the Installer or Profile Management Tool to create a deployment environment that includes a Business Process Choreographer configuration, Business Process Choreographer, Business Process Choreographer Explorer reporting function, and the Business Process Choreographer messaging engine will all use the WPRCSDB database. Therefore, you do not need to do any database planning for Business Process Choreographer.

Results

You have planned all the databases for your Business Process Choreographer configuration.

Planning the BPEDB database:

Plan the database for Business Process Choreographer.

About this task

Business Process Choreographer requires a database. SQL scripts are provided for all supported database systems to create and administer the database schema. When a database is in place, JDBC access to the database has to be configured for Business Process Choreographer. Depending on the database system, your topology, the purpose of the installation, and the administrative tool you choose to use, some or all of the tasks to create the database and to configure JDBC access can be automated. For a productions system, Business Process Choreographer should have its own database, but if performance is not important, you can also configure Business Process Choreographer to share a database with other WebSphere Process Server components.

Procedure

1. Make sure that your choice of BPEDB database and configuration path are compatible: The following databases are supported:
   - Derby

     Note: Derby serializes database access. Activities are therefore always performed sequentially, even in flows that are modeled to support the parallel execution of activities.
   - DB2 for z/OS

   If you have already decided how you are going to configure Business Process Choreographer, your choice of configuration path has implications on how

you can create the database. If you have not yet decided which configuration path to use to configure Business Process Choreographer, identifying your database requirements will help eliminate the configuration paths that do not support your needs. For details about which databases are supported by each configuration path, see Table 2 on page 16.

2. Business Process Choreographer will use a Derby database, or if you use a response file, DB2 for z/OS.
3. If you do not need the performance, scalability, and security that is normally required for a production system, you can have the database objects created in a single table space on a database server that is local to the WebSphere Process Server. This minimizes the planning and effort necessary to create the database, but requires that the user ID used to access the database also has database administration rights. The options that you need to plan depend on the configuration path that you choose:
   a. If you use the **Installer** or **Profile Management Tool** to get a sample Business Process Choreographer configuration, a separate Derby BPEDB database is created for Business Process Choreographer, which requires no further planning.
   b. If you use the administrative console **Deployment Environment wizard** to configure Business Process Choreographer, and it is sufficient for the default schema to be created in a single table space, plan to use a copy of the provided SQL script to create the BPEDB database.
   c. If you use the **bpeconfig.jacl** tool to configure Business Process Choreographer, plan which of the following apply in your case.
      - If you will run the bpeconfig.jacl script in interactive mode, you can select to create the tables in an existing database.
      - If you have a user ID with the authority to create the database objects, you can use the -createDB yes option, which causes the bpeconfig.jacl script to generate and run an SQL file to create the database objects in the default table space. In this case also plan to stop the server and use the `-conntype NONE` option for the wsadmin utility.
        – If you are using Derby database, bepconfig.jacl will create the database instance. If you are using a DB2 for z/OS database, the database instance must already exist.
        – If any error occurs while creating the database or objects, you can use the generated SQL scripts as if you used the -createDB no option.
      - If you do not have a user ID with the authority to create the database objects, you must use the -createDB no option, which causes the bpeconfig.jacl script to generate an SQL file to create the database objects in the default table space, but it does not run the script. In this case, plan to ask your database administrator to customize and run the script for you.

      For more information about the tool and other database parameters, see bpeconfig.jacl script.
   d. If you use the administrative console's **Business Process Choreographer configuration page**:
      - To have the Business Process Choreographer database objects created in the common database WPRCSDB, plan to use the default database as the target for the Business Process Choreographer data source.
      - To reuse an existing database, plan to specify the existing database as the target for the Business Process Choreographer data source.

- If you select the `Create tables` option, Business Process Choreographer will create the database objects that it needs in the default table space, the first time that it uses the database. This option cannot be used for a DB2 on z/OS database nor for a remote Oracle database. To use this option for a DB2 UDB database, the database must have `AUTOMATIC STORAGE YES` enabled.
- To create the database using scripts, plan not to use the `Create tables` option.

e. Skip to step 14 on page 39.

4. Perform all of the following steps if you want a **high performance** database setup for Business Process Choreographer with the following characteristics:
   - The database is only used by Business Process Choreographer.
   - Ideally, the database is on a dedicated server, however it can also be local to the WebSphere Process Server system.
   - You can customize the allocation of tables space to disks for better performance.
   - You can use a different user ID to access the database to the one used to administer it.
5. If you have not already planned the user IDs for the database, perform Table 6 on page 29.
6. Plan the allocation of disks and table spaces. Ideally, the database host should have a fast storage subsystem, such as network-attached storage or a storage area network. For a production system, take into account the results of your experiences during development and system testing. The size of your database depends on many factors. Processes that run as microflows use very little space, yet each process template can require tens or hundreds of kilobytes.

   If you will use individual disks, and your database system supports allocating database tables to different disks, plan how many disks you will use and how you will allocate them. Hardware-assisted disk arrays usually offer better performance than single disks.

   For DB2 for z/OS, a table space is created for each table, and additional larger object (LOB) table spaces for LOB columns. They can be all on one high-performance RAID-array, but each table space should be in a different file to allow parallel access. Keep in mind that for a given number of disks, using a RAID configuration will have better performance than allocating table spaces to separate disks. For example, for a DB2 database that is running on a dedicated server with N processors, consider using the following guidelines:
   - For the table spaces, use a RAID-1 array with 2*N primary disks, 2*N mirror disks, and a stripe size of 256 kb.
   - For the database transaction log, use a RAID-1 array with 1.5*N primary disks, 1.5*N mirror disks, and a stripe size of 64 kb.
7. Plan that you or your database administrator will customize the SQL scripts that create the database objects before running them.
   - If you use the **bpeconfig.jacl** tool to configure Business Process Choreographer, use the -createDB no option. This prevents the tool from running the SQL script that it generates. The generated SQL files are based on the original SQL files that are provided for your database, but with all configuration parameters that are provided to the bpeconfig.jacl tool pre-filled in the SQL file, which minimizes the customization necessary.
   - If you use the administrative console's **Business Process Choreographer configuration page** or **Deployment Environment wizard** to configure Business Process Choreographer, plan to clear the `Create tables` option, to

make sure that you do not get the default schema. The generated SQL files are based on the original SQL files that are provided for your database, but all configuration parameters that you enter in the administrative console are pre-filled in the generated SQL file, which minimizes the customization necessary.

For more information about using the generated SQL scripts, refer to Using a generated SQL script to create the database schema for Business Process Choreographer. If you want to preview the original SQL files for your database, so that you can plan what customizations you will make, locate and view the SQL `createSchema.sql` script for your database, but do not modify it. The original SQL files are located in the following directory:

install_root`/dbscripts/ProcessChoreographer/`*database_type*Where *database_type* is one of the following:

- DB2zOSV8
- DB2zOSV9

8. If the database server is remote to the WebSphere Process Server system, plan to install either a Java Database Connectivity (JDBC) driver or a database client on the WebSphere Process Server system:
   - For a Type 2 JDBC driver: Decide which database client to install, and where to install it.
   - For a Type 4 JDBC driver: Locate the JAR file for the driver, which is provided as part of your product installation, and decide where to install it.
9. If the database server is local to the process server, the JDBC JAR files required to access the database are installed with the database system. Find and note the location of these JAR files.
10. If you use DB2 for z/OS, decide on the subsystem to use. Plan the values that you will substitute for storage group name, database name (not the subsystem name), and schema qualifier in the script files `createTablespace.sql` and `createSchema.sql`.
11. Check the requirements for the DB2 for z/OS Universal JDBC Driver provider and data source.
12. Decide which server will host the database. If the database server is remote, you need a suitable database client or a type-4 JDBC driver that has XA-support.
13. Decide the values for the following configuration parameters that you will need to specify for the database:
    - The Java Database Connectivity (JDBC) provider can be type-2 or type-4
    - The subsystem name.
    - The storage group name.
    - The database name.
    - The schema qualifier.

      Restriction: If you use an Informix database, it must be created in ANSI mode to support using a schema qualifier. Currently, it can support only one schema.
    - User name to create the schema.
    - The name or IP address of the database server.
    - The port number used by the database server. This is only required if you are using a type-4 JDBC driver.

- The user ID and password for the authentication alias. This is the user ID that the jdbc/BPEDB data source uses to access the database at runtime. These are the -dbUser and -dbPwd parameters for bpeconfig.jacl.

14. Plan to support enough parallel JDBC connections:
    a. Estimate the maximum number of parallel JDBC connection required to the Business Process Choreographer BPEDB database. This will depend on the nature of your business processes and the number of users. A good estimate is the maximum number of clients that can concurrently connect through the Business Process Choreographer API plus the number of concurrent endpoints defined in the BPEInternalActivationSpec and HTMInternalActivationSpec JMS activation specifications, plus a 10% safety margin to allow for overload situations.
    b. Make sure that your database system can support the necessary number of parallel JDBC connections.
    c. Plan suitable settings according to the best practices for your database system to support the expected number of parallel JDBC connections.
15. For a production system, make plans for the following administration tasks:
    - Tune your database after it is populated with typical production data.
    - Regularly delete completed process instances and task instances from the database. For an overview of the tools and scripts that are available, see Cleanup procedures for Business Process Choreographer.

Results

You have planned the database for Business Process Choreographer.

Planning the reporting database:

Plan the database for the Business Process Choreographer Explorer reporting function.

About this task

Business Process Choreographer Explorer reporting function can use the same database, but using an additional database gives better performance. If you will not reuse the BPEDB database, perform the following:

Procedure

1. If you plan to have multiple event collector instances, and they are going to use the same database, plan unique schema names for each event collector. For better performance, plan a database for each event collector.
2. Decide which database system to use for the database:

   Note: The Business Process Choreographer Explorer reporting function requires a database that is either Derby or DB2. Derby serializes database access. Activities are therefore always performed sequentially, even in flows that are modeled to support the parallel execution of activities.
3. Decide which server will host the database.
4. If you have not already planned the user IDs for the database, perform Table 7 on page 30.
5. If you are **not** using a Derby database for the reporting database, decide whether you will use SQL-based or Java-based user-defined functions (UDFs).

- The Java UDFs are more precise, but to be able to use them, you must install a JAR file in the database.
- If you use a DB2 for z/OS database, and would prefer that the database is created using Java-based UDFs, rather than SQL-based UDFs, then you have no choice but to use the menu-driven administration tool, setupEventCollector.
- If you use a Derby database, Java-based UDFs will be used because the embedded Derby database does not support SQL UDFs.

For more information about UDFs, see User-defined functions for Business Process Choreographer Explorer reporting function.

6. If you will not use the `bpeconfig.jacl` script to configure the Business Process Choreographer Explorer reporting function and event collector to use the BPEDB database, decide how you will create the reporting database.

   Using the menu-driven administration tool, setupEventCollector

   You can use this tool to create the database in an interactive mode, with your input validated against the runtime environment. If you use this tool, decide whether you want the tool to create an SQL file, but not run it – use this option if you want to customize the SQL before running it or to give to your database administrator to customize and run. For more information about this tool, see setupEventCollector tool.

   Unlike the other ways to create the database, this tool allows you to create either Java-based user-defined functions (UDFs) or SQL-based UDFs. You can also use it to switch between these two options, and also to install and remove the JAR file that is required to support the UDFs. For a DB2 on z/OS database, the tool supports creating the database using either Java-based UDFs or SQL-based UDFs. For a Derby database, only Java-based UDFs are used to create the database.

   Running SQL scripts

   You might need to use the SQL scripts if you are not allowed to use a tool to access the database. If you configured Business Process Choreographer using the bpeconfig.jacl script in batch mode or using the administrative console, an SQL script is generated that has all necessary parameters substituted. Otherwise, you can use the database design tool to generate the SQL scripts interactively.

   For a DB2 on z/OS database, because the scripts use SQL-based UDFs, you do not need to set up a Work Load Manager (WLM) environment that is enabled for Java. For a Derby database, only Java-based UDFs are used to create the database.

   Automatically create tables on first use

   Selecting the **Create tables** option on the administrative console's Business Process Choreographer event collector configuration page is an easy way to get a default database schema. This option is not suitable for high performance systems. This option cannot be used for a DB2 on z/OS database. For a Derby database, only Java-based UDFs are used to create the database.

   Note: If you use a Derby Network server data source, you must start the Derby network server from the directory *install_root*`/derby/bin/networkServer`, otherwise creating the tables will fail with the error `CWWBO4013E: The bpcodbutil.jar file could not be found on the Derby network server`.

7. If you use a DB2 for z/OS database, plan the following:
   - Location name (network name) of the subsystem.
   - The storage group name.
   - The database name as known by the subsystem. The default value is OBSRVRDB
   - The user ID to use to connect to the database. You must also know the password for this user ID.
   - The database schema name (SQLID), under which, the database objects are created.
   - Plan in which storage group the table spaces will be created:
     – Regular table space for OBSVR01, OBSVR02, OBSVR03, OBSVR04, OBSVR05, OBSVR06, OBSVR07, and OBSVR08.
     – LOB table space for OS26201, OS26202, OS26203, and OS26204.
   - If you want to use the Java-based user-defined function (UDFs) rather than the default SQL ones, decide on the name of the WLM environment that you will use to run the functions in.
   - If you will use the setupEventCollector tool to setup the database, also plan the following:
     – Decide which type of JDBC driver to use:
       - Type 4, connecting directly via JDBC. In this case, also make sure that you know the following:
         - The host name or IP address of the database server. The default is `localhost`.
         - The port number used for the database. The default is `446`.
         - The directory for the JDBC driver JAR files, `db2jcc.jar` and `db2jcc_license_cisuz.jar`.
       - Type 2, connecting using a native database client. In this case, also plan what the database alias will be in the local catalog.
   - Check the requirements for the DB2 for z/OS Universal JDBC Driver provider and data source.
8. If you use a Derby database, plan the following:
   - The database name. This must be the fully qualified path on the server's file system. The default value is *install_root*`/databases/BPEDB`.
   - The database schema name, under which, the database objects are created. The default value is `APP`.
   - If you will use the setupEventCollector tool to setup the database, also plan the following:
     – If you use the Derby Network JDBC driver, plan the user ID to use to connect to the database. You must also know the password for this user ID.
     – Decide which type of JDBC driver to use:
       - Embedded JDBC driver or Embedded 40 JDBC driver. In this case, also plan the directory for the JDBC driver JAR file `derby.jar`. The default location is *install_root*`/derby/lib`.
       - Network JDBC driver or Network 40 JDBC driver. In this case, also make sure that you know the following:
         - The directory for the JDBC driver JAR file `derbyclient.jar`. The default location is *install_root*`/derby/lib`.

- If using a Derby Network server, decide on the location of the UDF JAR file `bpcodbutil.jar` on the Derby network server. The default location is *`install_root`*`/derby/lib`.
- The host name of the database server. The default is `localhost`.
- The port number used for the database. The default is `1527`.

9. If you use the **bpeconfig.jacl** tool in batch mode with the -createEventCollector yes option, plan one of the following:
   - The -createDB yes option causes the tool to run the SQL script that bpeconfig.jacl generates. You can use the -dbSchema parameter to specify a schema qualifier for the BPEDB database, and you can use the -reportSchemaName and -reportDataSource parameters to make the Business Process Choreographer Explorer reporting function use a different database rather than using the BPEDB database.
   - The -createDB no option prevents the tool from running the SQL script that it generates. The generated SQL files are based on the standard SQL files that are provided for your database, but with all configuration parameters that are provided to the bpeconfig.jacl tool pre-filled in the SQL file, which minimizes the customization necessary. Plan that you or your database administrator will customize the generated SQL script that creates the database objects before running it. For more information about the tool and other database parameters, see Using the bpeconfig.jacl script to configure Business Process Choreographer.
10. If you will use administrative console's **Business Process Choreographer event collector page** to create the database tables, plan one of the following:
    - For a Derby database, you can use the Create tables option to cause the tool to create the default schema in the specified database the first time that Business Process Choreographer accesses the database.
    - If you want to run an SQL script to prepare the database tables, do not use the Create tables option. Plan that you or your database administrator will customize a copy of the SQL script that creates the database objects before running it. This option is most suitable for a production system.
11. If you want to preview the SQL script for your database, so that you can plan what customizations you will make: Locate and view the `createSchema_Observer.sql` file for your database, but do not modify it. The SQL files are located in:

    `install_root``/dbscripts/ProcessChoreographer/`*`database_type`*

    Where *database_type* is one of the following:
    - DB2zOSV8
    - DB2zOSV9
    - Derby

    Note: If you use the `bpeconfig.jacl` tool to configure Business Process Choreographer, plan to use the SQL script that the tool generates, which does not need to be edited to substitute values for placeholders for configuration parameters. The generated scripts are only available after running the tool, but they are based on the scripts in the locations listed above. You will still have to edit the generated script file if you want to customize the table space allocations. Alternatively, you can use the database design tool to generate the SQL scripts.

Results

You have planned the reporting database.

Planning the messaging engine database:

For high-load setups, where database logging might become a bottleneck, you can improve performance by using a separate database for the messaging engine for the Business Process Choreographer bus.

About this task

You can use the same messaging database for each messaging engine for the Service Component Architecture (SCA) system bus, each messaging engine for the SCA application bus, each messaging engine for the Common Event Infrastructure bus, and each messaging engine for the Business Process Choreographer bus. The database should be accessible to all members of the cluster that hosts the message engine to ensure failover availability of the message engine. If performance is important, plan to use a dedicated database for the Business Process Choreographer messaging engine, rather than using the default MEDB that is used for the SCA bus and applications.

Procedure

1. If you use the **Installer** or **Profile Management Tool** to get one of the sample Business Process Choreographer configurations, decide whether the Business Process Choreographer messaging engine will use Derby embedded, file store, or the WPRCSDB database.
2. The Java Database Connectivity (JDBC) provider. Note that the file store and embedded Derby database are not available in a network deployment environment.
3. If you want to use WebSphere MQ, you must use the bpeconfig.jacl configuration script to configure Business Process Choreographer. Using WebSphere MQ is deprecated.
4. If you use the bpeconfig.jacl configuration script to configure Business Process Choreographer, Business Process Choreographer will use the same type of messages store that is used by SCA.
   - If SCA uses a FILESTORE, then Business Process Choreographer will also use a FILESTORE.
   - If SCA uses a Derby Embedded database, then Business Process Choreographer will use its own Derby Embedded database.
   - If SCA uses any other database, then Business Process Choreographer will use its own schema in the same database.
5. If you use the administrative console's Business Process Choreographer configuration page, if you want to use the default configuration that is based on the SCA message store settings, plan to select the **Use the default configuration** check box, otherwise, plan the following configuration parameters:
   - Local or remote bus member location.
   - The name of the database. The default is `BPEME`.
   - The schema name. The default is `MEDBPM00`.
6. If you are using a file store or the embedded Derby JDBC provider, the message stores will be created automatically.

7. If you are not using a file store or the embedded Derby JDBC provider, plan the following configuration parameters.
   a. Plan that the database will already exist before Business Process Choreographer is started.
   b. The host name or IP address of the database server, and the port number that it uses.
   c. The user name used to connect to the database and to create the schema. This is the user ID that you planned in Table 8 on page 31.

Results

You have planned the database for the Business Process Choreographer messaging engine.

Planning for the Business Flow Manager and Human Task Manager

The core of a Business Process Choreographer configuration consists of the Business Flow Manager and the Human Task Manager. You must plan their configuration parameters.

Procedure

1. Make sure you know the Java Message Service (JMS) provider user ID that will be used as the run-as user ID for the Business Flow Manager message driven bean. In the administrative console, and in Table 9 on page 31, it is known as the **JMS API Authentication User**.
2. Make sure you know the Java Message Service (JMS) provider user ID that will be used as the run-as user ID for the Human Task Manager message driven bean. In the administrative console, and in Table 9 on page 31, it is known as the **Escalation User Authentication User**.
3. Make sure you know which groups or user IDs the security roles for administrator and monitor will map onto. For details, see Table 10 on page 32.
4. If you want the Human Task Manager to send e-mail notifications of escalation events, identify the host name or IP address where the Simple Mail Transfer Protocol (SMTP) e-mail server is located. Plan what the sender address should be for email notifications. If the e-mail service requires authentication, make sure you know the user ID and password to use to connect to the service.
5. Decide on the context root for the Web service binding of the API.
   - When configured on a server:
     - The default for the Business Flow Manager is `/BFMIF_nodeName_serverName`.
     - The default for the Human Task Manager is `/HTMIF_nodeName_serverName`
   - When configured on a cluster:
     - The default for the Business Flow Manager is `/BFMIF_clusterName`
     - The default for the Human Task Manager is `/HTMIF_clusterName`
6. If you are going to use the Business Process Choreographer Explorer, the Business Space, or a client that uses the Representational State Transfer (REST) API or the JAX Web Services API, decide on the context roots for the REST API and the JAX Web Services API.
   - The defaults for the Business Flow Manager are `/rest/bpm/bfm` and `/BFMJAXWSAPI`.
   - The defaults for the Human Task Manager are `/rest/bpm/htm` and `/HTMJAXWSAPI`.

- When configured on a server, or on a single cluster, or on multiple clusters that are mapped to different Web servers, you can use the default values.
- When configured in a network deployment environment on multiple deployment targets that are mapped to the same Web server, do not use the default values. The context root for each Business Process Choreographer configuration must be unique for each combination of host name and port. You will have to set these values manually using the administrative console after configuring Business Process Choreographer.

7. Decide whether you want to initially enable audit logging for the Business Flow Manager, or Human Task Manager, or both.
8. If you are going to use the Business Process Choreographer Explorer reporting function, decide whether you want the Business Flow Manager to be initially configured to generate Common Event Infrastructure logging events.

Results

You have planned all the initial configuration parameters for the Business Flow Manager and Human Task Manager. You can change any of these setting anytime later using the administrative console.

Planning for the people directory provider

Plan the people directory provider, people substitution, virtual member manager, and Lightweight Directory Access Protocol (LDAP) settings for Business Process Choreographer.

Procedure

1. If you are going to use human tasks, decide which people directory providers you will use:

   Virtual member manager (VMM) people directory provider
   The VMM people directory provider is ready to use federated repositories (also known as virtual member manager) as is preconfigured for WebSphere security – using a file repository. If you want to use a different user repository with federated repositories, you will need to reconfigure federated repositories. The VMM people directory provider supports all Business Process Choreographer people assignment features including substitution. It relies on the features provided by federated repositories, such as support for different repository types, such as LDAP, database, file based, and property extension repository.

   To use the VMM people directory provider requires that you have configured federated repositories for WebSphere Application Server security. You can associate federated repositories with one or more user repositories, based on a file, LDAP, or a database. For more information about this, see Managing the realm in a federated repository configuration. For more information about using federated repositories, see IBM® WebSphere Developer Technical Journal.

   Lightweight Directory Access Protocol (LDAP) people directory provider
   This people directory provider must be configured before you can use it. Perform the planning in step 2 on page 46.

   System people directory provider
   This people directory provider can be used without configuring it. Do not use this provider for a production system, it is only intended for application development testing.

User registry people directory provider

This people directory provider can be used without configuring it. Depending on the WebSphere security realm definition, the user registry can use one of the following repositories:

- Federated repository – which can use the following:
  - File registry
  - One or more LDAPs
  - One or more databases
- Standalone LDAP
- Standalone custom
- Local operating system

2. If you are going to use the Lightweight Directory Access Protocol (LDAP), plan the following.
   a. You might need to customize your own version of the `LDAPTransformation.xsl` file. For the location of that file and a list of properties that you might need to customize, see Configuring the LDAP people directory provider.
   b. Plan the following LDAP custom properties:

LDAP plug-in property	Required or optional	Description
AuthenticationAlias	Optional	The authentication alias used to connect to LDAP, for example, `mycomputer/My LDAP Alias`. You must define this alias in the administrative console by clicking **Security** → **Secure administration, applications, and infrastructure** → **Java Authentication and Authorization Service** → **J2C Authentication Data**. If this alias is not set or if AuthenticationType is not set to `simple` then an anonymous logon to the LDAP server is used.
AuthenticationType	Optional	If this property is set to `simple`, for simple authentication, then the AuthenticationAlias parameter is required. Otherwise, if it is not set, anonymous authentication is used.
BaseDN	Required	The base distinguished name (DN) for all LDAP search operations, for example, `o=mycompany, c=us`. To specify the directory root, specify an empty string using two single quotes, `''`.
Casesentiveness ForObjectclasses	Optional	Determines whether the names of LDAP object classes are case-sensitive.
ContextFactory	Required	Sets the Java Naming and Directory Interface (JNDI) context factory, for example, `com.sun.jndi.ldap.LdapCtxFactory`
ProviderURL	Required	This Web address must point to the LDAP JNDI directory server and port. The format must be in normal JNDI syntax, for example, `ldap://localhost:389`. For SSL connections, use the LDAP's URL. For a high availability configuration, where you have two or more LDAP servers that maintain mirrored data, plan to specify a URL for each LDAP server and use the space character to separate them.
SearchScope	Required	The default search scope for all search operations. Determines how deep to search beneath the baseDN property. Specify one of the following values: `objectScope`, `onelevelScope`, or `subtreeScope`
additionalParameter Name1-5 and additionalParameter Value1-5	Optional	Use these name-value pairs to set up to five arbitrary JNDI properties for the connection to the LDAP server.

3. If you are going to use the virtual member manager, plan the following.
   a. You might need to customize your own version of the `VMMTransformation.xsl` file. For the location of that file and a list of properties that you might need to customize, refer to Configuring the Virtual Member Manager people directory provider.
4. If you want to use people substitution, consider the following:
   - You must use the VMM people directory provider. The LDAP, system, and user registry people directory providers do not support people substitution.
   - If you are going to use people substitution in a production environment, plan to use the VMM Property Extension Repository to store the substitution information. The Property Extension Repository and, implicitly, the selected database must be unique and accessible from within the whole cell. As the BPEDB database is not necessarily unique within a cell, BPEDB cannot be used. You can use the common database, WPSRCDB, to host the Property Extension Repository, however, for a production environment, it is recommended to use a database that is independent of other WebSphere Process Server databases.
   - To use people substitution in a single-server test environment, you can store people substitution information in the internal file registry that is configured for federated repositories.

Results

You have planned the people directory provider and people assignment options.

Planning for the Business Process Choreographer Explorer

Plan the configuration options and parameters for the Business Process Choreographer Explorer.

About this task

If you will use the Business Process Choreographer Explorer you can either configure it at the same time as you configure Business Process Choreographer, or you can do it later. The Business Process Choreographer Explorer reporting function is optional.

Procedure

1. Decide how many Business Process Choreographer Explorer instances you want to configure. You can easily create the first instance while configuring Business Process Choreographer. Possible reasons and considerations include:
   - Because each Business Process Choreographer Explorer instance can only connect to one Business Process Choreographer configuration, if you have more than one Business Process Choreographer configuration in your environment, it makes sense to set up a Business Process Choreographer Explorer instance for each configuration.
   - You might want to have two or more different customized versions of the Business Process Choreographer Explorer connecting to the same Business Process Choreographer configuration. You can customize each version independently, for more information about what you can customize, see Customizing Business Process Choreographer Explorer.
   - You can configure multiple Business Process Choreographer Explorer instances on each server or cluster.

- The instances can be created on any deployment target regardless of where there are Business Process Choreographer or Business Process Choreographer event collector configurations.
- Because each Business Process Choreographer Explorer instance's reporting function can only connect to one Business Process Choreographer event collector, plan to configure equal numbers of Business Process Choreographer Explorer instances with the reporting function as there are Business Process Choreographer event collectors.

2. For each Business Process Choreographer Explorer instance that you want, plan the following:
   a. The context root for the Business Process Choreographer Explorer. It must be unique within the cell. The default is `/bpc`.
   b. The URL for the Business Process Choreographer Explorer that will be inserted in escalation e-mails.
   c. The URL for the Business Flow Manager and Human Task Manager representational state transfer (REST) APIs endpoints. They must match the values for the context roots that you planned for the REST APIs. For example, if the context root for the Human Task Manager Web service is `/rest/bpm/htm`, the endpoint URL for the Human Task Manager REST API endpoint would be `http://`*`hostname`*`:`*`port`*`/rest/bpm/htm`.
   d. The maximum number of results to be returned for a query - the default is 10000.
   e. The deployment target (server or cluster) of the Business Process Choreographer instance that this Business Process Choreographer Explorer will manage.
   f. Optional: If you will use the Business Process Choreographer Explorer reporting function, perform "Planning for the Business Process Choreographer Explorer reporting function." You can also plan and configure it later.

Results

You have planned the configuration options for the Business Process Choreographer Explorer.

Planning for the Business Process Choreographer Explorer reporting function:

Plan to configure the Business Process Choreographer Explorer reporting function and event collector.

About this task

If you will use the Business Process Choreographer Explorer reporting function, you can either configure it when you configure Business Process Choreographer Explorer, or you can do it later.

Procedure

1. Because security roles are not used to restrict access to the Business Process Choreographer Explorer reporting function, if you do not want all Business Process Choreographer Explorer users to have access to the reporting function, plan to configure a separate Business Process Choreographer Explorer instance for the reporting function, and make it inaccessible to normal users.
2. Understand the purpose and relationships between the different Business Process Choreographer Explorer reporting function topology elements.

The Business Process Choreographer Explorer reporting function.
Before version 6.2, this feature was available as the Business Process Choreographer Observer. Since version 6.2, this feature is integrated in the Business Process Choreographer Explorer, and is available on the **Reports** tab. You must configure the Business Process Choreographer Explorer reporting function before you can use it.

The event collector application.
This application must be deployed on a server or cluster where the Common Event Infrastructure (CEI) server is configured. You can only have one event collector on each CEI deployment target. It does not need to be deployed where Business Process Choreographer has been configured. It receives business process events from CEI, transforms them, and writes them to the reporting database.

The reporting database.
The event collector and Business Process Choreographer Explorer reporting function communicate by using the same database. For non-production systems, the database can be shared with other components.

Your choices are independent of the topology you have for your Business Process Choreographer setup. For more insight into the possibilities, see "Business Process Choreographer Explorer reporting function overview" on page 54.

3. Identify the purpose of your setup, your system requirements, and the topology implications.

   Simple setup
   For simpler configuration and administration, but lower performance, deploy the event collector application on the same deployment target as you have the Business Process Choreographer Explorer and CEI configured on, and use a local database system.

   High load production system: network deployment
   Use a cell of multiple nodes, with multiple clusters. Install instances of the Business Process Choreographer Explorer on any deployment targets in the cell. Install the event collector application on the cluster where you have configured the Common Event Infrastructure (CEI). Use a separate database server.

4. If you have not already planned the database for the Business Process Choreographer Explorer reporting function, perform "Planning the reporting database" on page 39.
5. For each event collector instance that you want to configure, plan the following:
   a. Decide where you will install it. You can only install one event collector instance per deployment target, and the deployment target must have CEI configured on it.
   b. Decide how you will configure this event collector instance:
      - Using the administrative console page. For more information about this option, see Using the administrative console to configure a Business Process Choreographer event collector.
      - Using the interactive setupEventCollector tool. For more information about this option, see Using the setupEventCollector tool to configure a Business Process Choreographer event collector.

- At the same time as creating a Business Process Choreographer configuration, using the `bpeconfig.jacl` script. The `-createEventCollector` option has the default value `yes`.

  Note: Do not use `bpeconfig.jacl` to configure the Business Process Choreographer Explorer reporting function for a high-performance system, because `bpeconfig.jacl` will configure the event collector and Business Process Choreographer Explorer reporting function applications on the same deployment target as the Business Process Choreographer configuration. For more information about this option, see Using the bpeconfig.jacl script to configure Business Process Choreographer. You cannot use `bpeconfig.jacl` to configure the event collector in interactive mode.

c. Plan the data source:
   - If the Business Process Choreographer Explorer reporting function shares the same physical database as Business Process Choreographer, plan to use a separate data source for the reporting database, and plan its JNDI name.
   - Plan an authentication alias that will be used for the database.
   - Plan to create the data source with a cell scope.

d. Plan the configuration parameters required when configuring the event collector:
   - The JNDI data source name for the reporting database.
   - The schema to be used for the database objects. The default is the user ID that is used to connect to the database.
   - The user ID to use to connect to the database. The default depends on the database: For DB2 the default is db2admin, for Oracle the default is system, and for other databases the default is the user ID of the logged on user.
   - The password for the user ID.
   - If you are using a type 4 JDBC connection, also collect the host name or IP address of the database server and the port number that it uses
   - Decide where the event collector will be deployed. The deployment target must have CEI configured on it, so if you have a separate cluster for CEI, plan to deploy the event collector on the same cluster.
   - If you will deploy the event collector in a network deployment environment, know on which deployment target the messaging engine for the CEI bus is configured.
   - If the CEI bus has security enabled, plan the JMS user ID that will be used to authenticate with the CEI bus.
   - Decide whether you want to enable CEI event logging business events when configuring the event collector, or whether you will enable it later using the administrative console or by running a script.

e. Plan the runtime configuration values, which you might need to customize to suit your needs after configuring the event collector:
   - BpcEventTransformerEventCount
   - BpcEventTransformerMaxWaitTime
   - BpcEventTransformerToleranceTime
   - ObserverCreateTables
   - If the authentication alias user ID will not own the database schema, plan the ObserverSchemaName.

For more information about these values, see Changing configuration parameters for the Business Process Choreographer Explorer reporting function.

6. For each Business Process Choreographer Explorer reporting function that you configure, plan the following:
   - Decide how you will configure this instance:
     - At the same time as creating the Business Process Choreographer Explorer, using the administrative console page for the Business Process Choreographer Explorer. For more information about this option, see Using the administrative console to configure the Business Process Choreographer Explorer reporting function.
     - At the same time as creating the Business Process Choreographer Explorer, using the `clientconfig.jacl` script.
     - At the same time as creating a Business Process Choreographer configuration, using the `bpeconfig.jacl` script.

       Note: Do not use `bpeconfig.jacl` to configure the Business Process Choreographer Explorer reporting function for a high-performance system, because `bpeconfig.jacl` will configure the event collector and Business Process Choreographer Explorer reporting function applications on the same deployment target as the Business Process Choreographer configuration. For more information about this option, see Using the bpeconfig.jacl script to configure Business Process Choreographer.
   - The schema name for the reporting database.
   - The JNDI name for the data source that is used by the Business Process Choreographer Explorer to connect to the reporting database.
7. If you will use the `bpeconfig.jacl` script to configure Business Process Choreographer:
   - When the script is run in batch mode, the default is that it will also configure the event collector and Business Process Choreographer Explorer applications, and that they will be configured on the same deployment target as the Business Process Choreographer configuration.
   - If you do not want `bpeconfig.jacl` to configure one or both of the event collector and Business Process Choreographer Explorer reporting function, plan to use one or both of the `bpeconfig.jacl` options `-createEventCollector no` and `-reportFunction no`, which prevent `bpeconfig.jacl` from configuring them.

Results

You have planned the configuration options for the Business Process Choreographer Explorer reporting function and event collector.

Planning for a remote client application

Planning for a remote Business Process Choreographer client application that uses the Business Process Choreographer APIs and runs on a WebSphere Process Server client installation.

About this task

If you want an application to use the Business Process Choreographer APIs, you can use a WebSphere Process Server client installation to run the applications

remotely against a full WebSphere Process Server server installation. The client is easier to configure and administer than a full WebSphere Process Server installation.

The WebSphere Process Server client installation does not contain WebSphere Process Server profile templates, but needs to augment the underlying WebSphere Application Server profile with Feature Pack for SCA Version 1.0 with SDO 2.1.1. This means that you can even install the WebSphere Process Server client on top of an existing WebSphere Application Server installation that has federated profiles and those federated WebSphere Application Server profiles can take advantage of the WebSphere Process Server client functionality immediately. This scenario is not possible with the full WebSphere Process Server server because WebSphere Process Server does not support augmentation of already federated profiles.

Procedure

1. Plan to install a WebSphere Process Server client.
   - If you want WebSphere Portal Server to access Business Process Choreographer, you must have a compatible WebSphere Process Server client installed.

   Table 14. WebSphere Process Server client versions that WebSphere Portal Server can use to access Business Process Choreographer

	WebSphere Process Server client version			
WebSphere Portal Server version	**6.1.0.1**	**6.1.2**	**6.2**	**7.0**
6.1.0.1	Yes	Yes	No	No
6.1.0.2	Yes	Yes	Yes	No

   - Any existing profiles, including already federated profiles, can use WebSphere Process Server client immediately, because the client installation does not augment the base profile.
   - If there is no existing WebSphere Application Server installation, a WebSphere Application Server network deployment installation will be created.
2. Decide which type of Business Process Choreographer client application you will use:
   - Custom client application
   - Business Process Choreographer Explorer

   Note: If you use customized JavaServer Pages (JSP), as described in Developing JSP pages for task and process messages, make sure that you know where they are located.
3. If you are going to develop a custom client application that will use Business Process Choreographer, plan which interfaces the application will use. You can handle processes and tasks using one of the following:
   - Web services API, Java Messaging Service (JMS) API or representational state transfer (REST) API – remote client applications that are based on these APIs do not need any WebSphere Process Server installation.
   - JavaServer Faces (JSF) components
   - Enterprise JavaBeans™ (EJB) API

Note: If you develop a client application, which uses the Business Process Choreographer EJB APIs, it must be packaged in the way that is described in Accessing the remote interface of the session bean.

4. Decide or identify the type of cell where the WebSphere Process Server client will be installed:
   a. In a cell where a managed server or cluster is located, on which Business Process Choreographer is configured, the default configuration of the Remote Artifact Loader (RAL) allows the unsecured transmission of artifacts between the client and the server. This is known as the "single-cell" scenario.
   b. In a cell that does not have a managed server or cluster with Business Process Choreographer configured on it, there are different deployment managers. This is known as the "cross-cell" scenario. If your client application uses the EJB API, you must define a namespace binding so that the client application can locate the server or cluster where Business Process Choreographer is configured.

Results

You have planned for a remote Business Process Choreographer client application.

Business Process Choreographer overview

Describes the facilities provided by the Business Flow Manager and the Human Task Manager.

Business Process Choreographer is an enterprise workflow engine that supports both business processes and human tasks in a WebSphere Application Server environment. These constructs can be used to orchestrate services as well as integrate activities that involve people in business processes. Business Process Choreographer manages the life cycle of business processes and human tasks, navigates through the associated model, and invokes the appropriate services.

Business Process Choreographer provides the following facilities:

- Support for business processes and human tasks. Business processes offer the standard way to model your business process using the Web Services Business Process Execution Language (WS-BPEL, abbreviated to BPEL). With human tasks, you can use the Task Execution Language (TEL) to model activities that involve people. Both business processes and human tasks are exposed as services in a service-oriented architecture (SOA) or Service Component Architecture (SCA); they also support simple data objects and business objects.
- Application programming interfaces for developing customized applications for interacting with business processes and human tasks.
- Business Process Choreographer Explorer. This Web application allows you to administer business processes and human tasks. It also includes the optional Business Process Choreographer Explorer reporting function, formerly known as the Business Process Choreographer Observer, which allows you to observe the states of running processes.
- Human workflow widgets as part of Business Space. These widgets allow you to manage work, create tasks for other people, and initiate services and processes.

Business Process Choreographer Explorer overview

Business Process Choreographer Explorer is a Web application that implements a generic Web user interface for interacting with business processes and human tasks.

It also includes an optional reporting function, which was previously known as the Business Process Choreographer Observer.

You can configure one or more Business Process Choreographer Explorer instances on a server or cluster. It is sufficient to have a WebSphere Process Server installation with a WebSphere Process Server profile, or a WebSphere Process Server client installation – it is not necessary to have Business Process Choreographer configured on the server or cluster. The WebSphere Process Server client installation is only the infrastructure that you need to connect a client to a WebSphere Process Server, it does not contain the Business Process Choreographer Explorer. Use the deployment manager to install the Business Process Choreographer Explorer on the servers in the WebSphere Process Server client installation as well.

A single Business Process Choreographer Explorer can only connect to one Business Process Choreographer configuration, though it does not have to connect to a local configuration. However, you can configure multiple instances of the Business Process Choreographer Explorer on the same server or cluster, and each instance can connect to different Business Process Choreographer configurations.

When you start Business Process Choreographer Explorer, the objects that you see in the user interface and the actions that you can take depend on the user group that you belong to and the authorization granted to that group. For example, if you are a business process administrator, you are responsible for the smooth operation of deployed business processes. You can view information about process and task templates, process instances, task instances, and their associated objects. You can also act on these objects; for example, you can start new process instances, create and start tasks, repair and restart failed activities, manage work items, and delete completed process instances and task instances. However, if you are a user, you can view and act on only those tasks that have been assigned to you.

Business Process Choreographer Explorer reporting function overview:

About Business Process Choreographer Explorer reporting function.

You can use the Business Process Choreographer Explorer reporting function to create reports on processes that have been completed. You can also use it to view the status of running processes. This describes the architecture and possible configuration paths.

The Business Process Choreographer Explorer reporting function uses the Common Event Infrastructure (CEI) to collect events that are emitted by WebSphere Process Server. You can either use a number of predefined reports or define your own reports to get an overview of the number of processes, activities, or other aggregate data. You can also get information about specific processes or activities.

The Business Process Choreographer Explorer reporting function is based on two Java EE applications, which are shown in the following figure:

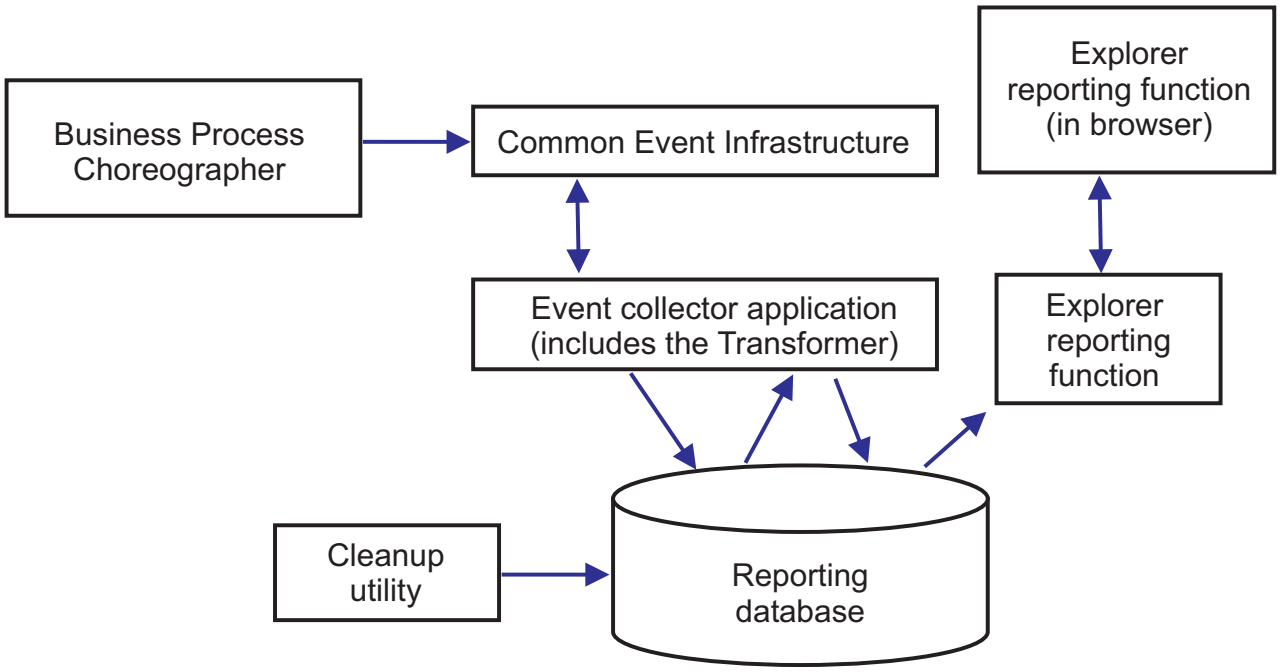


Figure 1. Architecture

- The event collector application reads event information from the CEI bus and stores it in the event collector table in the reporting database.
- The reporting database is a set of database tables that store the event data.
- Periodically the event transformer is triggered, which transforms the raw event data into a format that is suitable for queries from the Business Process Choreographer Explorer reporting function.
- The Business Process Choreographer Explorer reporting function generates the reports and performs other actions that the user can initiate using the graphical user interface (GUI).
- You can use the GUI to generate your reports. You can also store and retrieve reports that you have defined.
- A cleanup utility can be used to remove records from the database, which can help to improve the performance.

Simple configurations

A simple configuration, where performance is not an important consideration is illustrated in the following figure.

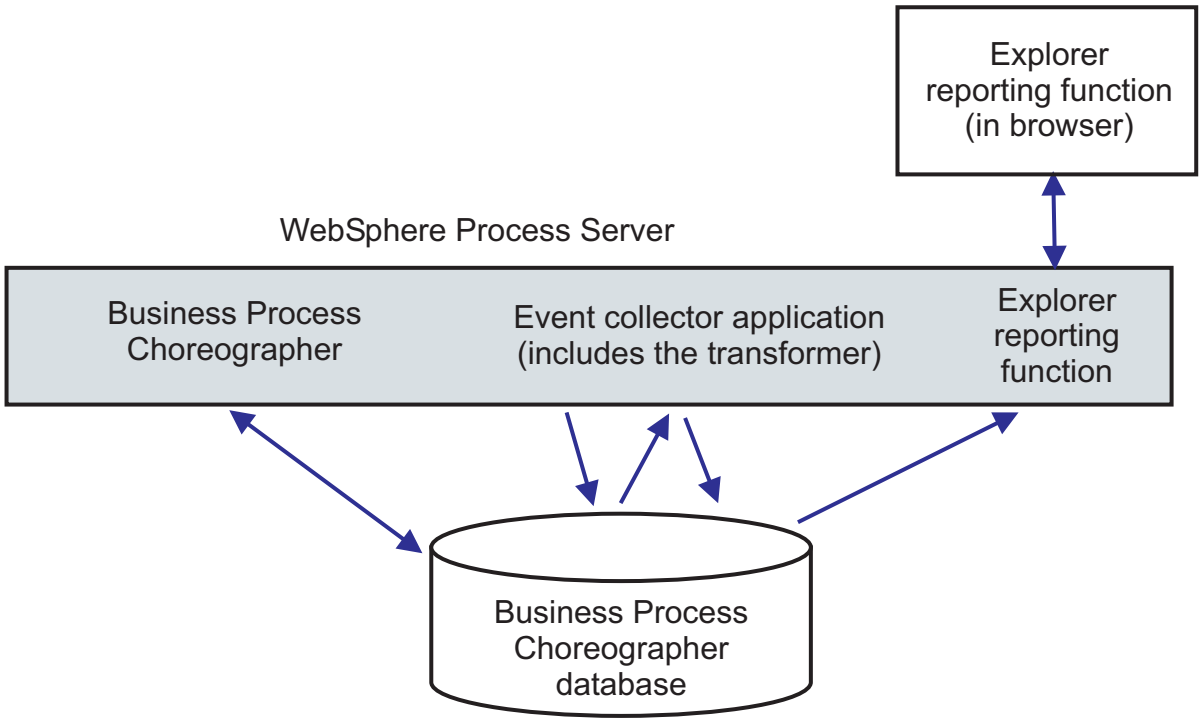


Figure 2. Stand-alone setup

Everything is installed on a single system, and Business Process Choreographer and Business Process Choreographer Explorer reporting function use the same database.

This kind of simple configuration is created if you create a sample Business Process Choreographer configuration. Also the bpeconfig.jacl tool defaults to configuring this kind of setup on the same deployment target as the Business Process Choreographer configuration. Common Event Infrastructure (CEI) logging will be enabled and the necessary database schema is created in the Business Process Choreographer Derby database BPEDB. This configuration path can be ideal if performance is not an important consideration.

High-performance configurations

Interactive configuration tools are provided which give you the freedom to exploit the full potential of the Business Process Choreographer Explorer reporting function architecture. For example, in an ideal configuration for performance, the Business Process Choreographer configuration, CEI event server, and the Business Process Choreographer Explorer (with reporting function) run on separate systems, and Business Process Choreographer and the Business Process Choreographer Explorer reporting function have their own databases.

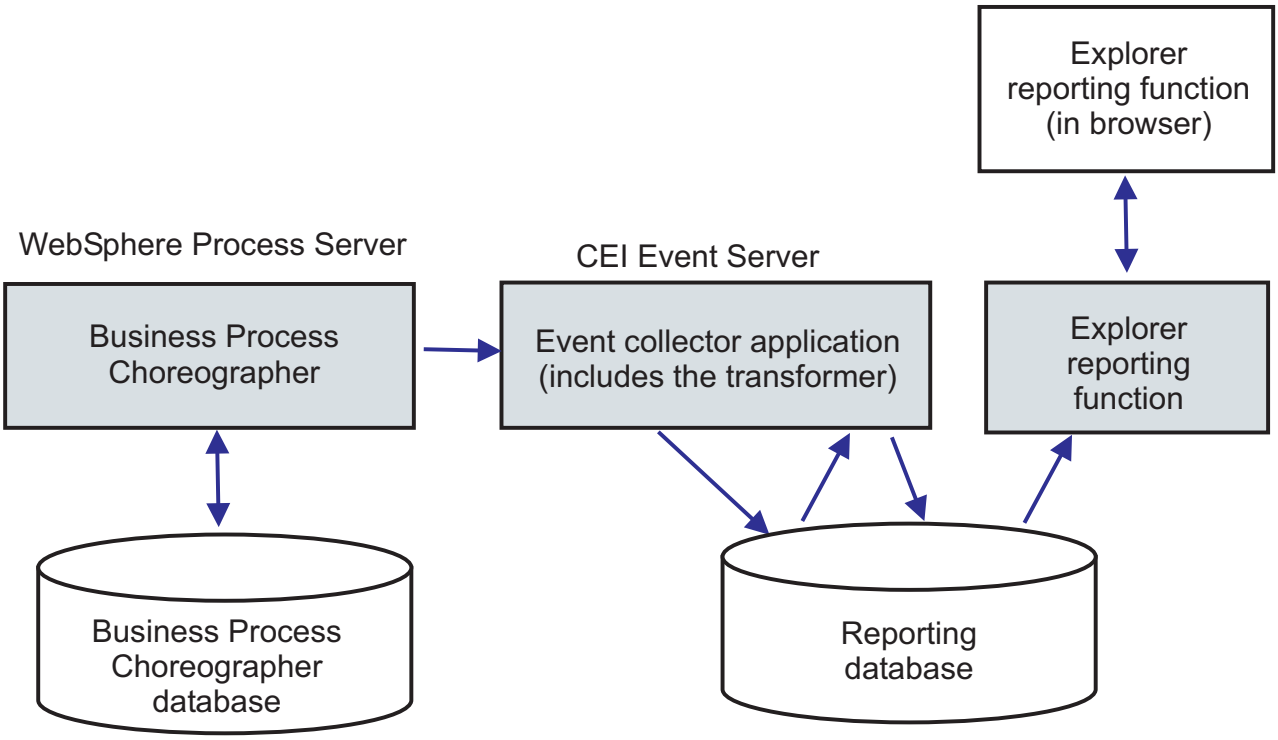


Figure 3. Business Process Choreographer Explorer reporting setup for production performance

If you want to use a separate database for the Business Process Choreographer Explorer reporting function, or to add the Business Process Choreographer Explorer reporting function to an existing Business Process Choreographer configuration in a clustered setup, or to use more sophisticated database options, perform Configuring the Business Process Choreographer Explorer reporting function and event collector.

In a network deployment environment

The following constraints apply if you want to configure Business Process Choreographer Explorer reporting function in a network deployment environment.

- CEI must be configured in your cell.
- As illustrated in the previous figure, the Business Process Choreographer event collector must be configured on a deployment target where the CEI Event server is configured. If the CEI Event server is configured on a different cluster than Business Process Choreographer, you must configure the Business Process Choreographer event collector on a deployment target where the CEI Event server is configured. The Business Process Choreographer Explorer reporting function application does not need to be installed on the same system as the event collector.

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.

Procedure

- 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 information

Configuring class loaders of a server

Learning about service integration buses

Messaging engines

Using multiple platforms within a cell

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

For example, you can create a deployment manager cell that includes 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 managers" on page 99
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 99
A *managed node* is a node that is federated to a deployment manager and contains a node agent and can contain managed servers. In a managed node, you can configure and run managed servers.

Related information

Heterogeneous Cells - cells with nodes on mixed operating system platforms

Understanding topologies

Your choice of a topology depends on your business goals and objectives and on the design and purpose of the applications that will be deployed to your environment.

Before installing and configuring the product you need to understand what a topology is. You need to know the basic set of functions provided in all topologies and you need to understand how topologies differ from one another, not only in how the components and resources are allocated, but under what scenarios is one topology is better suited than another.

Use the information in this section to:

- Learn about topologies
- Learn about the considerations for choosing a topology
- Learn about the methods for implementing a topology

Related concepts

"Planning overview" on page 1
The activities associated with implementing WebSphere Process Server as part of a BPM solution vary depending on the planned usage.

Topologies and deployment environment patterns

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

What is a topology?

A topology is the physical layout of the *deployment environment* required to meet your business needs for capacity, availability, and scalability.

There are a variety of factors that affect how you design and implement your topology. For example, business and application requirements, resource requirements and constraints, the intended purpose of the environment and the operating system on which you install and configure the product all play a role when choosing a topology and how to design and implement the topology. For information on assessing the business requirements and application requirements that can affect your topology, see Determining your software needs.

WebSphere Process Server includes patterns for the following topologies, which you can use to address a variety of business scenarios, from a proof-of-concept (POC) design to a fully functional production environment:

- Single Cluster
- Remote Messaging
- Remote Messaging and Remote Support
- Remote Messaging, Support and Web

Each of the topologies listed above have certain design characteristics that address or solve a particular business need. For example, the single cluster topology is typically used for a testing or proof of concept scenario. The design characteristics of each topology have been captured as *patterns* that are supplied as configuration templates with the product.

You are not obligated to use the IBM-supplied patterns. If none of the patterns address your specific need, you can create a custom topology.

The purpose of deployment environment patterns

A deployment environment pattern specifies the constraints and requirements of the components and resources involved in a deployment environment. There are IBM-supplied patterns for each topology layout. These patterns provide rules and guidelines of component interaction that are characteristic of the most commonly used BPM topologies. The IBM-supplied patterns are based on well-known and tested configuration scenarios and offer a repeatable and automated method of creating a deployment environment that best suits your needs. Each pattern is designed to meet the configuration requirements and business needs of the associated topology. Using patterns helps you create a deployment environment in the most straightforward way.

Because the deployment environment patterns represent well-known, tested, and recommended topologies with component configurations that work together, their use ensures that you build a high quality, fully functional deployment environment. You can use the configuration rules of a deployment environment pattern to generate a fast path configuration. This is possible because many design decisions are implemented in the pattern; for example, which components to configure, and which default parameters and resources are needed.

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

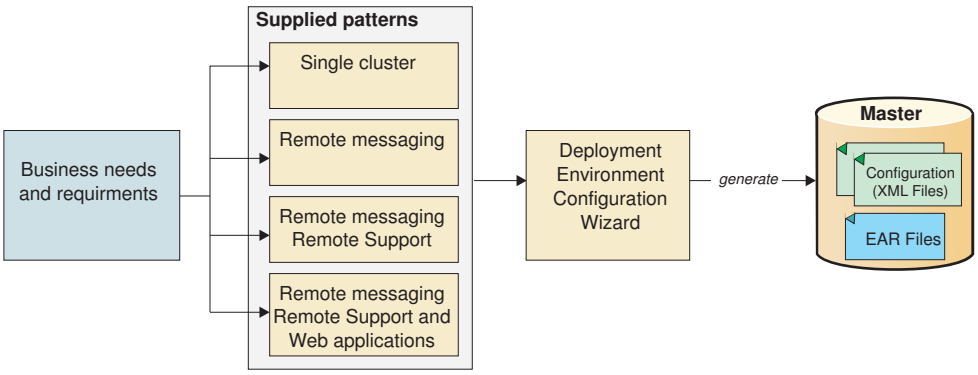
To create a deployment environment that fits one of the patterns, you make selections during installation, or profile creation, or in the administrative console.

With regard to using patterns, you should:

- Understand the requirements of the business solution you are creating
- Review and understand the capabilities and characteristics of the patterns that are available to you
- Decide which pattern to use.

  The following diagram demonstrates how business needs and requirements drive which pattern to choose and how the Deployment Environment configuration wizard generates the environment based on the chosen pattern.

  Note: Using the Deployment Environment Configuration wizard is one method for configuring the deployment environment, but it is not the only method. You can achieve the same results by configuring clusters manually (using scripting or the administrative console).



If none of the WebSphere Process Server patterns suit your needs you can create a custom environment.

Functions of IBM-supplied deployment environment patterns

Any WebSphere Process Server deployment contains a basic set of functions that together form a complete production environment.

To design a robust deployment environment, you need to understand the functionality each cluster can provide in a particular IBM-supplied pattern or a custom deployment environment. You can allocate a specific type of function (for example, the support infrastructure function) to a particular cluster. Understanding the functions 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.

The clusters configured in a deployment environment provide the functions listed below.

The functions can exist in a single cluster, or can be spread across multiple clusters. Each of the IBM-supplied topology patterns creates a different number of clusters to support the functions. The number of clusters in your deployment environment depends on the topology pattern that you are using.

Application deployment target

An application deployment target is the set of servers (cluster) to which you install your applications (human tasks, business processes, mediations, and so forth). 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

The supporting infrastructure includes the Common Event Infrastructure (CEI) server and other infrastructure services used to support your environment and 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.

Take note that the actual business rules themselves are not tied to the Supporting infrastructure cluster. In fact, business rules can exist and work everywhere in the cell. It is the business rules administrative function (performed from the Business Rules Manager) that can be deployed on the Supporting Infrastructure cluster (in a 3 cluster configuration) or in the Web application infrastructure cluster (in the 4 cluster configuration). The same principle applies to the human tasks and business processes. The human tasks and business processes actually run on the Application deployment target cluster (as that is where the human task and business process containers are configured). However, you administer processes and tasks from the Business Process Choreographer Explorer, which can reside on the Supporting Infrastructure cluster (in a 3 cluster configuration) or on the Web application infrastructure cluster (in a 4 cluster configuration).

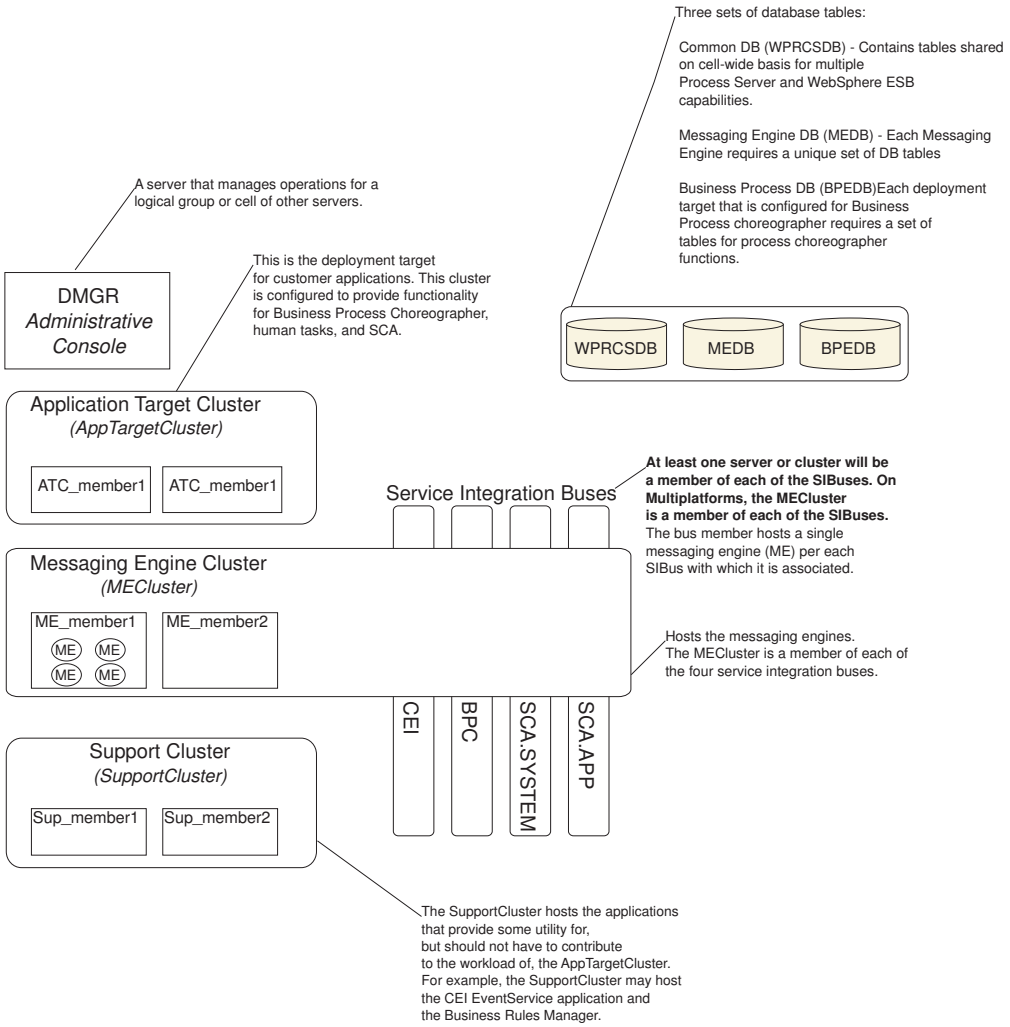
Messaging infrastructure

The messaging infrastructure is the set of servers (cluster) where the messaging engines are located and is used to provide asynchronous messaging support for your applications and for the internal messaging needs of the WebSphere Process Server components. 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.

Web Application infrastructure

Consists of a cluster where the web-based components Business Process Choreographer Explorer, Business Space and Business Rules Manager are located.

For topologies in all environments, the fundamental pieces of WebSphere Process Server and WebSphere ESB are always similar. In all WebSphere Process Server and WebSphere ESB cells, the deployment manager is the central point of administration for the cell. The following diagram illustrates the "points of interest" in a deployment environment configured using a Remote Messaging and Remote Support topology pattern. The number of clusters and the type of functions supported by the clusters will vary by pattern. For information on a specific topology pattern, see the topic that applies to that pattern.



Functions of 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.

Related concepts

"Considerations for selecting a topology" on page 80
Selecting an appropriate topology for your deployment environment depends upon several factors.

"Network deployment" on page 98
The term *Network deployment* refers to a WebSphere Process Server environment configuration comprised of a logical group of servers, on one or more machines, managed by a single deployment manager.

Overview of error prevention and recovery
The error prevention and recovery information describes how to avoid problems that might cause system failures, and provides or points to information on how to recover from system failures that can result from both ordinary and extraordinary circumstances.

Single cluster topology

One of the supplied BPM topology patterns. In a single cluster topology, all the functions of WebSphere Process Server environment are combined into a single cluster.

This is the default pattern for WebSphere Process Server for z/OS.

A Single cluster topology is ideal for limited hardware. Because all the components are installed in the same cluster, fewer physical machines are required. However, because each server instance must run the supporting applications and your integration applications, the memory requirements for the individual Java Virtual Machines (JVMs) are much greater. In addition, one or more members of the cluster must also run the messaging engines required for asynchronous interactions. Thus, Single cluster topologies are typically used for proof of concept, development, and testing environments.

On non-z/OS platforms, the single cluster topology is typically used for testing, proofs of concept, and demonstration environments. However, on z/OS a single-cluster topology is a viable production topology. This is because of the different architecture of WebSphere Process Server for z/OS. A single-cluster topology on z/OS has characteristics that are analogous to those of the non-z/OS remote messaging topology. There are operational advantages to using the single cluster topology on z/OS, such as zWLM-managed servant regions.

Combining all aspects of the WebSphere Process Server environment into a single cluster has other implications aside from the increased memory requirements. Because asynchronous interactions (involving JMS and MQ/JMS bindings), human tasks, state machines, and long-running business processes can make extensive use of the messaging infrastructure, a single cluster environment is not ideal for applications with these components. Any messaging requirements must be kept to a minimum with this pattern (except for z/OS). 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. So, the single cluster pattern is suitable for scenarios that are focused on running applications and on synchronous invocations. This topology is also not ideal if you intend to make extensive use of the Common Event Infrastructure (CEI). Generating events and CEI-related messaging traffic, places an additional burden on the cluster members.

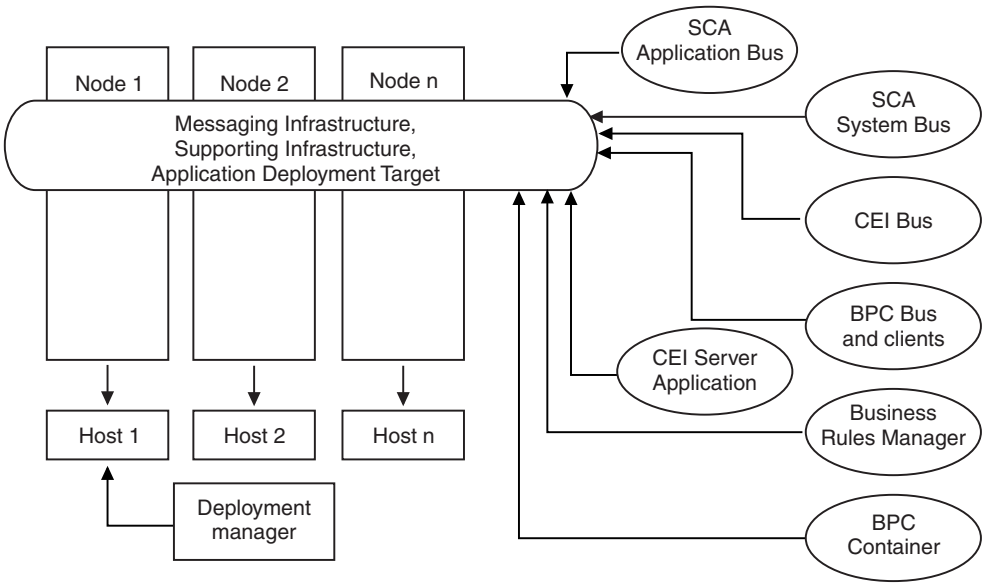
From an administrative and scalability perspective, the Single cluster topology has advantages. A single cluster where each member runs all the WebSphere Process Server components are simpler to administer. Instead of several server instances in multiple clusters, you have a single cluster with fewer members. If the needs of your environment grow, scaling the infrastructure is a simple matter of adding additional nodes and cluster members. Thus, the process of adding capability is simple, but all components are scaled at the same rate. For example, each additional cluster member adds CEI processing whether you need it or not. If you have the messaging engines spread across server members using policies, there could be some additional administrative effort in creating and maintaining the policies.

In a *Single cluster* pattern, all deployment environment functions / 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
- 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.

See the following graphical representation of Single cluster topology.



Related concepts

"Considerations for selecting a topology" on page 80
Selecting an appropriate topology for your deployment environment depends upon several factors.

Synchronous invocation
Service component interfaces (SCA) are always defined in the synchronous form. For each synchronous interface, one or more asynchronous interfaces can be generated.

Remote messaging topology

One of the supplied BPM topologies. In a *Remote messaging* pattern, the deployment environment functions are split across two separate clusters.

The remote messaging pattern provides a separate cluster for the messaging function. 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.

For environments that must support numerous human tasks, long-running business processes, state machines, and asynchronous interactions, a Remote Messaging topology has advantages over the Single Cluster topology. Separating the messaging infrastructure into a separate cluster removes the messaging overhead from the application target cluster. This reduces the memory requirements for the application target cluster members. This topology also differs from the Single Cluster topology in terms of the hardware required. Because there are now two clusters with multiple cluster members, the hardware requirements are greater for distributed environments.

From an administrative perspective, the requirements of the Remote Messaging topology are greater than those of the Single Cluster topology. Additional clusters and additional cluster members increase the administrative effort required. In addition, distributing the messaging engines across the members of the messaging cluster requires the creation and maintenance of policies.

In the Remote Messaging topology, the supporting applications and the CEI components are still part of the application target cluster. Thus, for environments that make extensive use of CEI, the Remote Messaging topology may not be ideal either. For small to medium-sized businesses, or for businesses without extensive monitoring or auditing requirements, this topology is generally suitable.

The scalability options for the Remote Messaging topology are as straightforward as the options for the Single Cluster topology. Because the messaging engines are subject to one of n policies (each messaging engine is active on only one server), adding additional members to the messaging cluster has little effect. Spreading the messaging engines across server members using policies can allow you to split the messaging burden across a maximum of three servers (the SCA.SYSTEM and SCA.APPLICATION engines should be active on the same server). Thus, adding more than three cluster members to the messaging cluster has no effect on the processing capability of the messaging infrastructure. Scaling the application target cluster is relatively easy. If you need additional processing capability for your applications or for the supporting infrastructure, you can simply add additional nodes and members to the application target cluster.

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
- Application deployment target

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

See the following graphical representation of Remote messaging cluster topology.

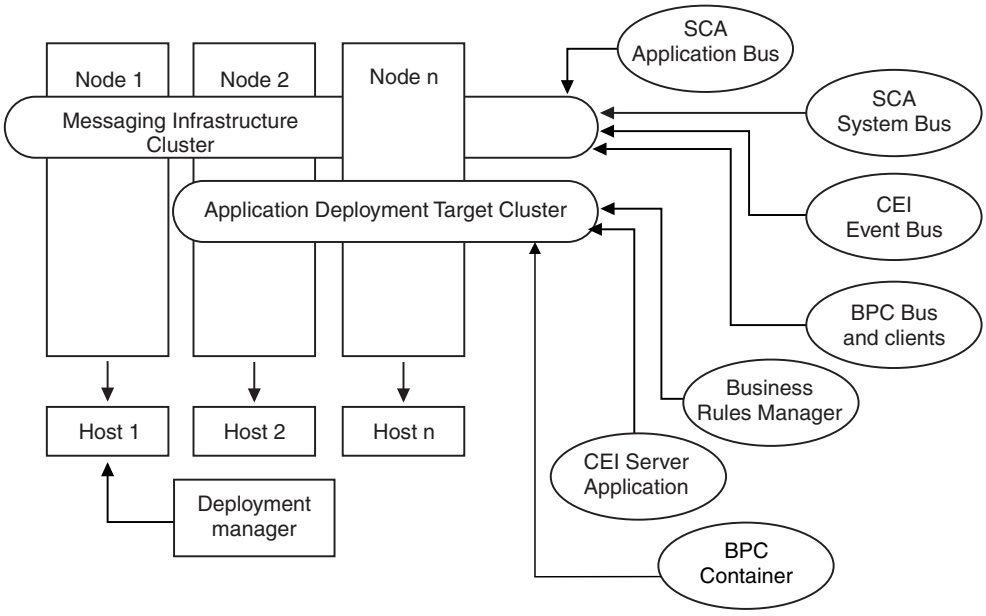


Figure 4. Remote messaging pattern

Related concepts

"Considerations for selecting a topology" on page 80
Selecting an appropriate topology for your deployment environment depends upon several factors.

Asynchronous invocation
WebSphere Process Server delivers a powerful programming model for developing asynchronous applications. With asynchronous invocation in SCA, there are three types of asynchronous interaction styles available: one way, deferred response, and request with callback. With all three types of asynchronous invocation, the client receives control back immediately from the SCA runtime upon an `invokeAsync()` call.

Remote messaging and remote support topology

One of the supplied BPM topology patterns. In a *Remote messaging and remote support* pattern the deployment environment functions are split across three separate clusters.

This is the default pattern for WebSphere Process Server for Multiplatforms. 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 (except for z/OS). The components are divided between the three clusters.

For the vast majority of customers (especially those with large computing infrastructures), the Remote Messaging and Remote Support topology is the preferred environment. The hardware requirements for distributed platforms are more intensive, but having three (or more) clusters with multiple members performing specific functions allows you greater flexibility in adjusting and tuning memory usage for the JVMs.

Creating three clusters, each with its own functions and applications, creates an additional administrative burden. As you add clusters and cluster members, your

performance tuning plan and the troubleshooting burden can expand greatly. Spreading messaging engines across the members of the messaging cluster also adds to the administrative burden associated with creating and maintaining policies.

From a scalability standpoint, the Remote Messaging and Remote Support topology provide the most flexibility. Because each of the distinct functions within WebSphere Process Server is divided among the three clusters, you can pinpoint performance bottlenecks and adjust the cluster size fairly easily. If you need additional CEI processing, you can simply add a node and cluster member to the support cluster. Similarly, if you need more processing capability for your business processes or human tasks, you can add additional nodes and members to the application target cluster. Because expanding the messaging infrastructure beyond three cluster members has no affect on processing capability, the scalability limitations present in the Remote Messaging policy also apply to the Remote Messaging and Remote Support topology.

As with the Remote Messaging topology, the Remote Messaging and Remote Support topology provide an ideal environment for long-running business processes, state machines, human tasks, and asynchronous interactions (including JMS and MQ/JMS bindings). Because the application target cluster is only responsible for running your business integration applications, performance tuning and diagnostics are much simpler than in the previous topologies where the application target cluster had additional responsibilities. The Remote Messaging and Remote Support topology is also ideal for environments that make extensive use of CEI for monitoring and auditing (including environments with WebSphere Business Monitor). Separating the support infrastructure into its own cluster provides you with a dedicated set of cluster members for CEI and for the supporting applications like BPC Explorer and Business Space.

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

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.

See the following graphical representation of Remote messaging cluster topology.

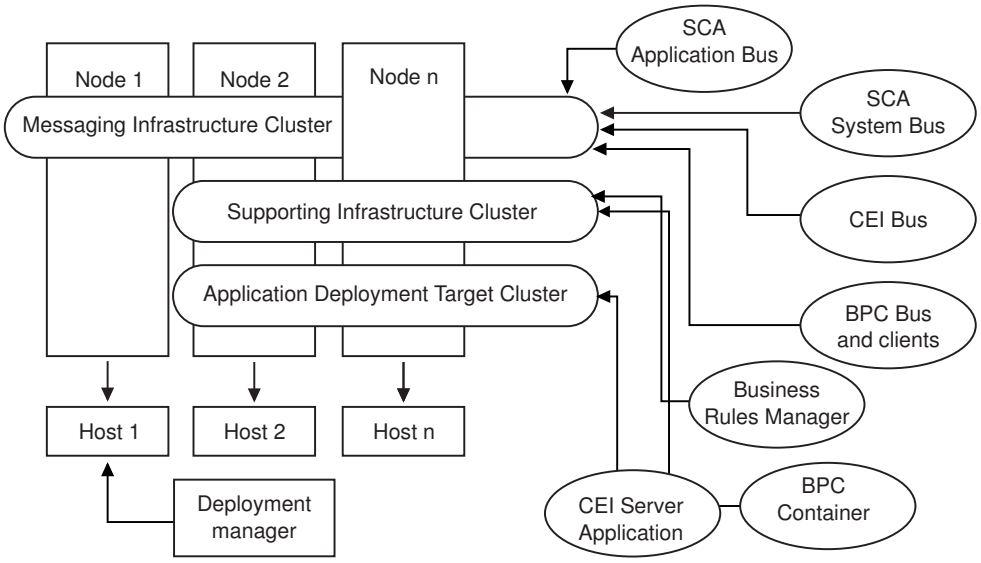


Figure 5. Remote messaging and remote support pattern

Resource allocation example

The following figure shows one way in which resources might be allocated using the remote messaging and remote support pattern. The figure shows three hosts. Host A has Server 1 and Server 3; Host B has Server 2, Server 4, and Server 5 and Host C has Server 6 and Server 7. Because the heaviest load for this installation is for application use, more resources for (Server1, Server2, and Server6) are allocated for the application deployment target cluster (Cluster 3) than the other functions.

Note: Load balancing is not available for the default configuration remote messaging and remote support pattern. That configuration uses a single messaging engine bus, while the load balancing feature requires at least two messaging engine buses.

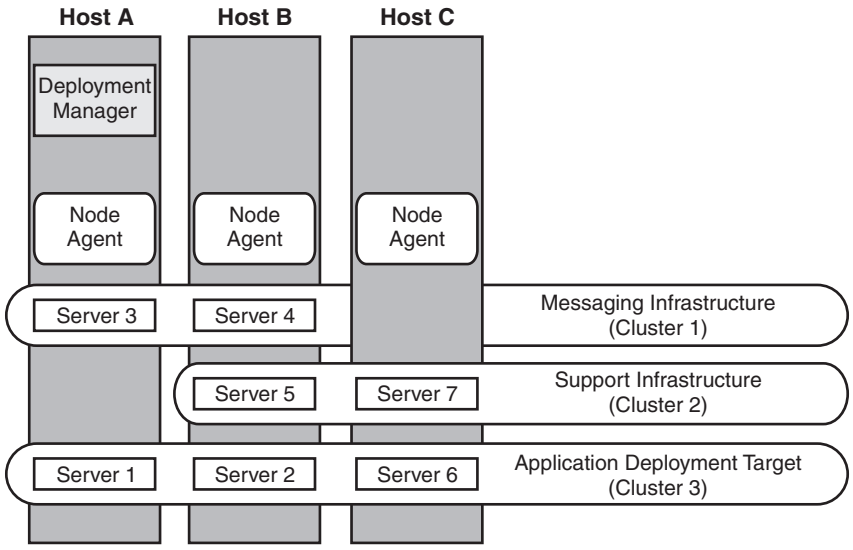


Figure 6. Resource allocation example

Related concepts

"Considerations for selecting a topology" on page 80
Selecting an appropriate topology for your deployment environment depends upon several factors.

Remote messaging, support and web applications topology

One of the patterned BPM topologies. In a *Remote messaging, support and web application* pattern the deployment environment functions are split across four separate clusters.

This pattern does not apply for an installation and deployment manager configuration for WebSphere Process Server only. It only applies if the deployment environment you are creating is for a deployment manager that has been augmented to include WebSphere Business Monitor.

This four-cluster pattern is similar to the remote messaging and remote support pattern, except that supporting Web applications reside on their own cluster.

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

Application deployment cluster:

- Application deployment target
- Business Process Choreographer container

Remote web application infrastructure cluster:

- Business Rules manager
- Business Process Choreographer Explorer
- Business Space

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

In a *Remote messaging, support and web application* pattern, the deployment environment functions are split across four separate clusters (one cluster for messaging functionality, one cluster for support functionality, one cluster for applications and one cluster for web-based functions.)

The Remote Messaging, Remote Support and Web topology pattern is the recommended starting topology when you are starting with WebSphere Dynamic Process Edition. WebSphere Dynamic Process Edition make greater use of the Web UI components. This pattern uses a fourth cluster to house the following Web applications:

- BPC Tools
- Business Rules Manager
- Business Space
- REST API Services
- AlphaBlox for dimensional view of data

Aside from giving you the ability to precisely control the individual components deployed in your environment, the advantages of the this topology pattern are similar to those in the Remote Messaging and Remote Support topology.

See the following graphical representation of a Remote messaging, support and web application topology.

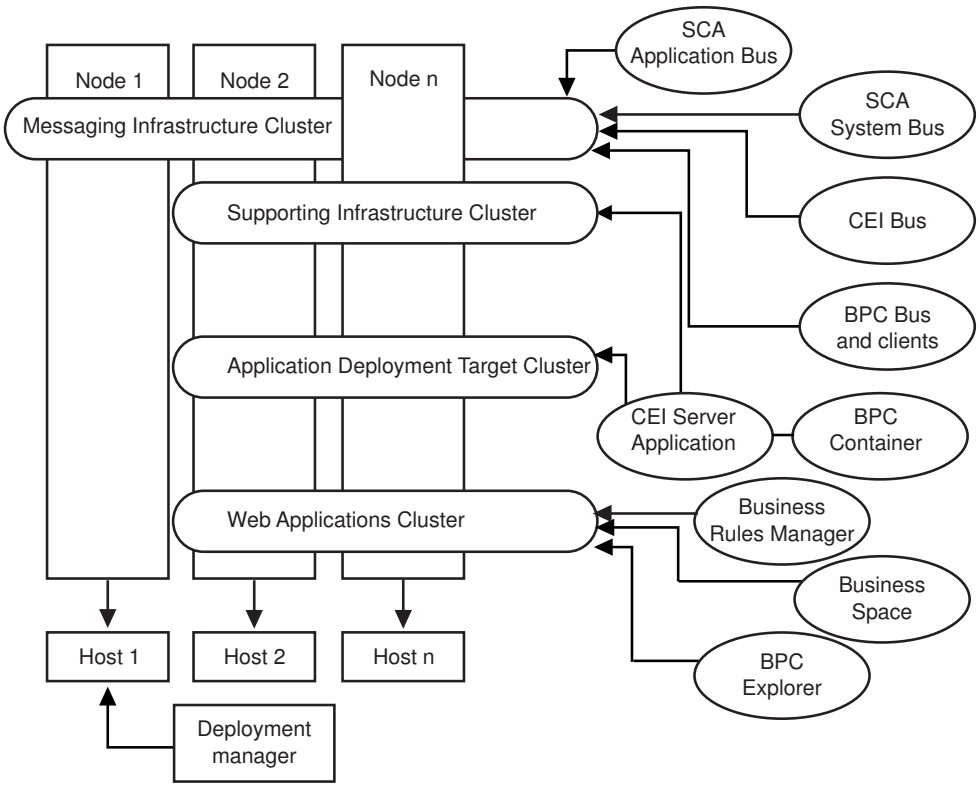


Figure 7. Remote messaging, support and web pattern

Related concepts

"Considerations for selecting a topology" on page 80
Selecting an appropriate topology for your deployment environment depends upon several factors.

Custom topology

A custom topology addresses the processing and business requirements unique to your situation. It is not a patterned topology, but rather a topology that your create and tailor to your specific needs.

By allowing you to define your own environment, the custom topology is by far the most flexible. As mentioned previously, the supplied topology patterns (Single Cluster, Remote Messaging, Remote Messaging and Remote Support and Remote Messaging Support and Web Applications), deploy all of the WebSphere Process Server components to their default locations. You may or may not need the additional overhead associated with these components. For example, if your organization has no need for the CEI, you could create a custom topology that removes CEI support and the BPC Observer from your environment. Similarly, if your organization has governance rules that prevent you from taking advantage of the Business Rules Manager, you could remove it from your deployment.

Aside from giving you the ability to precisely control the individual components deployed in your environment, the advantages of custom topologies are similar to those in the Remote Messaging and Remote Support topology. The disadvantages are also similar.

Topology patterns and supported BPM product features

The availability and default usage of topology patterns varies from one BPM product to another.

If you are using the Deployment Environment Configuration wizard on the administrative console to create the deployment environment, the availability of topology patterns on which you base your deployment environment varies depending on the following conditions and configuration decisions:

- The platform on which you have installed WebSphere Process Server
- The primary deployment environment feature and the complimentary feature

The table below shows the relationship between the topology patterns and BPM products.

Table 15. Available supplied patterns and their relationship to features

Topology pattern	Number of clusters	Description	Supported BPM products and features	Default status
Single cluster	1	The single cluster topology pattern provides one cluster for all the functional components. The user applications, messaging infrastructure, CEI, and support applications are all configured in the same cluster. Most businesses can use this pattern to support their deployment environments as it has operational advantages such as zWLM-managed servant regions, which make a single-cluster on z/OS known as a polished silver topology. It is the recommended topology on z/OS. This is the default pattern WebSphere Process Server production environments.	Supported by the following single BPM product installations: • WebSphere Process Server • WebSphere ESB • WebSphere Business Services Fabric • WebSphere Business Monitor Supported by the following multi-feature installation: • WebSphere Business Monitor + WebSphere ESB • WebSphere Business Monitor + WebSphere Process Server • WebSphere Business Monitor + WebSphere Business Services Fabric	This is the default pattern for a z/OS installation.

Table 15. Available supplied patterns and their relationship to features (continued)

Topology pattern	Number of clusters	Description	Supported BPM products and features	Default status
Remote messaging	2	The remote messaging 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.	Supported by the following single BPM product installations: • WebSphere Process Server • WebSphere ESB • WebSphere Business Services Fabric	This is not a default pattern.
Remote messaging and remote support	3	The remote messaging and remote support 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.	Supported by the following single BPM product installations: • WebSphere Process Server • WebSphere ESB • WebSphere Business Services Fabric	This is the default pattern for the following installations: • WebSphere Process Server on multiplatforms • WebSphere Enterprise Service Bus on multiplatforms • WebSphere Business Services Fabric

Table 15. Available supplied patterns and their relationship to features (continued)

Topology pattern	Number of clusters	Description	Supported BPM products and features	Default status
Remote messaging, remote support and web applications	4	This pattern does not apply for an installation and deployment manager configuration for WebSphere Process Server only. It only applies if the deployment environment you are creating is for a deployment manager that has been augmented to include WebSphere Business Monitor. This pattern defines one cluster for application deployment, one remote cluster for the messaging infrastructure, one remote cluster for supporting applications, and one for web application deployment (Business Process Choreographer Explorer, Business Space and Business Rules Manager).	Supported by the following single BPM product installations: • WebSphere Business Monitor Supported by the following multi-feature BPM installation: • WebSphere Business Monitor + WebSphere ESB • WebSphere Business Monitor + WebSphere Process Server • WebSphere Business Monitor + WebSphere Business Services Fabric	This is the default pattern for a WebSphere Business Monitor installation

Considerations for selecting a topology

Selecting an appropriate topology for your deployment environment depends upon several factors.

These factors include, but are not limited to the following:

- Available hardware resources
- Application invocation patterns
- Types of business processes that you plan to implement (interruptible versus non-interruptible)
- How heavily you intend to use the CEI
- Individual scalability requirements
- Administrative effort involved

Condensed topology selection criteria

Consider the information listed in the following table, which is a quick guide to selecting your production topology. This table provides a condensed list of the advantages and disadvantages of each of the topology patterns.

For information on which BPM products support the supplied topology patterns, see *Topology patterns and supported BPM product features*.

Table 16. Considerations for selecting a topology for your deployment environment

	Topology Pattern			
Consideration	**Single cluster**	**Remote Messaging**	**Remote Messaging and Remote Support**	**Remote Messaging, Remote Support and Web**
Number of clusters to maintain	One cluster for all components	One cluster for applications and for the support infrastructure One cluster for messaging	One cluster for applications One cluster for the support infrastructure One cluster for the support infrastructure	One cluster for applications One cluster for Web interfaces One cluster for support infrastructure One cluster for messaging
Hardware requirements	Can be implemented on limited hardware	More hardware required for distributed environments	More hardware required for distributed environments	Most hardware intensive
Asynchronous interactions	Use should be minimal	Use must be balanced against resource availability	Ideal environment for asynchronous interactions	Ideal environment for asynchronous interactions
Long-running processes, state machines, and human tasks	Use should be minimal	Use must be balanced against resource availability	Ideal environment for interruptible processes, state machines, and human tasks	Ideal environment for interruptible processes, state machines, and human tasks
Heavy CEI activity	Not recommended (Light CEI use should be balanced against resource usage.)	Not recommended (Light CEI use should be balanced against resource usage.)	Ideal environment for heavy CEI use	Ideal environment for heavy CEI use
Administrative burden	Relatively small	Requires additional effort	Requires additional administrative effort	Requires most administrative effort

Table 16. Considerations for selecting a topology for your deployment environment (continued)

Consideration	Topology Pattern			
	Single cluster	**Remote Messaging**	**Remote Messaging and Remote Support**	**Remote Messaging, Remote Support and Web**
Scalability	All components scaled at the same rate	Messaging cluster scalability limited (no benefit beyond three servers) All other components scaled at the same rate	Easy to scale All functions separated Messaging cluster scalability still limited (no benefit beyond three servers)	Easiest to scale All functions separated Messaging cluster scalability still limited (benefit comes when other BPM products are introduced)

Related concepts

"Topologies and deployment environment patterns" on page 63
There are different topology layouts. Before you install and configureWebSphere Process Server, review the information in this section. Understanding topology concepts will help you to make educated decisions on how to install and configure the product.

Invocation styles

Business process types

"Single cluster topology" on page 68
One of the supplied BPM topology patterns. In a single cluster topology, all the functions of WebSphere Process Server environment are combined into a single cluster.

"Remote messaging topology" on page 70
One of the supplied BPM topologies. In a *Remote messaging* pattern, the deployment environment functions are split across two separate clusters.

"Remote messaging and remote support topology" on page 72
One of the supplied BPM topology patterns. In a *Remote messaging and remote support* pattern the deployment environment functions are split across three separate clusters.

"Remote messaging, support and web applications topology" on page 75
One of the patterned BPM topologies. In a *Remote messaging, support and web application* pattern the deployment environment functions are split across four separate clusters.

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

Before you plan your deployment environment complete the following tasks:

- Choose a database type
- Identify available resources
- Identify necessary security authorizations

About this task

When you plan the layout of interconnected servers you must 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. Identify the functional requirements of the deployment environment
   a. Identify the features or runtime capabilities of your deployment environment

      Will the deployment environment support BPM products in addition to WebSphere Process Server?
   b. Identify the component types you will deploy.

      Consider the component types and the interactions between components as part of the requirements.
   c. 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.
   d. Identify any functional requirements not related to applications.

      Consider security servers, routers and any other hardware or software requirements to handle business events.
2. Identify the capacity and performance requirements for your environment.
3. Identify the redundancy requirements for your environment.
   a. Identify the number of servers you need for failover.
4. Design your deployment environment.

   Decide on the pattern. For WebSphere Process Server there are three established cluster patterns to choose from as follows.
   - Single cluster
   - Remote messaging
   - Remote messaging and remote support

   If none of these patterns meets your needs, you can create your own custom deployment environment.

Note: If your configuration supports BPM products in addition to, and compatible with, WebSphere Process Server, the patterns of those products would be available to you when you create your deployment environment. For example, a **Remote messaging, remote support and web** pattern is available with WebSphere Business Monitor.

See "Topology types and deployment environment patterns" for more information about the patterns and the differences amongst them.

5. Understand the methods available to you for configuring your deployment environment.

   You can configure a deployment environment for WebSphere Process Server using the following methods:

   - Creating a deployment environment using the Deployment Environment Configuration wizard from the administrative console

     You can create the Single cluster, Remote messaging, Remote messaging and remote support and (if applicable) Remote messaging, support and web clusters using the Deployment Environment Configuration wizard through the administrative console. You can also create the custom deployment environment through a Deployment Environment Configuration wizard in the administrative console or by building it yourself through the administrative console.
   - Creating a deployment environment using wsadmin
   - Creating a deployment environment at profile creation time using the Profile Management Tool (PMT)
   - Creating a deployment environment at profile creation time using the manageprofiles command-line utility

   Note: Creating a deployment environment from the installer is not supported.

What to do next

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

Related concepts

"Stand-alone server" on page 97
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 100
A deployment environment is a collection of configured clusters, servers, and middleware that collaborates to provide an environment to host Service Component Architecture (SCA) interactions. For example, a deployment environment might include a host for message destinations, a processor of business events, and administrative programs.

"Topologies and deployment environment patterns" on page 63
There are different topology layouts. Before you install and configureWebSphere Process Server, review the information in this section. Understanding topology concepts will help you to make educated decisions on how to install and configure the product.

"Clusters in deployment environments" on page 101
Clusters give your applications more capacity and higher availability than a single server.

 Service integration buses for WebSphere Process Server
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 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.

   For more information about installing WebSphere Integration Developer, see the IBM WebSphere Integration Developer Information Center.

What to do next

Install the software.

Related concepts

"Stand-alone server" on page 97
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 96
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.

The zPMT tool

Related tasks

Creating a response file using the zPMT tool

Configuring WebSphere Process Server for z/OS

Granting table privileges to the JCA authentication alias user ID

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.

   For systems using a DB2 database (including DB2 for i5/OS, DB2 for IBMi and DB2 for z/OS) you cannot automatically 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 run 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 affect 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 affect 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.

   For more information about installing WebSphere Integration Developer, see the IBM WebSphere Integration Developer Information Center.

What to do next

Install the software.

Related concepts

"Stand-alone server" on page 97
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 96
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.

The zPMT tool

DB2 decisions

Related tasks

"Determining the products to install" on page 11
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 9
Identify your assets to make the best use of those resources already available and also to make informed purchasing decisions.

Creating a response file using the zPMT tool

Configuring WebSphere Process Server for z/OS

Granting table privileges to the JCA authentication alias user ID

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.

   For systems using a DB2 database (including DB2 for i5/OS, DB2 for IBMi and DB2 for z/OS) you cannot automatically 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 run 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 affect 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 affect 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 a **development stand-alone server profile** 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.

   Note: The **development stand-alone server profile** is for testing purposes only, it cannot be used in a production environment.

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 or use different profiles or, on z/OS, use different logical partitions to accomplish the sharing.

   For information on WebSphere Process Server for z/OS partitioned data set planning, see the WebSphere Process Server for z/OS documentation.

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 96
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 managers" on page 99
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 101
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 in deployment environments" on page 101
Clusters give your applications more capacity and higher availability than a single server.

"Topologies and deployment environment patterns" on page 63
There are different topology layouts. Before you install and configureWebSphere Process Server, review the information in this section. Understanding topology concepts will help you to make educated decisions on how to install and configure the product.

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

"Planning for interoperability between WebSphere Process Server and other WebSphere Application Server products" on page 11
When analyzing your software environment, you need to know whether requests can pass between the various software levels that exist in your deployment environment.

"Error handling strategy and solution recovery" on page 125
WebSphere Process Server has error handling capabilities and tools that you can use for recovery purposes.

"Recovery in a production environment" on page 128
In the production environment, the goal is to process all the requests that have entered the system in a methodical and consistent manner. Data preservation is required for this environment and all measures must be taken to minimize system unavailability and data loss.

Related tasks

"Planning your deployment environment" on page 83
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.

"Identifying available resources" on page 9
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 11
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.

Granting table privileges to the JCA authentication alias user ID

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.

   For systems using a DB2 database (including DB2 for i5/OS, DB2 for IBMi and DB2 for z/OS) you cannot automatically 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 run 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 affect 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 affect 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.

    For more information about installing WebSphere Integration Developer, see the IBM WebSphere Integration Developer Information Center.

What to do next

Install your deployment environment.

Related concepts

"Servers" on page 96
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 managers" on page 99
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 101
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 in deployment environments" on page 101
Clusters give your applications more capacity and higher availability than a single server.

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

"Planning for interoperability between WebSphere Process Server and other WebSphere Application Server products" on page 11
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

"Planning your deployment environment" on page 83
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.

"Identifying available resources" on page 9
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 11
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.

Granting table privileges to the JCA authentication alias user ID

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 on WebSphere Process Server systems: 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 WebSphere Customization Tools application, 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 the z/OS platform, all 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 of 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 is installed. For example, `installation_file_system_root/AppServer/profiles/default`.

Related concepts

"Stand-alone server" on page 97
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 managers" on page 99
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 99
A *managed node* is a node that is federated to a deployment manager and contains a node agent and can contain managed servers. 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 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 95
A profile defines a unique runtime environment, with separate command files, configuration files, and log files. Profiles define three different types of environments on WebSphere Process Server systems: stand-alone server, deployment manager, and managed node.

Messaging engines

Data sources

Service integration buses for WebSphere Process Server
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

The term *Network deployment* refers to a WebSphere Process Server environment configuration comprised of a logical group of servers, on one or more machines, managed by a single deployment manager.

Network deployment provides the capacity, scalability, and robustness that is generally required of a production environment. In network deployment, a group of servers can collaborate 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 concepts

"Topologies and deployment environment patterns" on page 63
There are different topology layouts. Before you install and configureWebSphere Process Server, review the information in this section. Understanding topology concepts will help you to make educated decisions on how to install and configure the product.

Related information

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

Deployment managers

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

"Using multiple platforms within a cell" on page 61
With careful planning, you can create a deployment manager cell that includes nodes on both distributed and z/OS operating system platforms.

"Profiles" on page 95
A profile defines a unique runtime environment, with separate command files, configuration files, and log files. Profiles define three different types of environments on WebSphere Process Server systems: stand-alone server, deployment manager, and managed node.

Managed nodes

A *managed node* is a node that is federated to a deployment manager and contains a node agent and can contain managed servers. 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.

A managed node has a node agent that manages all servers on a node.

When a node is federated, a node agent process is created automatically. 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

"Using multiple platforms within a cell" on page 61
With careful planning, you can create a deployment manager cell that includes nodes on both distributed and z/OS operating system platforms.

"Profiles" on page 95
A profile defines a unique runtime environment, with separate command files, configuration files, and log files. Profiles define three different types of environments on WebSphere Process Server systems: 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 makes up 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*.

When to create the deployment environment

Install the software on the host systems that you intend to use. Use the Profile Management Tool or manageprofiles command-line utility to create deployment manager and custom profiles. Then create the deployment environment using the administrative console of the deployment manager.

You can still manage some aspects of the deployment environment using the administrative console. (For example, you can add more nodes to the deployment environment).

Related tasks


Creating deployment environments using the command line
You can use wsadmin to create a deployment environment. The createDeploymentEnvDef and generateDeploymentEnv provide a command-line equivalent to creating the deployment environment using the deployment environment wizard.

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 in deployment environments

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.

Choosing your deployment environment pattern

You can configure your deployment environment by choosing one of the IBM-supplied topology patterns or by creating your own custom deployment environment. This topic section lists and describes the available IBM-supplied topology patterns and presents considerations for choosing a topology.

Before you begin

You should familiarize yourself with the information in:

- Assessing your business requirements
- Identifying available resources
- Review *Considerations for selecting a topology*

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

   See Topology patterns and supported BPM product features for information on the available deployment environment patterns.
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

"Topologies and deployment environment patterns" on page 63
There are different topology layouts. Before you install and configureWebSphere Process Server, review the information in this section. Understanding topology concepts will help you to make educated decisions on how to install and configure the product.

"Considerations for selecting a topology" on page 80
Selecting an appropriate topology for your deployment environment depends upon several factors.

"Error handling strategy and solution recovery" on page 125
WebSphere Process Server has error handling capabilities and tools that you can use for recovery purposes.

General steps for implementing a deployment environment

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

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.
- 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. Optional: Design the database configuation

   If you choose to, you can design the database configuration using the database design tool (DDT). Designing the database configuration before profile creation time can result in simplifying the profile creation process. If you design the database configuration early in the configuration process, you can import the database design document at profile creation time.

   Whether or not you choose to use the DDT as a way to create your database configuration is a design decision that you need to work out with members of your solution implementation team.
3. Create the deployment manager.

WebSphere Process Server provides you with multiple ways to create the deployment manager, including using the Profile Management Tool or the manageprofiles command-line utility. The method that you select for creating the deployment manager profile is a matter of preference. Each method is documented in the section on *Creating profiles*.

4. Start the deployment manager.
5. Create as many managed nodes as you need.
6. Federate the nodes from step 5 to the deployment manager created in step 3 on page 105.
7. Configure the cell.

   You can use the Deployment Environment Configuration wizard to configure the cell. You can use the wizard to create a deployment environment based on patterns. Deployment environment patterns are rules-based configurations of the most commonly used business integration topologies. A pattern provides a template for an environment configuration. Because the deployment environment patterns represent well-known, tested and recommended topologies with component configurations that work together; using patterns ensures reliable deployment environment functionality.

   Important: If you are using a script to create the deployment environment, 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 "Java Management Extensions connector properties" in the WebSphere Application Server information center for information on SOAP request timeout. To change the default timeout value, open the file `$WAS_HOME/profiles/<profile name>/properties/soap.client.props` in any ASCII text editor and find the following line (shown here with default value of 180 seconds):

   ```
   com.ibm.SOAP.requestTimeout=180
   ```

   If you need to change the default you can edit this line to set the timeout to a value more appropriate for your situation. Setting the above value to 0 will disable the timeout check altogether.

   Configuring the cell 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.
8. 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. These tables are in the common database. See Creating the DB2 databases and storage groups using SPUFI, DSNTEP2, or DButility.sh and Ddl2Pds.sh script for more information.

b. Configure the messaging engine database tables. These tables are 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.

   If you are using the Business Process Choreographer Explorer reporting function, you also need to configure the Business Process Choreographer Explorer reporting database (OBSRVDB).

d. Create the enterprise service bus logging mediation database table. These tables are in the common database.

e. Configure the Common Event Infrastructure database.

9. Create a proxy server in WebSphere Application Server. The proxy server routes HTTP requests to content servers that perform the work.

   You can use other routing servers in place of, or in front of the proxy server, for example IBM HTTP Server. The benefit of using the proxy server is that it is integrated with WebSphere Application Server and therefore easy to use and maintain.

   Attention: The proxy server (or an alternate routing server) is required when you are load balancing HTTP requests between two or more cluster members. This server allows clients to access the applications within this topology.

10. 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

"Topologies and deployment environment patterns" on page 63
There are different topology layouts. Before you install and configureWebSphere Process Server, review the information in this section. Understanding topology concepts will help you to make educated decisions on how to install and configure the product.

"Deployment environments" on page 100
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 in deployment environments" on page 101
Clusters give your applications more capacity and higher availability than a single server.

"Servers" on page 96
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 managers" on page 99
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.

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

Creating and configuring the databases

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

Related information

Configuring Business Process Choreographer

Creating a proxy server

Communicating with Web servers

Installing IBM HTTP server

Messaging engines

Using the wsadmin scripting tool

- Managing node agents
- Starting clusters
- Stopping clusters
- Java Management Extensions connector properties

Planning for WebSphere Process Server: Additional considerations

Middleware, such as WebSphere Process Server, requires that you evaluate many aspects of your enterprise information system (EIS), such as capacity and security before installing the product. Also, you must plan ahead before using the capabilities of WebSphere Process Server, like error prevention and recovery.

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

- 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 require for the installation?
- Must you have 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 require handle increased load, or handle more concurrent users in the future? Will you have to add additional resources in the future to meet additional demands?
- Will your system have to dynamically add or remove resources to handle daily fluctuations in demand?
- Does your system have 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.

Planning error prevention and recovery

You can develop error-prevention and recovery strategies to minimize the impact of system and application errors.

Topics in *Planning error prevention and recovery* include links to a variety of resources, such as information center topics, technical articles and IBM Redbooks® that provide detailed information on development processes and system configuration patterns designed to take advantage of WebSphere system recovery capabilities.

Overview of error prevention and recovery

The error prevention and recovery information describes how to avoid problems that might cause system failures, and provides or points to information on how to recover from system failures that can result from both ordinary and extraordinary circumstances.

WebSphere Process Server is a middleware server optimized for enabling the running and management of business process management (BPM) and service-oriented architecture (SOA) solutions. WebSphere Process Server is built on the foundational capabilities of WebSphere Application Server.

Middleware systems run under various conditions, not all of which are traditionally "good path" conditions. Many of the key features within WebSphere Process Server are intended to deal with the uncertainty that might arise through what can appear to be normal operations.

Assumptions and expectations

Before using the information about system failure and recovery as described in the *Planning error prevention and recovery* section, read the following list of assumptions:

- You are familiar with WebSphere Process Server and the basic architectural principles upon which it is built and the basic kinds of applications that it runs.
- You have a foundational understanding of integration projects, including how to plan for and implement integration projects.
- Unless otherwise specified, the information about system failure and recovery is relevant to version 6.1.0 and later of WebSphere Process Server.

  Note: The information contained in the *Planning error prevention and recovery* section assume a remote messaging and remote support pattern, which consists of three separate clusters, one for the WebSphere Process Server and one each for the messaging engine and CEI event server.

Related concepts

"Topologies and deployment environment patterns" on page 63
There are different topology layouts. Before you install and configureWebSphere Process Server, review the information in this section. Understanding topology concepts will help you to make educated decisions on how to install and configure the product.

Related reference

"Peer recovery" on page 132
Peer recovery is recovery as performed by another member of the same cluster, and can be initiated either manually or automatically. Peer recovery processing (either automated peer recovery or manual peer recovery) is tightly intertwined with the WebSphere's high availability environment.

Planning error prevention

As with all IT endeavors, planning against and practicing for extreme situations will increase the possibility for a successful recovery.

There are a number of required considerations associated with preparing for system and application recovery. These considerations can be grouped under two categories as follows:

- Error prevention practices as part of application design
- Error prevention practices as part of development process

Error prevention as part of application design

Including error prevention practices as part of your application design means implementing specific design techniques and using the capabilities of the product to help prevent system and application errors.

A strong system of governance, complete with architectural and design guidelines and appropriate standards combined with reviews and checkpoints are essential to building the right kind of application.

Error prevention practices as part of application design include the following:

- Implementing design considerations for exceptions and faults
- Implementing an error handling strategy that uses existing WebSphere Process Server error handling capabilities and tools
- Creating connectivity groups and using module application design techniques

Connectivity groups

A connectivity group represents a specific pattern of behavior found in an SCA module.

Create connectivity groups to represent the possible request sources for the system.

In a connectivity group you:

- Put all the logic to get the inbound data into one module

  This is also true for outbound data when it is going to an external system or legacy system
- Put all the logic to connect and transform the data into one module

  All the other modules can now use a standard set of interfaces and not have to worry about extra transformations.

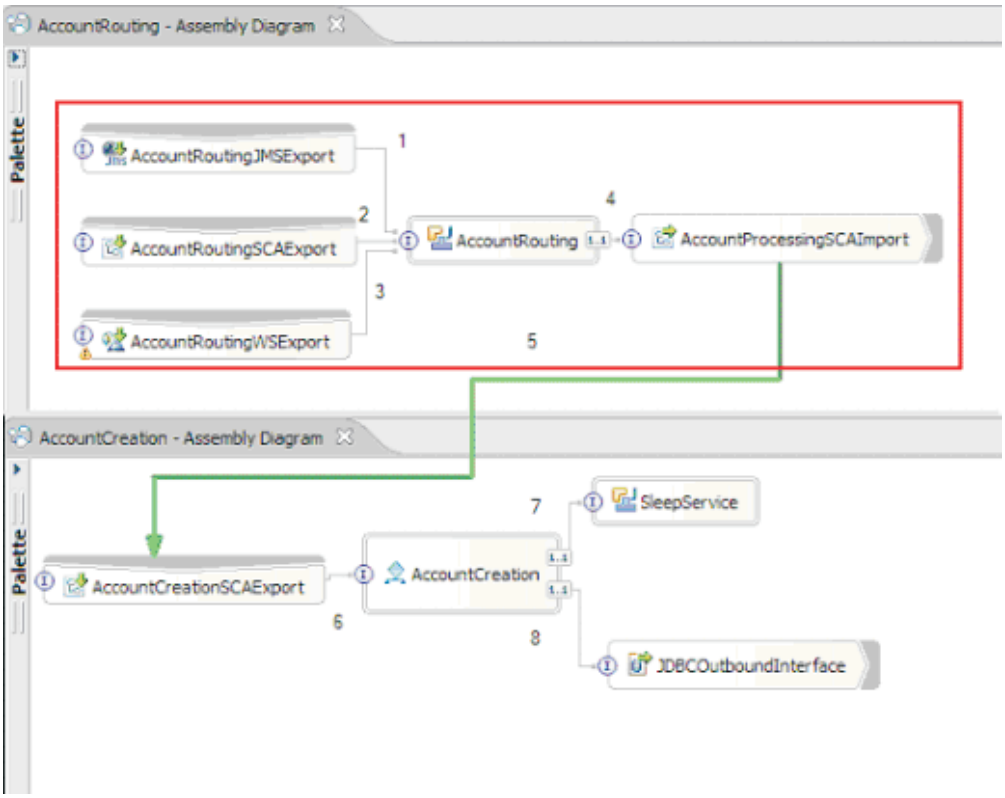
The connectivity group will not contain stateful component types like long-running business processes and Business State Machines. These connectivity groups provide encapsulation and isolation of the specific endpoint's integration requirements. Commonly, WebSphere ESB mediation modules are used for this purpose as they represent convenient ways implement "infrastructure" related tasks.

The concept of connectivity groups also provide a convenient way to quiesce the system in case there is a need for recovery. Because the connectivity group module is stateless, the module can be temporarily stopped thus cutting off the inbound flow of new events while the system finishes processing the events it has.

Note: If you want to stop the flow of inbound events, then the connectivity modules **should not** support inbound and outbound in the same module (even though the same EIS system may have both inbound and outbound). If inbound and outbound support are in the same module, then the outbound is turned off with the inbound. This may cause internal work to stop from completing. Consider separating inbound and outbound in this case.

When the system is recovered and able to process new work, these modules can be restarted.

The module that is outlined in the following screen capture is considered part of a connectivity group.



Connectivity groups can be used for input from an external source or an existing system such as SAP or CICS®. Or for new work from a web browser-based clients.

Related concepts

Use case: recovering data from failed events
A use case provides a context for a recovery scenario. In the use case, a business has an application that receives a request to create a new Account.

Related reference

"Export bindings" on page 134
To completely quiesce a system, consider the different types of request invocations supported by the available Export bindings.

Application design considerations for exceptions and faults

You need to consider your application design so that it can take advantage of the error handling and fault processing capabilities in WebSphere Process Server.

In order to create a comprehensive error handling strategy, solution architects need to understand how WebSphere Process Server and WebSphere ESB represent declared and undeclared exceptions.

The SCA programming model provides two types of exceptions:

- Service Business Exceptions

  Service Business Exceptions are checked exceptions declared in a business method's function signature (WSDL faults or Java throws). Service Business Exceptions identify error conditions that are anticipated by the application or service. These exceptions are sometimes referred to as "checked exceptions"

  An example is an `InvalidSymbolException` for a stock quote service. Such exceptions are wrapped by ServiceBusinessException and passed back to the client.
- Service Runtime Exceptions

  Also known as "system exceptions" service runtime exceptions are not declared in the method signature. In general, they represent error conditions that are not anticipated by the application, such as a `NullPointerException` in a Java Component.

  These exceptions are wrapped by `ServiceRuntimeException` and passed back to the client, which can interrogate the `ServiceRuntimeException` to determine the cause.

  Note: When working at the SCA level these exceptions are sometimes referred to as faults. However, when using Java code they are usually referred to as exceptions.

  When a `ServiceRuntimeException` is thrown from a component, the current transaction will be rolled back.

Service Business Exception handling:

Service Business Exceptions represent known and declared exceptions anticipated by the application or service.

Service Business Exceptions are defined on the service interface.

Component developers should take care to declare the possible exceptions that may be thrown, so that the consuming service can handle them. For example, a business fault to a banking application would include "Invalid Account Number", or "Insufficient Funds" as *business exceptions*. So the application that calls the service needs to include logic to handle a situation where they have passed in an invalid

account number, or where they tried to transfer $100 but there was only $50 in the account. These are the types of business errors that a calling application is designed to handle. The WebSphere Process Server business exceptions are returned to the client to catch and handle appropriately.

When handling business service exceptions, service consumers should implement the client such that it will perform one of the following actions for a declared business exception:

1. Catch the exception and create the appropriate Service Business Exception for the calling application.

   This could mean including the original exception in the new exception (wrapping it). This is most often done when the calling module does not have the same Business Exceptions as the service that it is calling. Here is an example of the flow catching an exception and creating a Service Business Exception for the calling application:

   a. `Module A` has SBE "MoneyTransferFailed"
   b. `Module B` has SBE "InsufficientFunds"
   c. `Module A` calls `Module B` and gets "InsufficientFunds" exception
   d. `Module A` must create a new exception "MoneyTransferFailed", which may have a place where a string defining the original error of insufficient funds can be included.
2. Catch the exception and perform alternate logic.

Related concepts

Use case: recovering data from failed events
A use case provides a context for a recovery scenario. In the use case, a business has an application that receives a request to create a new Account.

Service Runtime Exception handling:

Service Runtime Exceptions are undeclared exceptions. In general, they represent error conditions that are not anticipated by the application.

Service Runtime Exceptions are used to signal an unexpected condition in the runtime.

Component developers can handle Service Runtime Exceptions in the following ways:

1. Catch them and perform some alternative logic.

   For example, if one partner is not able to service a request perhaps another one might.
2. Catch the exception and "re-throw" it to your client.
3. Remap the exception to a business exception.

   For example, a timeout for a partner may result in a business exception that indicates most of the request was processed but there was one piece of the request that was not completed and should be retried later or tried with different parameters.

If an exception is not caught, the exception is passed on to the component that called the current component. This call chain continues back to the original caller in the chain. For example, `Module A` calls `Module B` and `Module B` calls `Module C` and

then Module C throws an exception, Module B might or might not catch the exception. If Module B does not catch the exception, then the exception travels back to Module A.

When a ServiceRuntimeException is thrown from a component, the current transaction will be rolled back. This type of exception processing is repeated for all components in the chain. For example, if a ServiceRuntimeException is thrown from Module C, that transaction will be marked for rollback. Then the exception is thrown to Module B, where if it is not caught and another transaction is present, that transaction also will be rolled back. Component developers can use quality of service (QoS) qualifiers to control whether invocations occur in the current transaction or a new transaction. So, if Module A calls Module B and Module B is part of a new transaction, then Module A can "catch" a ServiceRuntimeException from Module B and continue processing, without Module A's transaction rolling back.

Note: Because runtime exceptions are not declared as part of the interface, component developers should attempt to resolve the exception and thus prevent a runtime exception from inadvertently being propagated to the client if the client is a user interface.

You should be aware that the contents of the rolled back transaction can vary, depending on the nature of the transaction. For example, long-running BPEL processes can be segmented into many smaller transactions. Asynchronous request and response calls are broken out of a transaction automatically (otherwise the calling application might have to wait a long time for the response).

In instances where a transaction is broken into multiple asynchronous calls (as opposed to one large transaction), the initial work for the transaction would rollback at the occurrence of a ServiceRuntimeException. However, the response from the asynchronous call is sent from a different transaction, and because the response from the asynchronous call would have no place to go, an event is created in the Failed Event Manager (FEM).

The following list is of 4 current subclasses of ServiceRuntimeException:

1. ServiceExpirationRuntimeException

   This exception is used to indicate that an asynchronous SCA message has expired. Expiration times can be set using the RequestExpiration qualifier on a service reference.

2. ServiceTimeoutRuntimeException

   This exception is used to indicate that the response to an asynchronous request was not received within the configured period of time. Expiration times can be set using the ResponseExpiration qualifier on a service reference.

3. ServiceUnavailableException

   This exception is used to indicate that there was an exception thrown while invoking an external service via an import.

4. ServiceUnwiredReferenceRuntimeException

   This exception is used to indicate that the service reference on the component is not wired correctly.

Related concepts

Use case: recovering data from failed events
A use case provides a context for a recovery scenario. In the use case, a business has an application that receives a request to create a new Account.

Related information

Setting qualifiers and transactions

Error prevention as part of development

You can include error prevention processes as part of your development processes.

Error prevention practices as part of your development process are intended to focus on the governance and development process that is in place for rolling out projects and mainly involves testing, tuning, measuring and retesting activities.

Error prevention practices as part of your development process can include the following:

- Preventing problems through comprehensive testing
- Continual and regularly scheduled environment tuning
- Infrastructure monitoring

Error prevention: Comprehensive testing

You can prevent problems that will require recovery by implementing a comprehensive functional and system test plan.

In general, tests for deployed solutions can be categorized as follows:

- Functional test

  Functional tests confirm that the functionality implemented in an application meets the stated business requirements. Functional tests are created by business users and application designers.
- System test

  System tests are designed to verify performance, high availability and recovery service level agreements.

  In a system test, it is important to combine aspects like performance testing and high availability testing to evaluate the recovery of a system in extreme production situations.

For both functional and system testing, automation is strongly recommended. Automated testing provides the organization with an efficient way to prevent regressions bugs from being introduced.

Related concepts

Recovery: First steps
Administrators can facilitate solution recovery processes by following a first steps checklist of general practices.

Related information

Problem determination in WebSphere Process Server

Error prevention: Environment tuning

Tuning exercises are a regular part of the system development life cycle. With each major application deployment you should schedule a performance evaluation.

As a prerequisite to deploying a solution to a production environment, you should evaluate and test the solution in a preproduction environment. This will allow you to measure the impact of the new solution to existing applications and the current system parameters and resources. Failure to evaluate and test the solution in a preproduction environment increases the likelihood that the solution will have issues with recovery.

There are many publicly available resources that describe the process and execution of performance test plans. Review the materials and construct a test plan that is right for your application and topology.

Consult IBM Redbooks that contain information on WebSphere Process Server performance and tuning, as well as technical white papers on WebSphere Process Server performance and tuning. Also, you should consult the performance reports that accompany each new release of the business process management (BPM) and Connectivity products from IBM.

Related information

Tuning

 IBM WebSphere Business Process Management Performance Tuning

 Endurance testing with WebSphere Process Server

 WebSphere Business Integration V6.0.2 Performance Tuning

 Performance Tuning Automatic Business Processes for Production Scenarios with DB2

 WebSphere Process Server V6 – Business Process Choreographer Performance Tuning of Human Workflows Using Materialized Views

Error prevention: Infrastructure monitoring

Infrastructure monitoring and the use of infrastructure monitoring tools is a requirement for a production system.

Monitoring tools like *ITCAM for SOA* and *Tivoli® Performance Viewer* allow the system administrators to monitor critical system behavior and to detect problems that may cause an outage.

A basic level of IT monitoring for the production system is essential to meeting availability service level agreements.

For more information on monitoring the performance and business processes of your service component events, see the section on Monitoring in the WebSphere Process Server information center.

Related information

Monitoring

IBM Tivoli Composite Application Manager Family for SOA:

You can use IBM Tivoli Composite Application Manager Family (ITCAM) for service-oriented architecture (SOA) to monitor WebSphere Process Server. In addition, you can use ITCAM for SOA to automate problem mediation and manage solution configuration and deployment.

ITCAM for SOA includes the following features:

Manage SOA services

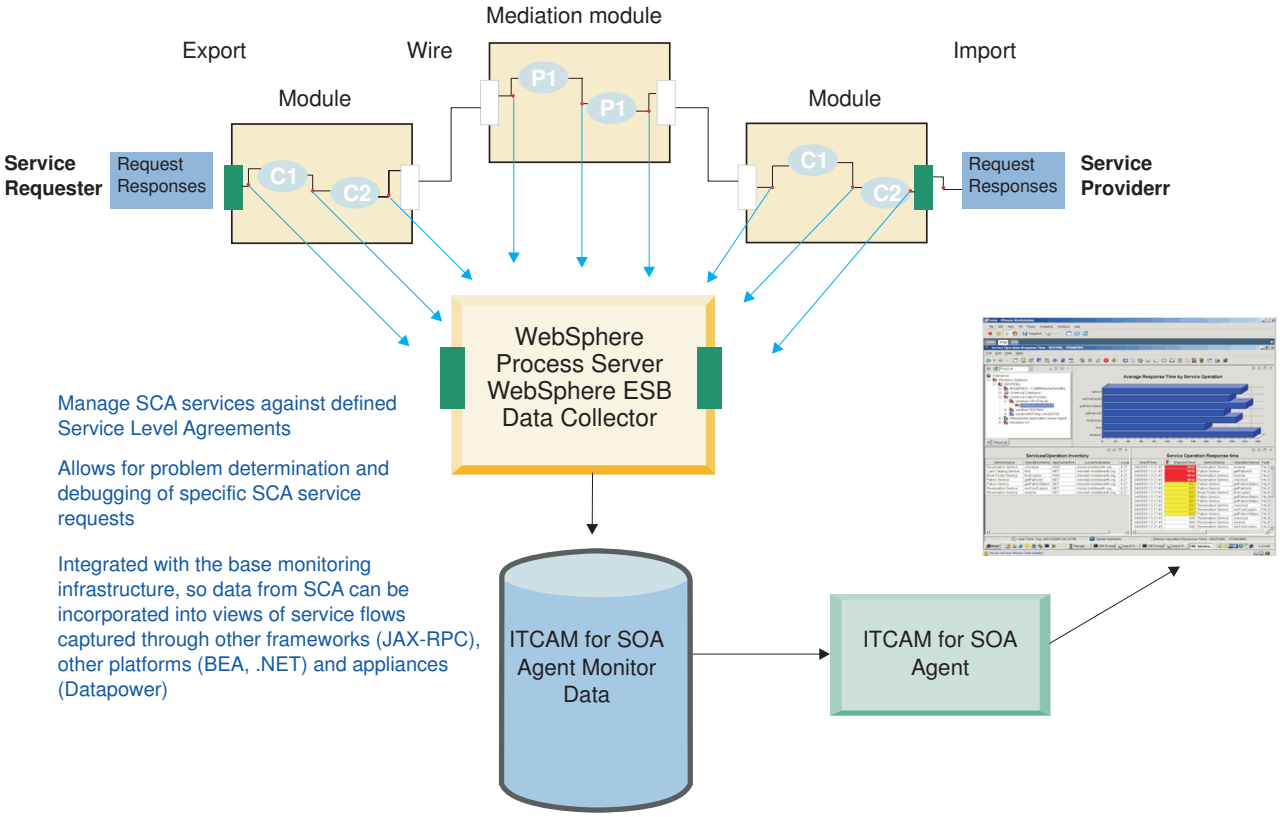
- Visibility into SOA service interactions
- Visibility into message content and transaction flow patterns
- Ability to identify and isolate performance bottlenecks across technology and platform boundaries
- Lightweight, industry standard Application Response Measurement (ARM)-based Performance instrumentation
- High performing and flexible enforcement of policies
- Standards-based Instrumentation for simple integration

Monitor business processes

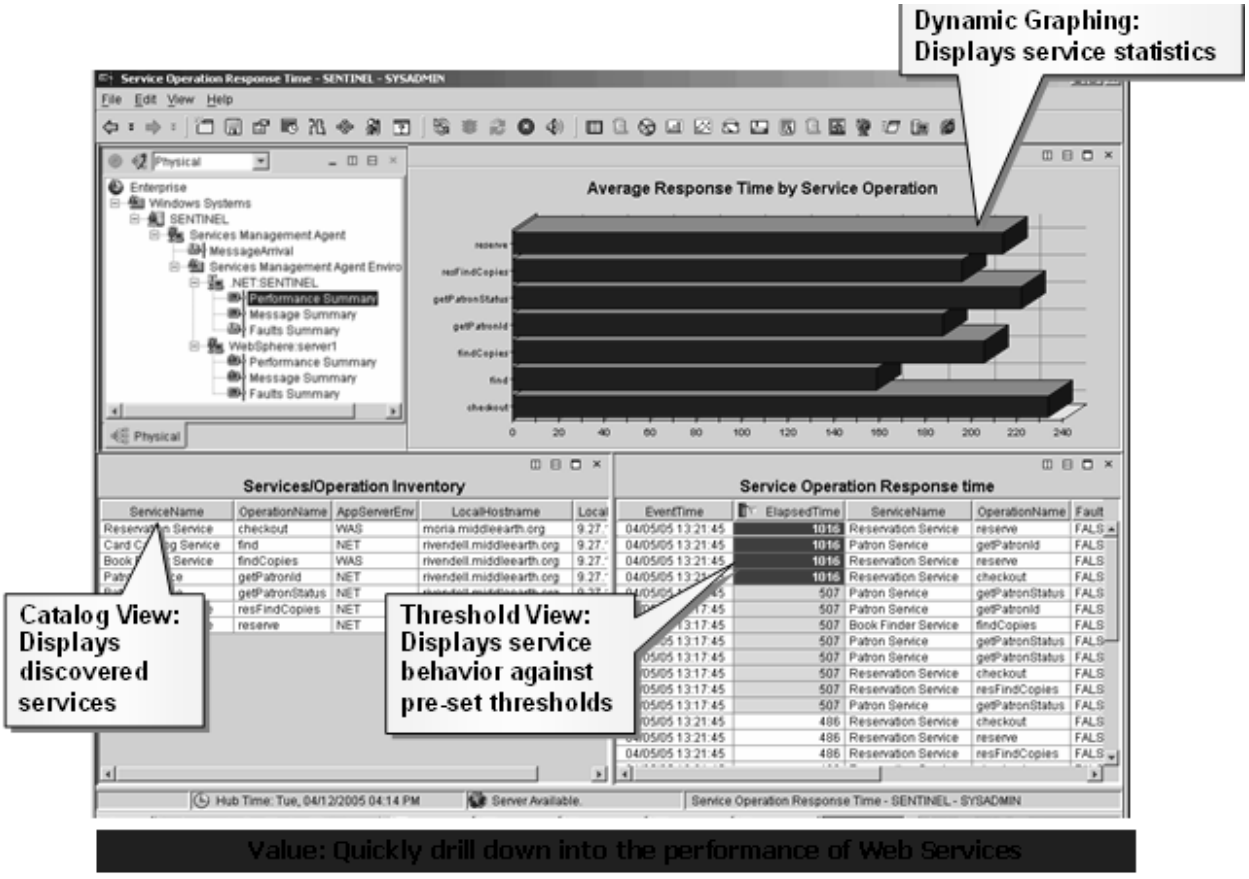
- Manage in flight processes
- Monitor the Business Performance of active processes
- Detect Business Situations and take action
- Gather Business Intelligence from collected process data
- Comprehensive deep-dive monitoring to identify and quickly fix down or slow performing applications
- Real-time metrics and historical data analysis

IBM Tivoli Composite Application Manager Family (ITCAM) for SOA examples

The following example shows how IBM Tivoli Composite Application Manager Family (ITCAM) for SOA monitors services, response times, message counts, and message sizes.

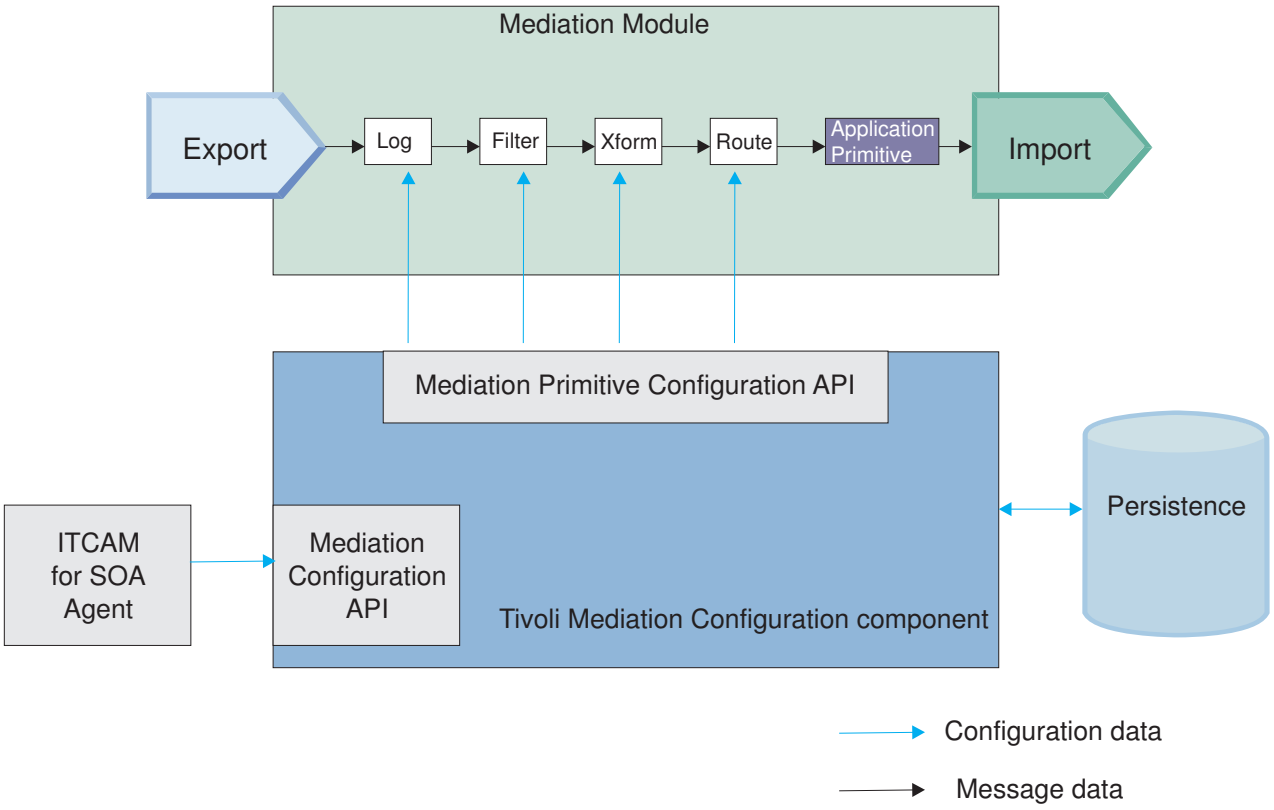


The following example shows a screen measuring statistics per operation and thresholds that can be set to detect problems as provided by IBM Tivoli Composite Application Manager Family (ITCAM) for SOA.



As stated previously, IBM Tivoli Composite Application Manager Family (ITCAM) for SOA includes special functions to work with WebSphere ESB to dynamically modify mediation flow configuration.

The following diagram illustrates the mediation flow configuration capabilities provided by IBM Tivoli Composite Application Manager Family (ITCAM) for SOA.



Related information

IBM Tivoli Composite Application Manager Family Installation, Configuration, and Basic Usage

Problem-determination methodology documentation

Establish a well-articulated and clear problem determination methodology for the solution that you will deploy to your production environment.

This means maintaining a document of your problem determination methods and practicing the documented methods on a consistent basis.

It is recommended that you document your solution-specific problem determination methodology in an operations manual. This operations manual should contain the following types of information pertaining to solution-specific problem determination:

- An established format for recording observations during problem determination

  By using an established format you can achieve consistency in recording your observations. Excel spreadsheets are common “observational reporting tools".
- List the trace information

  You should include the following trace information for solution-specific problem determination:
  - A list of the traces to enable
  - A list of the servers on which you will enable traces
  - A description of the conditions in which you will enable traces

  Before implementing a trace, make sure that the trace will not make matters worse. It is not appropriate to “enable everything”. Take care when enabling

trace, as trace specifications should be appropriate for the observed condition. Use intelligent situational analysis to collect the correct diagnostic information. If you are unsure how to implement the correct level of tracing, contact IBM support.

- Enabling verbose garbage collection (verbosgc)

  Verbose garbage collection (GC) data provides extensive details about how the GC is running for a specific application. This can be useful for analyzing performance problems and tuning the GC settings for the application.
- Generating a heap dump

  Heap dump capability is a feature of the IBM JVM that prints a record of all objects in the Java heap to a text file.

  The size and address of each object are recorded, as well as the addresses of all objects that it references. This information can help you understand which objects are responsible for taking up large amounts of memory.
- Creating Java.cores

  Performing problem determination by analyzing the javacore file is an effective means of determining root cause of error conditions that might occur in an IBM Java virtual machine (JVM).
- Where and what logs need to be collected before opening a problem management record (PMR). Define the proper usage of IBM's “must gather” scripts.
- Gathering version information (versionInfo) such that all the maintenance package information is included
- Database-specific procedures for gathering logs and information that is recorded by the database as various problems arise

Treat your solution-specific problem determination documentation as a *living document* and maintain and update it as often as new observational practices are learned from functional and system test.

Note: You should become familiar with and use IBM Support Assistant and other tools that can be leveraged for problem determination and problem reporting. Collection of the aforementioned information should be a prerequisite for opening any new PMR as the inclusion of this data will significantly reduce the PMR cycle times.

Related tasks

Enabling Cross-Component Trace with data snapshot using the administrative console
Enable Cross-Component tracing with data snapshot to collect data associated with Service Component Architecture (SCA) processing and call chain data associated with WebSphere Process Server and WebSphere Enterprise Service Bus modules and components.

Related information

Locating and analyzing heap dumps

Product version and history information

Enabling verbose garbage collection (verbosegc) in WebSphere Application Server

IBM Support Assistant

Software currency

Software currency is the practice of maintaining the latest software for the deployed solution.

It is important to maintain software currency for the deployed solution.

IBM creates regular fix packs to aid in the application of Authorized Program Analysis Reports (APARs) found in the product base. The service package does contain mandatory code changes. For more information, refer to the published list of APAR fixes.

Related information

Contacting IBM Software Support

Getting fixes

WebSphere Process Server Support

Error handling strategy and solution recovery

WebSphere Process Server has error handling capabilities and tools that you can use for recovery purposes.

The architecture team that is building the solution must understand how to use WebSphere Process Server tools and capabilities for error handling and recovery.

The architecture team is responsible for creating the error handling standards that the application development team must adhere to.

The error handling strategy for the project must account for the following:

- Appropriate usage of Units of Work (Transactions and Activity Sessions)
- Declaration and usage of faults and ServiceBusinessExceptions
- Consistent fault processing for all component types, especially BPEL and Mediation Flow Components
- Usage of retry logic and "continue on error" Business Process Choreographer capabilities
- Appropriate settings for completed process instance deletion
- Correct usage of synchronous and asynchronous invocation patterns

- Appropriate usage of Import and Export types.
- Proper usage of the retry capability in mediation flows

In addition to the above, the architecture team must create design patterns where the built-in recovery capabilities,of WebSphere Process Server are leveraged appropriately; for example, recovering from a failed event manager.

Related concepts

Fault handling in business processes
When a fault occurs in a process, the navigation continues with the fault handler or fault link.

Related tasks

"Choosing your deployment environment pattern" on page 102
You can configure your deployment environment by choosing one of the IBM-supplied topology patterns or by creating your own custom deployment environment. This topic section lists and describes the available IBM-supplied topology patterns and presents considerations for choosing a topology.

"Planning for a deployment environment based on one of the supplied patterns" on page 88
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.

Related information

Fault handling and compensation handling in business processes

Stable-environment maintenance

There are several additional steps you can take to achieve a stable environment and reduce the likelihood of system and application failures.

The following sections discuss measures that your infrastructure team can leverage to reduce the number of manual processes that can affect solution stability and system recovery.

Automated environment creation

A scripted framework lends itself to consistency when creating an environment.

All actions that you can run from the administrative console can also be run using a script. There are existing IBM service assets that should be used and customized for your specific needs. These scripts can then be maintained with each tuning exercise. In many cases when working in a test environment, you will often have to re-create the environment. A script is the most efficient way of implementing repeated actions, such as creating a test environment. The test system script can then modified for use to create the production system.

Discuss automated deployment with your IBM Software Services for WebSphere (ISSW) representative or build on similar procedures that are being leveraged in your WebSphere Application Server production environments

Related information

Using scripting (wsadmin)

Commands and scripts

Automated application deployment

Use automated scripts to assist in the deployment of an application or solution groups to the proper environment.

A well-designed "build, package, and deploy" model has numerous benefits, including improved developer productivity, reduced turnaround time for builds and code fixes, better consistency in application code, and reinforcement of development policies

Automated scripts used to deploy applications or solutions groups will complement the automated process for creating your environment.

An automated application deployment that uses scripts will reduce the manual intervention with the environments and will also reduce the chances of human error on redeployment or recovery.

Discuss automated deployment with your IBM Software Services for WebSphere (ISSW) representative or build on similar procedures that are being leveraged in your WebSphere Application Server production environments.

Related information

Administering applications using wsadmin scripting

Jython script library

Planning a recovery strategy

Planning a recovery strategy will increase the possibility for a successful recovery.

High availability

High availability (HA) pertains to the ability of IT services to withstand all outages and continue providing processing capability according to some predefined service level.

One of the most important things you can do to facilitate solution recovery is to configure your system for High Availability (HA). Covered outages include both planned events, such as maintenance and backups, and unplanned events, such as software failures, hardware failures, power failures, and disasters. Clustered environments are highly available by nature because a clustered system is re-configured when a node or daemon failure occurs, so that workloads can be redistributed to the remaining nodes in the cluster.

A highly available solution is made up of a combination of hardware, software, and services that fully automate the recovery process and does not disrupt user activity. HA solutions must provide an immediate recovery point with a fast recovery time.

In a highly available solution, when the application server detects a problem, the transaction and the related data are moved to another server (either within the

same data center or, in the case of a disaster, to a server in another geographic location) automatically. Moving the transaction and related data to another server is known as *peer recovery*.

Related reference

"Peer recovery" on page 132
Peer recovery is recovery as performed by another member of the same cluster, and can be initiated either manually or automatically. Peer recovery processing (either automated peer recovery or manual peer recovery) is tightly intertwined with the WebSphere's high availability environment.

Related information

Learning about high availability and workload sharing

Recovery environments and objectives

The recovery spectrum can span test and production environments as well as different recovery objectives (system recovery and application recovery). Recovery goals and objectives vary depending on the environment from which you want to recover.

Related concepts

"Transactional properties and solution recovery" on page 130
WebSphere Process Server is based on WebSphere Application Server and as such, supports a *transactional model* conducting business transactions.

Related information

Transactional high availability

Asynchronous replication of WebSphere Process Server and WebSphere Enterprise Service Bus for disaster recovery environments

Recovery in a production environment

In the production environment, the goal is to process all the requests that have entered the system in a methodical and consistent manner. Data preservation is required for this environment and all measures must be taken to minimize system unavailability and data loss.

Consider the following aspects of a production environment:

- Topology type

  You need understand which topology type best suits your the production environment. Finding the topology that is right for you requires an analysis of application properties and of nonfunctional requirements.

  For more information on topology types, see the topic *Planning for a deployment environment based on one of the supplied patterns* in the WebSphere Process Server information center.
- Understanding and insight into the condition from which you need to recover

  For example, if a cluster has multiple cluster members, it is possible that the only thing that needs to be recovered is a single-cluster member and that the workload management machinery has already redirected work to 'running servers'. If this is the case, restarting servers should force recovery and that server should join back into the cluster again.

  Some High Availability (HA) configurations have the ability to recover failed transactions from one server on another (known as peer recovery).

Recovery of production data requires success on two levels; system and application.

Related tasks

"Planning for a deployment environment based on one of the supplied patterns" on page 88
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.

Recovery in a test environment

The goal and the number of assumptions for a test environment are different than that of a production environment.

In the test environment, the goal would be to recover the system such that new tests can be conducted as soon as possible. Data preservation is not required and it is assumed that all the requests in the system can be discarded.

Note: This is not the same as a "recovery" test. Recovery tests would leverage the recommendations provided for the production scenarios and should be conducted during the System Test Phase of the project.

System recovery

System recovery refers to the operations performed (either manually or automatically) to correct the negative conditions that affect the infrastructure of the solution.

WebSphere Process Server solutions rely on fundamental infrastructure requirements. WebSphere Process Server can be affected in a negative way by any of the following conditions:

- Power outage
- Loss of network
- Database failure
- Hardware failure

If any of these types of interruptions occur, you must address and correct them before WebSphere Process Server system recovery.

As soon as you address any failures or interruptions in the fundamental infrastructure requirement, WebSphere Process Server will rely on the inherited capabilities from WebSphere to begin application recovery.

Related concepts

Recovery: First steps
Administrators can facilitate solution recovery processes by following a first steps checklist of general practices.

Application recovery

Application recovery refers to the recovery and resolution of inflight business transactions.

If your system suffers a failure, (a power outage for example), there would be many active transactions affected, and all of them at various stages of the process flow. The system handles these transactions as part of its recovery process.

In order to have a completely successful application recovery, the applications themselves must observe the preventive practices provided.

If the applications are not developed with best practices, with recovery and transaction scope in mind, then application recovery will likely not be completely successful.

A poorly designed or "untuned" system or application will inevitably leave a percentage of inflight transactions or processes that remain unresolved after the rest of the application starts processing new events. This statement is true for not only WebSphere Process Server but for all Java EE applications and application servers.

Note: The phrase "untuned" refers to a solution that uses default settings for all components without regard to performance considerations or error handling practices.

Unresolved events can come in different forms such as processes that stay in a running state or failed events that cannot be resubmitted. A post recovery analysis of these events is required to determine what changes are necessary within the application for a full recovery. These changes should be found during the execution of the comprehensive functional and system test plan.

Related concepts

Recovery: First steps
Administrators can facilitate solution recovery processes by following a first steps checklist of general practices.

Related information

Managing failed events

Transactional properties and solution recovery

WebSphere Process Server is based on WebSphere Application Server and as such, supports a *transactional model* conducting business transactions.

WebSphere Process Server builds on this transactional model, providing for loosely-coupled SOA applications and BPM applications.

Technically, this means two things:

1. WebSphere Process Server relies on databases and messaging systems to achieve transactional application execution patterns.
2. Transactions are incumbent in messaging systems and database systems.

   Transactions are compliant with ACID properties. Transactions are considered to be ACID-compliant when they include atomicity, consistency, isolation, and durability.

   WebSphere Process Server uses databases and messaging systems to achieve a “loosely-coupled” pattern. WebSphere Process Server updates a database and sends a message. Both the update to the database and the message are committed in the same transaction.

   Another characteristic of a “loosely-coupled” pattern is to pull a message from a messaging system and update databases. If there is a failure during this processing, the event goes back to the message queue as though it had not been read. WebSphere Process Server has a retry mechanism, in which after 5 tries, the event goes to the Failed event manager. The phrase “loosely-coupled” refers to the fact that all work does not have to happen in one big transaction.

Avoiding lost data in the event of system failures

With proper tuning and configuration of the available resource managers, no data is lost if there is a failure of a given part of the system. Transactional integrity, including rollback and recovery mechanisms, are the key components in WebSphere that ensure data is not lost if failures occur.

In order for WebSphere rollback and recovery mechanisms to work, you need to set up the resource managers (database and messaging) properly. For example, lock time-outs in databases must be set properly, so that when a server recovers, it can complete either a commit or a rollback without encountering lock conditions.

WebSphere Process Server adds additional capabilities that augment those of WebSphere Application Server, to provide a complete solution for recovering data from unexpected failures.

High-level description of enabling recovery features

The core recovery model for WebSphere Process Server is based on units of work. The system can handle and recover from failures that occur during system operations centered on a single unit of work being accomplished, providing uninterrupted service. This type of recovery occurs through a series of retry mechanisms and error queues. Part of your application design should include the capability to differentiate system errors from application errors. System errors are passed back to the infrastructure supporting the calling component, where additional system level recovery can be attempted or a transformation into a more generic business exception can occur. You can configure various retry mechanisms to run automatically. Additionally, WebSphere Process Server provides a set of consoles and corresponding programming interfaces that enable more human intervention where appropriate. Many of these capabilities and the failures that they deal with can be leveraged while the server that contains the work continues processing new requests.

Unavailable server - High-level description

If a failure causes one or more servers in a highly-available WebSphere cluster to become unavailable, additional recovery capabilities within the system are called upon as follows:

1. Inbound work is routed away from the failing system

   This is done using underlying WebSphere Application Server workload management facilities, which can vary based on protocol, topology and configuration.
2. Administrator initiates actions

   While the system as a whole remains active and available, the administrator can perform recovery operations.

   Administrator actions are aimed at performing basic triage and then restarting the failing server. This restart replays transactions logs and should clean up most server down situations.

   The use of the error handling mechanisms provided by WebSphere Process Server is sometimes required to administer a complete recovery.

Unavailable cluster - High-level description

If an entire server cluster becomes unavailable or unresponsive, then a more involved set of recovery actions are necessary. For example, if a shared resource such as a database becomes unavailable, then all servers in a cluster have the same difficulties completing the work.

Procedures that deal with shared resource recovery depend on which shared resource suffered the failure. You can apply various WebSphere techniques to minimize overall downtime and restart stalled work.

Catastrophic failure - High-level description

In catastrophic situations, entire machines can become unavailable or servers deemed not recoverable. In such cases, you can rely on the advanced features in WebSphere for recovery of a server's failures to be run on another server in the same cluster. Through the use of this feature and the prerequisite of having network-attached storage or some other mechanism to share logs, this kind of recovery is also possible. For more information about recovery of a failed server by another member of the same cluster, see "Peer recovery."

Related concepts

Use case: recovering data from failed events
A use case provides a context for a recovery scenario. In the use case, a business has an application that receives a request to create a new Account.

Administering servers

"Recovery environments and objectives" on page 128
The recovery spectrum can span test and production environments as well as different recovery objectives (system recovery and application recovery). Recovery goals and objectives vary depending on the environment from which you want to recover.

Related reference

"Peer recovery"
Peer recovery is recovery as performed by another member of the same cluster, and can be initiated either manually or automatically. Peer recovery processing (either automated peer recovery or manual peer recovery) is tightly intertwined with the WebSphere's high availability environment.

Related information

Working with events

Transactional behavior of business processes

Compensation handling in business processes

Peer recovery

Peer recovery is recovery as performed by another member of the same cluster, and can be initiated either manually or automatically. Peer recovery processing (either automated peer recovery or manual peer recovery) is tightly intertwined with the WebSphere's high availability environment.

High availability manager

WebSphere employs a High Availability Manager component to monitor services provided by the application server. These services include messaging, transaction managers, workload management controllers and other application servers in a

cluster. The High Availability Manager component also makes use of Network-attached storage (NAS) devices to store transaction logs from each application server in the cluster.

The High Availability Manager is responsible for the automatic peer recovery of both *indoubt* and *inflight* transactions for any server that fails in the defined HA cluster. An indoubt transaction is any transaction that becomes stuck in the indoubt state indefinitely, because of an exceptional circumstance such as the removal of a node causing messaging engines to be destroyed. An *indoubt transaction* state results after the database finishes its phase 1 commit processing and before it starts phase 2. An inflight transaction is a transaction that has not yet completed the "prepare phase" of the commit process and where the transaction or message is persisted somewhere where it can be recovered. The automatic recovery functionality performed by the High Availability Manager enables the cluster to rebalance itself if one or more cluster members fail.

Automated peer recovery vs. manual peer recovery

Automated peer recovery is the default style of peer recovery initiation. If an application server fails, WebSphere Application Server automatically selects a server to perform peer recovery processing on its behalf. Apart from enabling high availability and configuring the recovery log location for each cluster member, no additional WebSphere Application Server configuration steps are required to use this model.

Manual peer recovery is a particular style of peer recovery that must be explicitly configured. If an application server fails, the operator can use the administrative console to select a server to perform recovery processing on its behalf.

Peer recovery reference information

The article titled IBM WebSphere Developer Technical Journal: Transactional high availability and deployment considerations in WebSphere Application Server V6 discusses the requirements, setup, and management of both automated and manual peer recovery.

Additional documentation can be found in the WebSphere Application Server information center and in the WebSphere Application Server V6 Scalability and Performance Handbook.

- WebSphere Application Server V6 Scalability and Performance Handbook
- Configuring transaction properties for peer recovery in the WebSphere Application Server information center.
- Managing manual peer recovery of the transaction service in the WebSphere Application Server information center.

Related concepts

Triggers for recovery
The need for solution recovery can result from a variety of triggers.

"Transactional properties and solution recovery" on page 130
WebSphere Process Server is based on WebSphere Application Server and as such, supports a *transactional model* conducting business transactions.

"Overview of error prevention and recovery" on page 113
The error prevention and recovery information describes how to avoid problems that might cause system failures, and provides or points to information on how to recover from system failures that can result from both ordinary and extraordinary circumstances.

Triggers for recovery
The need for solution recovery can result from a variety of triggers.

"High availability" on page 127
High availability (HA) pertains to the ability of IT services to withstand all outages and continue providing processing capability according to some predefined service level.

Export bindings

To completely quiesce a system, consider the different types of request invocations supported by the available Export bindings.

SCA invocation pattern

The following tables represents the type of SCA invocation pattern used for the different Export bindings.

Table 17. EIS export binding and associated invocation patterns

Export binding	Operation type	Performance attributes and interaction style	Invocation style
EIS	one-way	Asynchronous	asynchronous (default)
		Synchronous	synchronous
	request-response	any value	synchronous

Table 18. Export bindings and associated operation types and invocation styles

Export binding	Operation type	Invocation style
EIS	one-way OR request-response	synchronous
MQ or MQ JMS	one-way	asynchronous
SCA JMS	one-way request-response	asynchronous asynchronous with callback
Web services (soap/http) or (soap/jms)	one-way OR request-response	synchronous

Depending on the application and the topology used, a variety of techniques maybe used to quiesce synchronous communication. It is strongly recommended

that the project create a quiescing strategy based on the unique characteristics of the export used and the topology.

Related concepts

"Connectivity groups" on page 114
A connectivity group represents a specific pattern of behavior found in an SCA module.

Related information

Exports and export bindings

Working with exports

Bindings

About the failed event manager

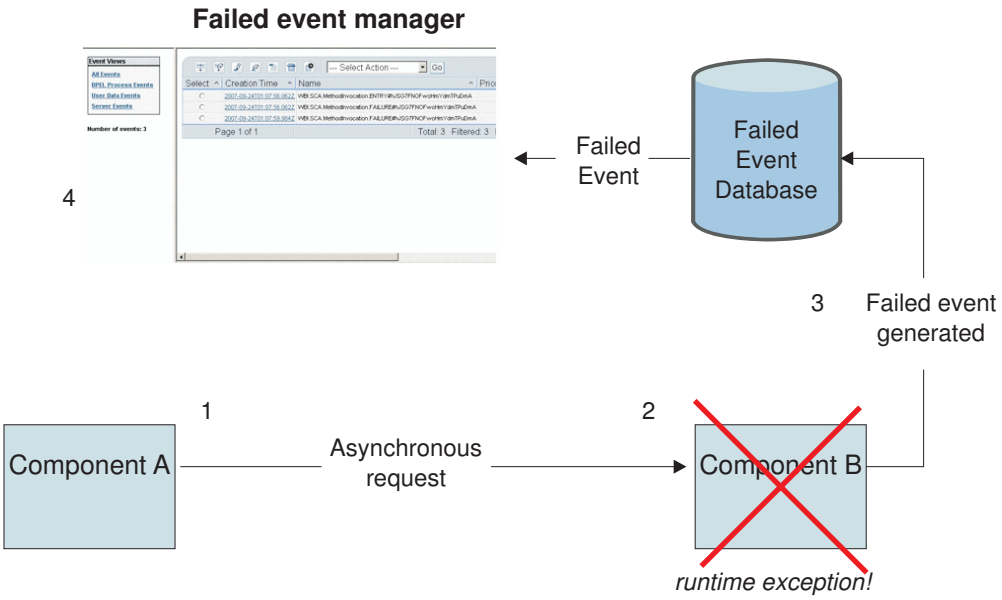
The failed event manager is a web-based client for working with and resubmitting the failed invocations.

The failed event manager is an integration application and is available in the Administrative console.

It displays the number of failed events and provides a number of search capabilities.

You can query for failed events using a variety of criteria such as date, last successful or failed event, by exception text or a combination of these.

The following illustration provides a high-level description WebSphere Process Server exception processing and its relationship to the failed event manager. Explanations of the numbered steps follow the illustration.



1. Component A calls component B in an asynchronous manner
2. Component B encounters a runtime exception and a failed event record is generated

3. The failure recovery service captures this failure and stores it in the failed event database
4. The system administrator opens the failed event manager to investigate the problem

About resubmitting failed events through the failed event manager

Events matching the search criteria entered in the failed event manager are displayed. You can resubmit single or multiple failed events. While resubmitting, you can also change the payload. For instance, the failure could have been caused by passing in some inappropriate data. In this case, the payload can be updated from within the failed events manager and resubmitted. Only the data stored in memory would be updated, so the original source of the data will not be corrected. If a resubmitted event fails, this will show up as a new failed event in the failed event manager. There is also the ability to delete single or multiple events and this is often the appropriate action due to data becoming invalid since the time of the failure.

Related concepts

Use case: recovering data from failed events
A use case provides a context for a recovery scenario. In the use case, a business has an application that receives a request to create a new Account.

Related information

Managing failed events

Resubmitting failed events

Coexisting

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

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

Coexistence support

Coexistence support gives you the ability to install, configure, run, and manage more than one product on your system.

You can have the following products installed together on your system:

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

Be aware of the following points when setting up your system for coexistence:

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

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

  You can have previous versions of the product on your system at the same time. Previous versions of the product can exist in the same LPAR or in separate LPARs.

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

  The following installation scenarios for coexistence are supported:

  – WebSphere Application Server for z/OS, WebSphere Process Server and WebSphere ESB coexisting where none of the products are currently installed.

  – Coexisting WebSphere Process Server for z/OS and WebSphere Application Server for z/OS when WebSphere Application Server for z/OS already exists

  – Coexisting WebSphere Process Server for z/OS and WebSphere Process Server for z/OS configured as an ESB-only server in an existing WebSphere Application Server for z/OS

Printed in USA