WebSphere® Process Server for Multiplatforms

IBM

Version 6.2.0

Planning the Installation

WebSphere® Process Server for Multiplatforms

IBM

Version 6.2.0

Planning the Installation

Note

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

24 April 2009

This edition applies to version 6, release 2, modification 0 of WebSphere Process Server for Multiplatforms (product number 5724-L01) 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

Chapter 1. Introduction: Planning for WebSphere Process Server

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 should plan ahead before exploiting 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 needs:

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

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

Chapter 2. Determining your software needs

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

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

Note: For the latest information on 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 your business requirements

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

Before you begin

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

About this task

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

Procedure

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

   The process might be strictly linear, or might contain loops, detours and workarounds. Draw a rough diagram and indicate connections and interactions. For each section of your product life-cycle, analyze the procedures used to move the product forward.
   - Does the procedure use paper forms and notations, is it computerized or both?
   - If it is computerized, what software is used? What hardware?
   - Are there any logjams in the process? Is there any confusion? For example, handwriting may be difficult and time-consuming to decipher; staff may have difficulty in mastering required computer skills.
   - What areas in this process are running smoothly? What are the strengths?
2. Determine how the sections of the diagram you created in step 1 perform their tasks.

- Do any of the sections use the same software? Hardware? Forms?
- If sections use different software, how well do the applications communicate with each other, if at all?
- Does each section interact only with the sections immediately preceding and following it, or does it also detour to a section in a different stage of the cycle? If so, why? Does this cause confusion or delay?
- If there is an existing intranet that sections use for communication, do some sections bypass this intranet? If so, why? Does the intranet have a history of delays or downtime that impacts other processes?
- What areas of interaction are running smoothly? In which areas are the bottlenecks? How serious are they?

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

What to do next

Identify your available resources.

Identifying available resources

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

Before you begin

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

About this task

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

Procedure

1. Itemize each piece of physical hardware. Note:
   - Amount of installed memory
   - Number and type of installed microprocessors
   - External media

   - Whether a particular unit can be upgraded
2. Itemize the currently installed software and database applications. Note:
   - Function
   - Breadth of use across the company
   - Security requirements
3. List your current IT personnel and note whether you have available the means for installation and maintenance of WebSphere Process Server, as well as the required expertise in database management. Make sure all involved have user IDs with the appropriate authorizations to successfully install all products and files.

Related concepts

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

Development and deployment version levels

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

WebSphere Process Server version 6.2 and WebSphere Integration Developer version 6.2 are compatible with prior releases as follows:

- Deployment from WebSphere Integration Developer version 6.0.2.x or 6.1 to WebSphere Process Server 6.2 is supported.
  - Applications authored and generated using WebSphere Integration Developer 6.0.2.x or 6.1 can be published to WebSphere Process Server 6.2 servers.
  - Applications authored in, generated in, and exported from WebSphere Integration Developer 6.0.2.x or 6.1 can be installed on WebSphere Process Server 6.2 servers.

  Note: For version 6.0.1 WebSphere Adapters, some additional steps might be required for compatibility. Please check the product technotes at WebSphere Process Server technotes for more information.
- Running WebSphere Process Server 6.2 artifacts on WebSphere Process Server 6.0.2.x or 6.1 is *not* supported.
  - Applications authored with WebSphere Integration Developer 6.2 cannot be published to or installed on WebSphere Process Server 6.0.2.x or 6.1 (any prior release) servers. Such content will not run correctly on WebSphere Process Server 6.0.2.x or 6.1, and changes in code generation will cause the applications to not run correctly on WebSphere Process Server 6.0.2.x or 6.1.
  - Applications authored with WebSphere Integration Developer 6.0.2.x or 6.1 and generated in WebSphere Integration Developer 6.2 cannot be published to or installed on WebSphere Process Server 6.0.2.x or 6.1 servers. Changes in code generation will cause the applications to not run correctly on WebSphere Process Server 6.0.2.x or 6.1
  - Applications generated using serviceDeploy from a WebSphere Process Server 6.2 server cannot be installed on a WebSphere Process Server 6.0.2.x or 6.1

server. Changes in code generation will cause the applications to not run correctly on WebSphere Process Server 6.0.2.x or 6.1

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
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 tasks

Installing fix packs and refresh packs with the Update Installer
You can use the IBM® Update Installer for WebSphere Software to install interim fixes, fix packs, and refresh packs collectively known as maintenance packages. The Update Installer for WebSphere Software is also known as the update installer program, the UpdateInstaller program, and the Update Installation Wizard.

Related information

Interoperating (WebSphere Application Server)

Determining the products to install

The design of your deployment environment includes determining how many and which type of software products you may require. Based on your needs the product requirements may vary among the computer 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 computers that will host the various components of the deployment environment.

   Important: When you are running multiple servers on the same computer system, you must install the software that provides the most functionality you need for any server running on that computer 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 computers to reduce costs.
   - Create multiple server instances of the same type on higher-capacity computers rather than separate instances on separate computers so the number of instances remains the same but the software needed decreases.
   - Decide whether you need unique computers 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
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
Clusters give your applications more capacity and higher availability than a single server.

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.

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

Planning for the

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

About this task

If you will use the , you can either configure it when you configure , or you can do it later.

Procedure

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

   The .
   Before version 6.2, this feature was available as the Business Process Choreographer Observer. Since version 6.2, this feature is integrated in the , and is available on the **Reports** tab. You must configure the 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 .

   The reporting database.
   The event collector and 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 doc/bpc/c2observer.dita.

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 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 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 , perform doc/bpc/t1pl_database_observer.dita.
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 doc/bpc/t2config_con_console.dita.
      - Using the interactive setupEventCollector tool. For more information about this option, see doc/bpc/t2config_con_tool.dita.
      - 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 for a high-performance system, because bpeconfig.jacl will configure the event collector and applications on the same deployment target as the Business Process Choreographer configuration. For more information about this option, see doc/bpc/t2co_bpeconfig.dita.
        You cannot use bpeconfig.jacl to configure the event collector in interactive mode.
   c. Plan the data source:
      - If the shares the same physical database as Business Process Choreographer, plan to use a separate data source for the , 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 .
      - 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 doc/bpc/r_observer_parameters.dita.

6. For each that you configure, plan the following:
   - Decide how you will configure this instance:
     - At the same time as creating the , using the administrative console page for the . For more information about this option, see doc/bpc/t2config_obs_console.dita.
     - At the same time as creating the , using the clientconfig.jacl script.
     - At the same time as creating a Business Process Choreographer configuration, using the bpeconfig.jacl script.

     Note:
   - The schema name for the reporting database.
   - The JNDI name for the data source that is used by the 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 , 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 and event collector.

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.

Choosing a database

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

Before you begin

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

About this task

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

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

On i5/OS® platforms, instead of separate databases as shown here for other distributed platforms, the same component tables reside in uniquely named database collections.

Procedure

Select the components your installation will use to determine which database tables you need. Table 1 lists the components and their corresponding required database tables.
For the latest information on platform-specific disk space requirements, supported operating systems, 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.

Table 1. Databases required by individual components

Component	**Database (name specified is the default and can be changed for your use)**
AppScheduler	Common database (WPRCSDB)
Business Process Choreographer	Business Process Choreographer database (BPCDB)
Business Process Choreographer Explorer reporting	Business Process Explorer reporting database (OBSRVDB) **Important:** To prevent performance degradation, make sure that the reporting database has its own database and not just tables within another database.
Business Space	WPRCSDB (the Common database)
Common Event Infrastructure (CEI)	CEI database (EVENT) **Important:** To prevent performance degradation, make sure that CEI has its own database and not just tables within another database.
Enterprise Service Bus	WPRCSDB (the Common database)
EventSequencing (LockManager)	WPRCSDB (the Common database)
Mediation	WPRCSDB (the Common database)
Recovery	WPRCSDB (the Common database)
Relationships	WPRCSDB (the Common database)
Selectors/Business Rules	WPRCSDB (the Common database)
Service Integration Bus	SIBDB (created during messaging engine configuration)

Related concepts

Database configurations
WebSphere Process Server uses a number of database tables to hold, store and track information. Creating these database tables is part of the process of configuring the WebSphere Process Server. You can create these database tables during profile creation or you can choose to create them separately using scripts.

Common database configurations
The Common database configurations contain information about supported database types; scripts and their locations; profile creation configuration actions; installation parameters; types of created tables and user ID privileges.

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

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

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

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

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

JDBC providers
JDBC providers allow applications to interact with relational databases.

Data sources
Data sources provide the link between applications and relational databases.

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

Related tasks

Creating profiles
Learn how to create new WebSphere Enterprise Service Bus or WebSphere Process Server profiles. You can create profiles from a command line by using the manageprofiles command, or interactively by using the Profile Management Tool graphical user interface (GUI).

Augmenting profiles
You can augment existing WebSphere Application Server, WebSphere Application Server Network Deployment, or WebSphere Application Server Network Deployment with Web Services Feature Pack profiles into WebSphere Enterprise Service Bus or WebSphere Process Server profiles, or WebSphere Enterprise Service Bus profiles into WebSphere Process Server profiles. Use the instructions in this topic to augment profiles from a command line by using the manageprofiles command, or interactively by using the Profile Management Tool graphical user interface (GUI).

Identifying necessary security authorizations

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

Before you begin

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

About this task

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

- Acquire, or supply to your security administrator, user IDs and passwords that have enough authority to install software on the systems.

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

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

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

Results

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

Related information

Planning security, user IDs, and authorizations

Considerations for Service Component Architecture support in servers and clusters

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

SCA applications (also called service applications) require the use of one or more of the automatically created service integration buses. Each application uses a set of messaging resources, which are called *destinations*. These destinations require configured messaging engines, and they can be hosted on the same server or cluster as the application or on a remote server or cluster. Messaging engines typically use database data sources; note that a file store can be used in place of a database data source in a stand-alone server profile if that option was selected during profile creation.

By default, new servers and clusters in a network deployment or managed node environment 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

Chapter 3. Using multiple platforms within a cell

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

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

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

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

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

Related concepts

Deployment 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.

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

Related information

➡ Heterogeneous Cells - cells with nodes on mixed operating system platforms

Chapter 4. Planning your deployment environment

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

Before you begin

Make sure you completed these tasks.
- Identified available resources
- Chose a database type
- Identified necessary authorities

About this task

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

Procedure

1. Determine the purpose for the deployment environment.
2. Identify the functional requirements of the deployment environment
   a. Identify the component types you will deploy.

      Consider the component types and the interactions between components as part of the requirements.

   b. Identify the import and export implementation types and transports.

      Consider the resources needed for the databases or Java Message Service (JMS) resources and the need for business events and their transmission mechanism.

   c. Identify any functional requirements not related to applications.

      Consider security servers, routers and any other hardware or software requirements to handle business events.
3. Identify the capacity and performance requirements for your environment.
4. Decide on the number of physical servers you need for each function.
5. Identify the redundancy requirements for your environment.
   a. Identify the number of servers you need for failover.

   b. Identify the number of routers you need.

      Your choice of router will be influenced by exports of deployed modules, the types of queues you define on the service integration bus, Service Component Architecture (SCA) exports, and the type of load balancing you want among your clusters. IBM provides an embedded router used for Web Services exports with Service Object Access Protocol (SOAP)/JMS transports or JMS exports. However, if you choose not to use this embedded router provided by IBM, you will need to determine how to balance the load among your clusters based on the technology you are using.
6. Design your deployment environment.

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

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

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

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

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

What to do next

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

Related concepts

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

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

Clusters in deployment environments
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 to install WebSphere Process Server while installing WebSphere Integration Developer

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

Before you begin

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

About this task

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

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

Procedure

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

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

What to do next

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

Related concepts

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

Related tasks

Configuring profiles with default values
Learn how to create or augment profiles using the Profile Management Tool with default configuration settings.

Augmenting profiles
You can augment existing WebSphere Application Server, WebSphere Application Server Network Deployment, or WebSphere Application Server Network Deployment with Web Services Feature Pack profiles into WebSphere Enterprise Service Bus or WebSphere Process Server profiles, or WebSphere Enterprise Service Bus profiles into WebSphere Process Server profiles. Use the instructions in this topic to augment profiles from a command line by using the manageprofiles command, or interactively by using the Profile Management Tool graphical user interface (GUI).

Related reference

Users and schemas for databases
During the installation of WebSphere Process Server, you have the option of using the default schema name and user ID privileges when installing your databases. However, your database design may require separate user ID and schema name privileges. You can review the three provided scenarios to determine when and how to configure different schema name and user ID privileges when installing WebSphere Process Server.

Related information

Installing the software
You can obtain WebSphere Process Server product files in two ways, from the disks in the product package or by downloading installation images from the Passport Advantage® site, if you are licensed to do so. You install the software using the installation wizard in graphical interface mode or in silent mode. In silent mode, the installation wizard does not display a graphical interface, but reads your responses from a response file.

Configuring Business Process Choreographer

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

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

Before you begin

Review instructions for installing WebSphere Process Server in this information center. Also, since this scenario requires you to install WebSphere Integration Developer on one server, review the installation instructions for this product at Installing WebSphere Integration Developer.

About this task

Use this procedure when you want to install WebSphere Process Server for use as the test environment server for WebSphere Integration Developer.

Examples of when you would use this scenario include:

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

Procedure

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

Location	Considerations
WebSphere Process Server (for testing) and WebSphere Integration Developer (for development) will be installed on the same physical server	• Make sure that the server has the capacity to handle both workloads. • Make sure all developers can access the server. • Consider installing WebSphere Process Server at the same time you install WebSphere Integration Developer.
WebSphere Process Server (for testing) and WebSphere Integration Developer (for development) will be installed on different physical servers	• Make sure that both servers can communicate. • Make sure all developers can access the server.

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

   Important: To use this remote installation of WebSphere Process Server as a test environment, you must obtain and apply to it a specific integration test client update. You install this update using the WebSphere Update Installer.

To obtain and install the update, refer to the instructions in the following technote: Update required for the integration test client V6.2. This technote provides both the download package and a link to instructions for installing it.

8. Install WebSphere Integration Developer on the development server you selected.

What to do next

Configure WebSphere Integration Developer to use the server you isolated.

Related concepts

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

Related tasks

Verifying the product installation
Use the installation verification tools to verify that the installation of WebSphere Process Server and the creation of the stand-alone server or deployment manager profiles were successful. A *profile* consists of files that define the runtime environment for a deployment manager or a server. Verify the core product files with the installver_wbi checksum tool. Verify each profile with the installation verification test (IVT) tool.

Configuring profiles with default values
Learn how to create or augment profiles using the Profile Management Tool with default configuration settings.

Augmenting profiles
You can augment existing WebSphere Application Server, WebSphere Application Server Network Deployment, or WebSphere Application Server Network Deployment with Web Services Feature Pack profiles into WebSphere Enterprise Service Bus or WebSphere Process Server profiles, or WebSphere Enterprise Service Bus profiles into WebSphere Process Server profiles. Use the instructions in this topic to augment profiles from a command line by using the manageprofiles command, or interactively by using the Profile Management Tool graphical user interface (GUI).

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

Related reference

Users and schemas for databases
During the installation of WebSphere Process Server, you have the option of using the default schema name and user ID privileges when installing your databases. However, your database design may require separate user ID and schema name privileges. You can review the three provided scenarios to determine when and how to configure different schema name and user ID privileges when installing WebSphere Process Server.

Related information

Installing the software
You can obtain WebSphere Process Server product files in two ways, from the disks in the product package or by downloading installation images from the Passport Advantage site, if you are licensed to do so. You install the software using the installation wizard in graphical interface mode or in silent mode. In silent mode,

the installation wizard does not display a graphical interface, but reads your responses from a response file.

Configuring Business Process Choreographer

Planning for a default stand-alone environment

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

Before you begin

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

About this task

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

Procedure

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

   Important: If your future plans include federating this environment into a deployment manager cell, make sure that you use a database and database drivers that support remote access. Examples of these types of products are Derby Network and Java toolbox JDBC.
4. Schedule and coordinate the installation of WebSphere Integration Developer and WebSphere Process Server to minimize impacts to the development community.

What to do next

Install the software.

Related concepts

Stand-alone server
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
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.

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

Related tasks

Verifying the product installation
Use the installation verification tools to verify that the installation of WebSphere Process Server and the creation of the stand-alone server or deployment manager profiles were successful. A *profile* consists of files that define the runtime environment for a deployment manager or a server. Verify the core product files with the installver_wbi checksum tool. Verify each profile with the installation verification test (IVT) tool.

Augmenting profiles
You can augment existing WebSphere Application Server, WebSphere Application Server Network Deployment, or WebSphere Application Server Network Deployment with Web Services Feature Pack profiles into WebSphere Enterprise Service Bus or WebSphere Process Server profiles, or WebSphere Enterprise Service Bus profiles into WebSphere Process Server profiles. Use the instructions in this topic to augment profiles from a command line by using the manageprofiles command, or interactively by using the Profile Management Tool graphical user interface (GUI).

Related reference

Users and schemas for databases
During the installation of WebSphere Process Server, you have the option of using the default schema name and user ID privileges when installing your databases. However, your database design may require separate user ID and schema name privileges. You can review the three provided scenarios to determine when and how to configure different schema name and user ID privileges when installing WebSphere Process Server.

Related information

Configuring Business Process Choreographer

Installing the software
You can obtain WebSphere Process Server product files in two ways, from the disks in the product package or by downloading installation images from the Passport Advantage site, if you are licensed to do so. You install the software using the installation wizard in graphical interface mode or in silent mode. In silent mode, the installation wizard does not display a graphical interface, but reads your responses from a response file.

Planning for a custom stand-alone environment

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

Before you begin

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

About this task

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

Procedure

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

   Some systems, such as z/OS and i5/OS do not have automated methods to create databases and tables for messaging engines and Common Event Infrastructure (CEI). When creating databases for these systems, make sure you have sufficient authorization to 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 impact what other software and resources you must install.
4. Decide how the applications access any resources they require.

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

   You can create and configure the server while installing the software or you can create and configure the server using the Profile Management Tool. You can also use the administrative console to create and configure the server. Experienced installers can also use scripts to handle these tasks. Understand the benefits and drawbacks of all methods before making a choice.
6. Identify or create any user IDs with the authorization you need to complete installation.

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

   Important: If your future plans include federating this environment into a deployment manager cell, make sure that you use a database and database drivers that support remote access. Examples of these types of products are Derby Network and Java toolbox JDBC.
8. Schedule and coordinate the installation of WebSphere Integration Developer and WebSphere Process Server to minimize impacts to the development community.

What to do next

Install the software.

Related concepts

Stand-alone server
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
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.

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

Related tasks

Verifying the product installation
Use the installation verification tools to verify that the installation of WebSphere Process Server and the creation of the stand-alone server or deployment manager profiles were successful. A *profile* consists of files that define the runtime environment for a deployment manager or a server. Verify the core product files with the installver_wbi checksum tool. Verify each profile with the installation verification test (IVT) tool.

Augmenting profiles
You can augment existing WebSphere Application Server, WebSphere Application Server Network Deployment, or WebSphere Application Server Network Deployment with Web Services Feature Pack profiles into WebSphere Enterprise Service Bus or WebSphere Process Server profiles, or WebSphere Enterprise Service Bus profiles into WebSphere Process Server profiles. Use the instructions in this topic to augment profiles from a command line by using the manageprofiles command, or interactively by using the Profile Management Tool graphical user interface (GUI).

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

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 systems involved in the environment. Not every server in a deployment environment requires a WebSphere Process Server.

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

Related reference

Users and schemas for databases
During the installation of WebSphere Process Server, you have the option of using the default schema name and user ID privileges when installing your databases. However, your database design may require separate user ID and schema name

privileges. You can review the three provided scenarios to determine when and how to configure different schema name and user ID privileges when installing WebSphere Process Server.

Related information

Configuring Business Process Choreographer

Installing the software

You can obtain WebSphere Process Server product files in two ways, from the disks in the product package or by downloading installation images from the Passport Advantage site, if you are licensed to do so. You install the software using the installation wizard in graphical interface mode or in silent mode. In silent mode, the installation wizard does not display a graphical interface, but reads your responses from a response file.

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

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

Before you begin

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

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

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

About this task

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

Procedure

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

   Some systems, such as z/OS and i5/OS do not have automated methods to create databases and tables for messaging engines and Common Event Infrastructure (CEI). When creating databases for these systems, make sure you have sufficient authorization to 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 impact what other software and resources you must install.
7. Decide how the applications access any resources they require.

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

   You can create and configure servers while installing the software or you can create and configure servers using the Profile Management Tool. You can also use the administrative console or scripts to create and configure servers. Understand the benefits and drawbacks of all methods before making a choice.
9. Decide how any servers created on the same hardware will share the resources on that system.

   You can either install the software into separate locations, use different profiles, or, on i5/OS use different logical partitions to accomplish the sharing.
10. Identify or create any user IDs with the authorization you need to complete installation.

What to do next

Install your deployment environment.

Related concepts

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

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

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

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

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.

Related tasks

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

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

Identifying available resources
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
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 systems involved in the environment. Not every server in a deployment environment requires a WebSphere Process Server.

Related reference

Users and schemas for databases
During the installation of WebSphere Process Server, you have the option of using the default schema name and user ID privileges when installing your databases. However, your database design may require separate user ID and schema name privileges. You can review the three provided scenarios to determine when and how to configure different schema name and user ID privileges when installing WebSphere Process Server.

Related information

Planning to install Network Deployment

Introduction: Clusters

Configuring Business Process Choreographer

Planning for a custom deployment environment

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

Before you begin

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

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

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

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

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

About this task

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

Procedure

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

   Some systems, such as z/OS and i5/OS do not have automated methods to create databases and tables for messaging engines and Common Event Infrastructure (CEI). When creating databases for these systems, make sure you have sufficient authorization to 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 impact what other software and resources you must install.
7. Decide how the applications access any resources they require.

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

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

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

    Important: If your future plans include federating this environment into a deployment manager cell, make sure that you use a database and database drivers that support remote access. Examples of these types of products are Derby Network and Java toolbox JDBC.
11. Schedule and coordinate the installation of WebSphere Integration Developer and WebSphere Process Server to minimize impacts to the development community.

What to do next

Install your deployment environment.

Related concepts

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

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

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

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

Identifying available resources
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
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 systems involved in the environment. Not every server in a deployment environment requires a WebSphere Process Server.

Related reference

Users and schemas for databases
During the installation of WebSphere Process Server, you have the option of using the default schema name and user ID privileges when installing your databases. However, your database design may require separate user ID and schema name privileges. You can review the three provided scenarios to determine when and how to configure different schema name and user ID privileges when installing WebSphere Process Server.

Related information

Planning to install Network Deployment

Introduction: Clusters

Configuring Business Process Choreographer

Profiles

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

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

The first profile can be automatically created when you install WebSphere Process Server. You can later use the Profile Management Tool or the manageprofiles command to create further profiles on the same system, without installing a second copy of the binary files.

Note: On distributed platforms, each profile has a unique name. On z/OS all the profiles are named “default”.

The profile directory

Every profile in the system has its own directory containing all its files. You specify the location of the profile directory when you create the profile: by default it is in the profiles directory in the directory where WebSphere Process Server was installed, for example, the Dmgr01 profile is in C:\Program Files\IBM\WebSphere\ProcServer\profiles\Dmgr01.

The First steps console

Linux UNIX Windows i5/OS Every profile in the system has a First steps console, which is a user interface for familiarizing yourself with the stand-alone server, deployment manager, or managed node.

The default profile

The first profile that you create within one installation of WebSphere Process Server is the *default profile*. The default profile is the default target for commands issued from the \bin directory in the directory where WebSphere Process Server was installed. If only one profile exists on a system, every command operates on that profile. If you create another profile, you can make it the default.

Note: The default profile is not necessarily a profile whose name is "default".

Augmenting profiles

If you already have a deployment manager, a custom profile, or a stand-alone server created for WebSphere Application Server Network Deployment or WebSphere ESB, you can *augment* its profile to support WebSphere Process Server in addition to existing function. To augment a profile, first install WebSphere Process Server. Then use the Profile Management Tool or the manageprofiles command.

Restriction: You cannot augment a profile if it defines a managed node that is already federated to a deployment manager.

Related concepts

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.

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.

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

Profile commands in a multiprofile environment
When two or more profiles exist on a server, certain commands require that you specify the profile to which the command applies. These commands use the `-profileName` attribute to identify which profile to address. To overcome having to specify the `-profileName` attribute for each command, use the versions of the commands that exist in the `bin` directory of each profile.

Related tasks

Creating profiles using the Profile Management Tool
Use the Profile Management Tool graphical user interface (GUI) to create a stand-alone server profile, a deployment manager profile, or a custom profile.

Creating profiles using the manageprofiles command
Learn about creating a profile from the command line using the manageprofiles command and a property file.

Related information

Starting the First steps console
After installing WebSphere Process Server, use the First steps console to start product tooling, access product documentation, or direct elements such as servers and administrative consoles related to individual profiles. A generic version of the console, plus a version for each profile in your installation are available.

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.

A stand-alone server is easy to set up, and has a First steps console from which you can start and stop the server and open the samples gallery and the administrative console. If you install the WebSphere Process Server samples, and then open the samples gallery, a sample solution is deployed to the stand-alone server. You can explore the resources used for this sample in the administrative console.

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

It is possible to start off with a stand-alone server and later include it in a network deployment environment, by federating it to a deployment manager cell, *provided that no other nodes have been federated to that cell*. It is not possible to federate multiple stand-alone servers into one cell. To federate the stand-alone server, use the administrative console of the deployment manager or the addNode command. The stand-alone server must not be running when you federate it using the addNode command.

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

Related concepts

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

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

Data sources
Data sources provide the link between applications and relational databases.

Service integration buses 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.

Related tasks

Federating stand-alone server profiles to a deployment manager
Learn how to use the **addNode** command to federate a stand-alone server profile into a deployment manager cell. After federation, a node agent process is created. Both this node agent and the server process are managed by the deployment manager. If you federate a stand-alone server profile and include all of its applications, the act of federation installs the applications on the deployment manager. A stand-alone server profile can be federated only if there are no other federated profiles.

Network deployment

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

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

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

Upgrading WebSphere Application Server Network Deployment

WebSphere Application Server Network Deployment, as its name implies, supports network deployment of applications. If you already have a WebSphere Application Server Network Deployment installation, which you are upgrading with WebSphere Process Server, you are familiar with the concept of network deployment. You probably have one or more network deployment cells each with its deployment manager and managed nodes. You can *augment* their profiles to support WebSphere Process Server, using the WebSphere Process Server Profile Management tool. After augmentation, the servers still continue to function as application servers, but they are also capable of supporting Service Component Architecture (SCA) modules.

Implementing WebSphere Process Server network deployment

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

Related information

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

Deployment 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.

When creating a deployment environment, the deployment manager profile is the first profile that you create. The deployment manager has a First steps console, from which you can start and stop the deployment manager and start its administrative console. 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
With careful planning, you can create a deployment manager cell that includes nodes on both distributed, i5/OS and z/OS operating system platforms.

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

Overview of managed nodes

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

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

- Start and stop server processes.

- Synchronize configuration data on the deployment manager with the copy on the node.

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

A managed node can contain one or more servers, which are managed by a deployment manager. You can deploy solutions to the servers in a managed node, but the managed node does not contain a sample applications gallery. The managed node is defined by a custom profile and has a First steps console.

Related concepts

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

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

Deployment environments

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

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

Stand-alone environment

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

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

Interconnected servers

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

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

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

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

In addition to the performance, availability, scalability, isolation, security, and stability characteristics that cannot be provided by a stand-alone server, a deployment environment of interconnected servers or clusters has the additional advantage that you can manage all the servers or clusters from a centralized *deployment manager*.

Deployment environment patterns

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

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

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

Deciding when to create the deployment environment

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

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

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

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

Managed servers

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

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

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

Clusters 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.

Deployment environment patterns

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

A guided installation wizard exists to implement the patterns to simplify the installation process.

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

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

Single cluster pattern

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

All components are run on a single cluster:

- Service Component Architecture (SCA) application bus members
- SCA system bus members
- Business Process Choreographer bus members
- Business Process Choreographer components such as the explorer
- Business Process Choreographer container
- Common Event Interface (CEI) bus members
- CEI server
- Business Rules manager
- Application deployment target

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

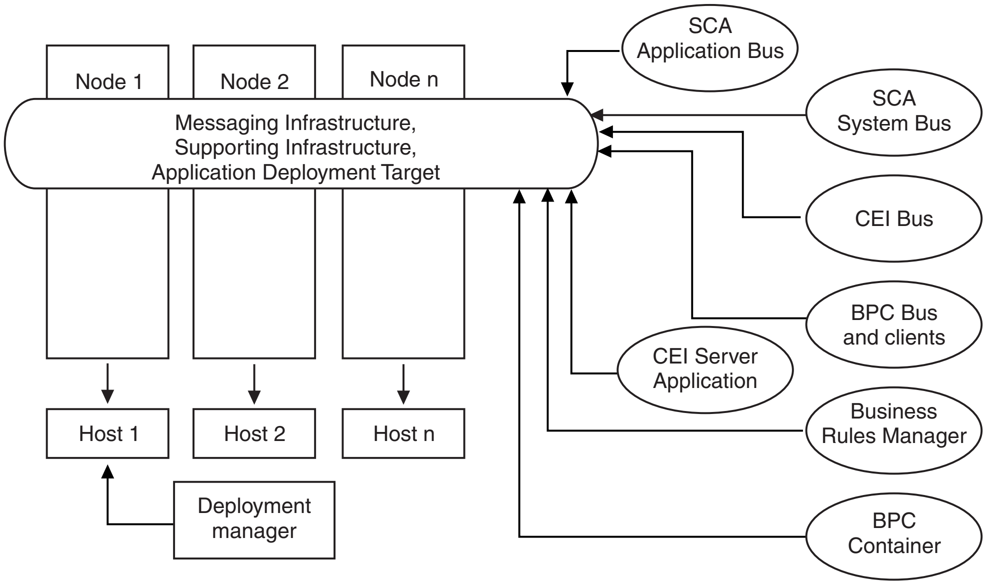


Figure 1. Single cluster pattern

Remote messaging pattern

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

Remote messaging cluster:

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

Support infrastructure and application deployment target cluster:

- CEI server application
- Business Rules manager

- Business Process Choreographer components such as the explorer
- Application deployment target

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

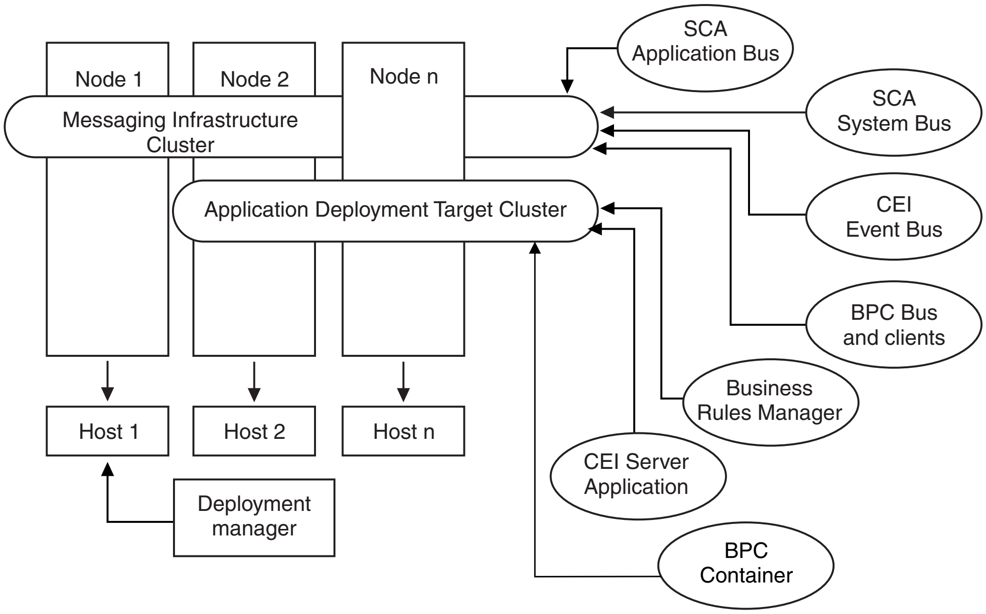


Figure 2. Remote messaging pattern

Remote messaging and remote support pattern

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

Remote messaging infrastructure cluster:
- Service Component Architecture (SCA) application bus members
- SCA system bus members
- Business Process Choreographer (BPC) bus members
- Common Event Interface (CEI) bus members

Remote support infrastructure cluster:
- CEI server application
- Business Rules manager
- Business Process Choreographer components such as the explorer

Application deployment cluster:
- Application deployment target
- Business Process Choreographer container

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

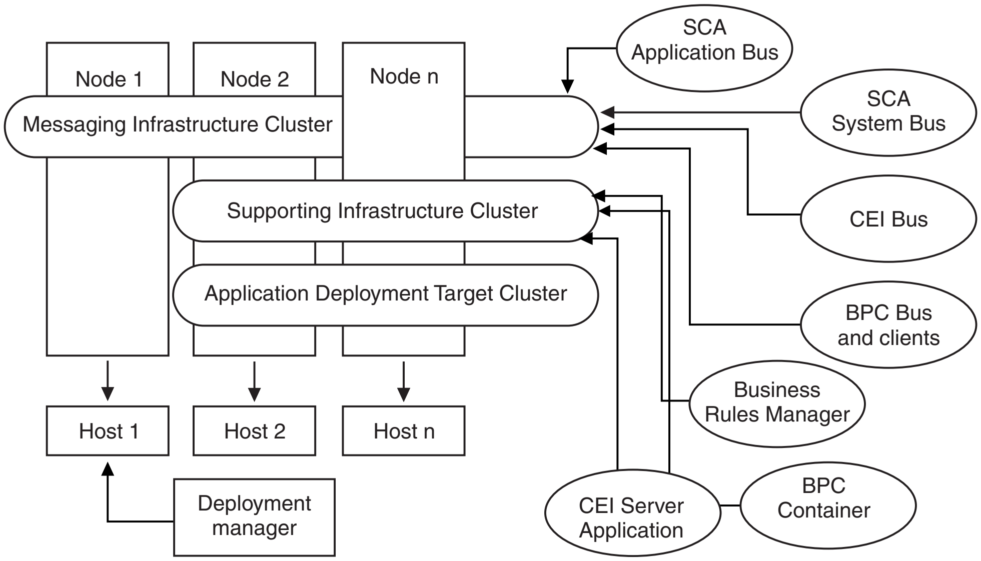


Figure 3. Remote messaging and support pattern

Resource allocation example

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

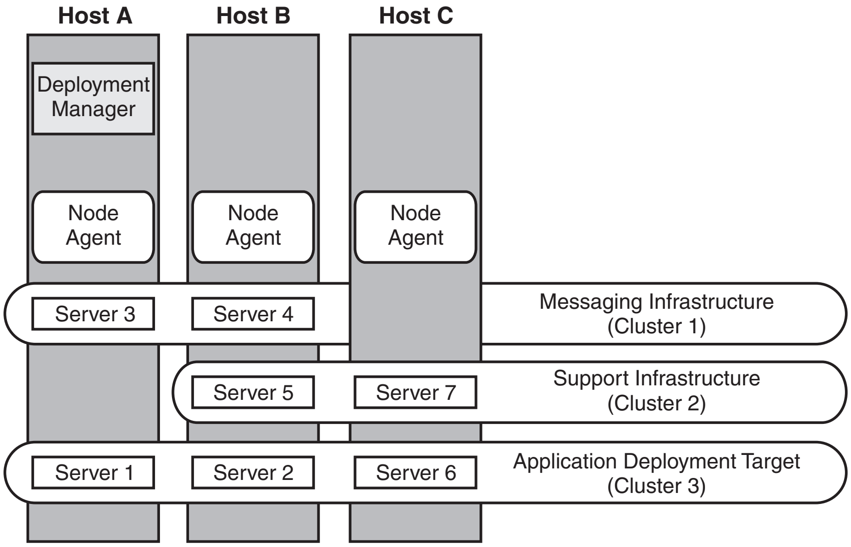


Figure 4. Resource allocation example

Related concepts

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 recover from system failures that can result from both ordinary and extraordinary circumstances.

Deployment environment functions

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

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

IBM-supplied deployment environment patterns

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

Application deployment target

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

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

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

Supporting infrastructure

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

- Business rules
- Selectors
- human tasks
- Business processes

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

Messaging infrastructure

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

Custom deployment environments

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

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

The administrative console groups collaborative units as follows:

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

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

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

Choosing your deployment environment pattern

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

Before you begin

You should familiarize yourself with the information in:

- Assessing your business requirements
- Identifying available resources

About this task

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

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

Procedure

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

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

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

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

What to do next

Install and configure the product.

Related concepts

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

Error handling strategy and solution recovery
WebSphere Process Server has error handling capabilities and tools that you can utilize for recovery purposes.

Chapter 5. Implementing a deployment environment

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

Before you begin

- Plan your topology and record the decisions you make about:
  - The servers and clusters involved.
  - The number of databases required.

    Note: i5/OS If you are installing on an i5/OS system, you can only have one DB2 Universal database in the system. On i5/OS, instead of separate databases, the component tables reside in uniquely named database collections.
  - 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.
- Windows Linux UNIX Install and configure your database product following the product documentation. You must:
  - Configure the product as a server.
  - Define a userid for WebSphere Process Server to use to access the data and tables within the database.
  - Optional: **Optional:** Create the WebSphere Process Server common database, called WPRCSDB by default.

    If you created this database during product installation or through the Profile Management Tool, bypass this step.
  - Create all other databases you need for your configuration. If you do not create a database for a specific function, the system uses the WebSphere Process Server common database.
- Optional: i5/OS **Optional:** Create the WebSphere Process Server common database schema. The schema name must be unique in the system.
- Synchronize the system clocks on all the servers. When adjusted for the same time zone, the clocks must be within five minutes of each other.
- Make sure all servers involved in the topology can be located by both IP address and Domain Name Server (DNS) name.
- Make sure you have a user ID that has the appropriate authority to create directories and files on all systems.
- Make sure you perform any other preparation that might be needed to coexist with other products and provide any needed redundancy.

About this task

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

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

Procedure

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

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

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

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

   If your design implements a custom deployment environment, you must create all the clusters needed to provide the necessary functions. These functions include messaging support for application deployment, application support and Common Event Infrastructure support.
7. Configure the databases or database tables required for your topology, if you chose deferred table creation.

   Configuration consists of running the scripts generated by the deferred option.
   a. Configure the common database tables. This table is in the common database.
   b. Configure the messaging engine database tables. This table is in the common database.
   c. Optional: Configure the Business Process Choreographer database tables.

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

      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. This table is in the common database.

    e. Configure the Common Event Infrastructure database.

8. Install and configure a routing server. This could be an IBM HTTP server or another server of your choosing. This server allows clients to access the applications within this topology.
9. Verify the installation by installing and running test applications.

What to do next

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

Related concepts

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

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

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

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.

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.

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

Related tasks

Verifying the product installation
Use the installation verification tools to verify that the installation of WebSphere Process Server and the creation of the stand-alone server or deployment manager profiles were successful. A *profile* consists of files that define the runtime environment for a deployment manager or a server. Verify the core product files with the installver_wbi checksum tool. Verify each profile with the installation verification test (IVT) tool.

Configuring profiles with default values
Learn how to create or augment profiles using the Profile Management Tool with default configuration settings.

Configuring profiles with customized values
Learn how to create or augment a profile with customized configuration settings using the Profile Management Tool.

Configuring profiles for a deployment environment

Learn how to create or augment a profile with customized configuration settings to be used in a new or existing deployment environment pattern. Use the Profile Management Tool to configure the profile.

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

Federating custom nodes to a deployment manager
You can use the addNode command to federate a custom node into a deployment manager cell. The following instructions guide you through the process of federating and deploying custom nodes.

Creating profiles
Learn how to create new WebSphere Enterprise Service Bus or WebSphere Process Server profiles. You can create profiles from a command line by using the manageprofiles command, or interactively by using the Profile Management Tool graphical user interface (GUI).

Augmenting profiles
You can augment existing WebSphere Application Server, WebSphere Application Server Network Deployment, or WebSphere Application Server Network Deployment with Web Services Feature Pack profiles into WebSphere Enterprise Service Bus or WebSphere Process Server profiles, or WebSphere Enterprise Service Bus profiles into WebSphere Process Server profiles. Use the instructions in this topic to augment profiles from a command line by using the manageprofiles command, or interactively by using the Profile Management Tool graphical user interface (GUI).

Verifying your deployment environment
Before moving your production applications to the new environment, you must test to make sure that all of the components operate correctly.

Related information

Installing the software
You can obtain WebSphere Process Server product files in two ways, from the disks in the product package or by downloading installation images from the Passport Advantage site, if you are licensed to do so. You install the software using the installation wizard in graphical interface mode or in silent mode. In silent mode, the installation wizard does not display a graphical interface, but reads your responses from a response file.

Configuring Business Process Choreographer

Communicating with Web servers

Installing IBM HTTP server

wsadmin tool

Managing node agents

Starting clusters

Stopping clusters

Java Management Extensions connector properties

Chapter 6. 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 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 recommendations contained in the *Planning error prevention and recovery* section assume a remote messaging and remote support pattern, which is comprised of three separate clusters, one for the WebSphere Process Server and one each for the messaging engine and CEI event server.

Related concepts

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

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.

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 utilizing 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 utilizes existing WebSphere Process Server error handling capabilities and tools
- Creating connectivity groups and utilizing 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.

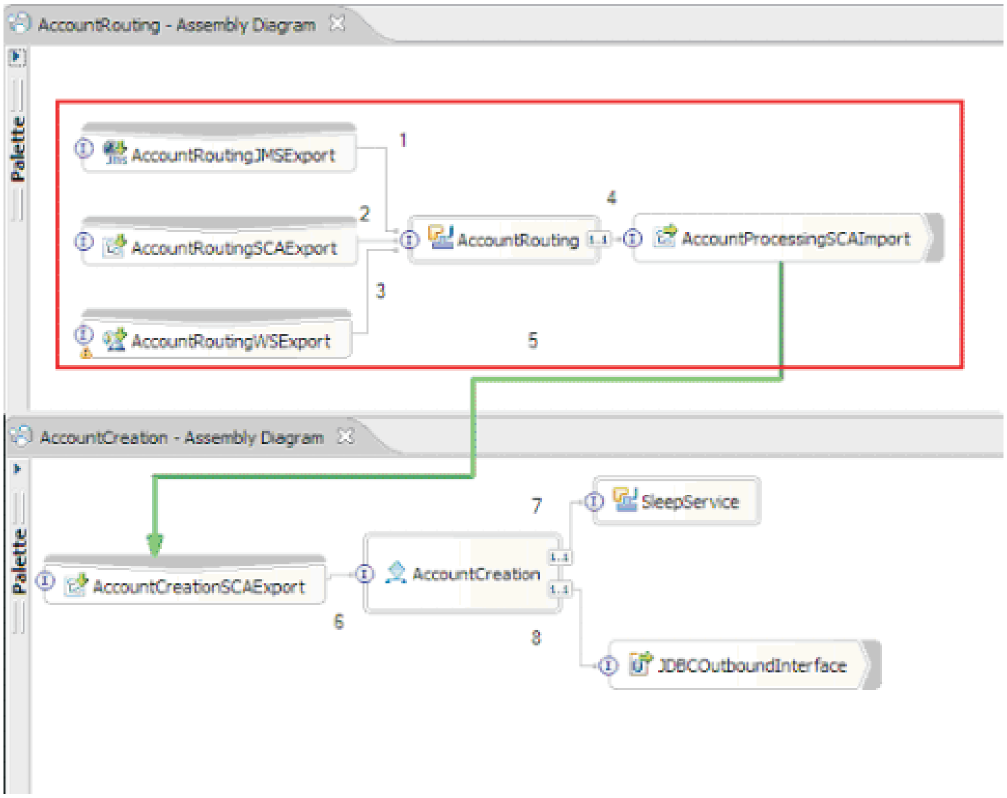
The connectivity group will not contain stateful component types like long-running business processes and Business State Machines. These connectivity groups provide encapsulation and isolation of the specific endpoint's integration requirements. Commonly, WebSphere ESB mediation modules are used for 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
To completely quiesce a system, we must 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. Please 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 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

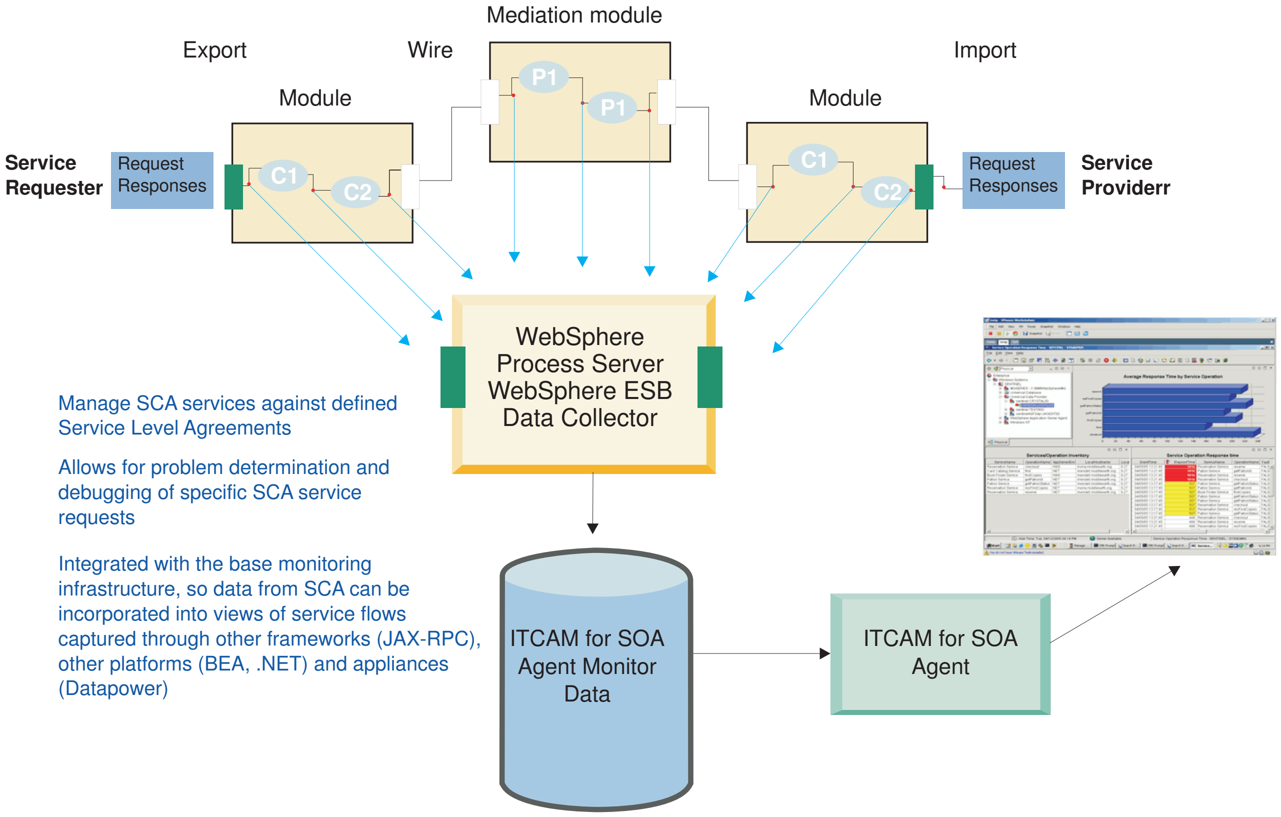
- Visibility into SOA service interactions
- Visibility into message content and transaction flow patterns
- Ability to identify and isolate performance bottlenecks across technology and platform boundaries
- Lightweight, industry standard ARM-based Performance instrumentation
- High performing and flexible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

Monitor business processes

- Manage inflight 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.

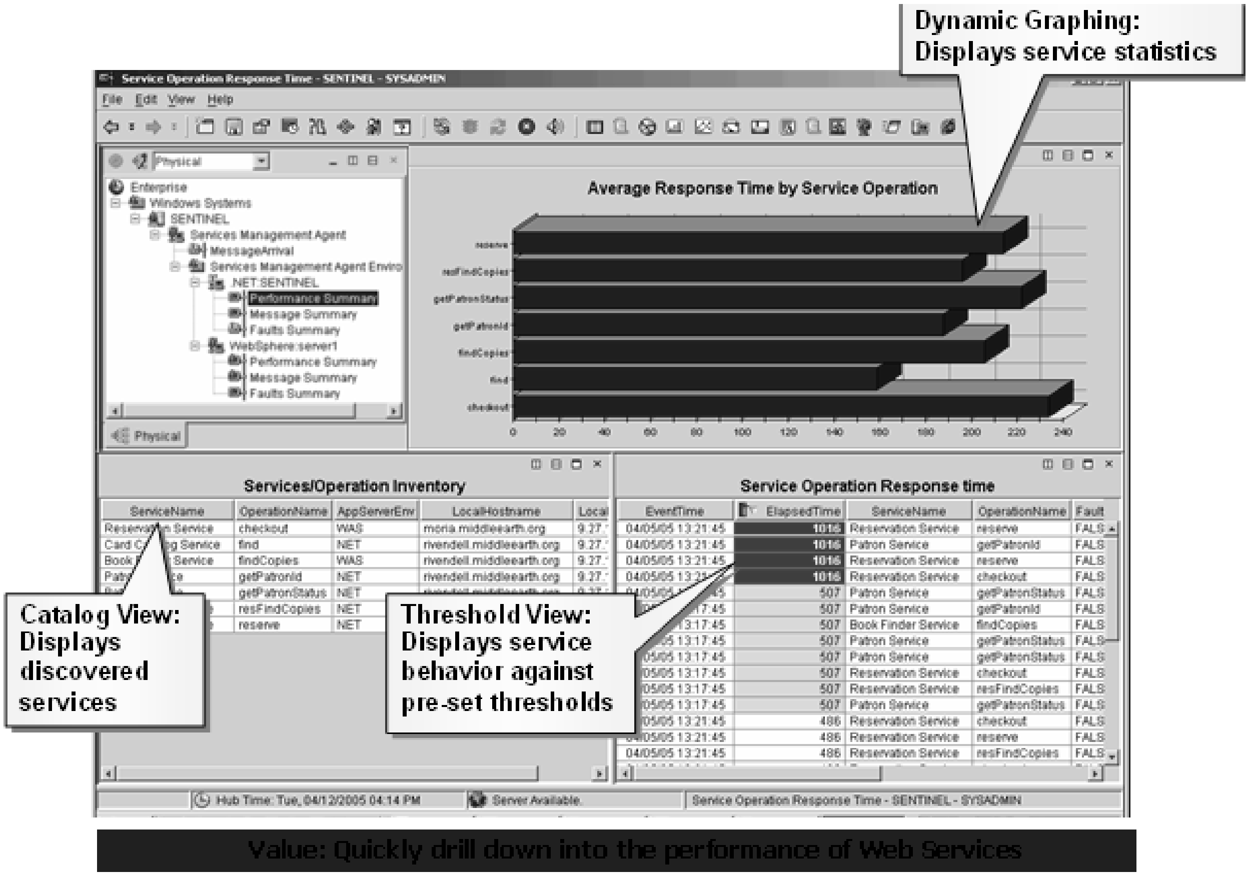


Manage SCA services against defined Service Level Agreements

Allows for problem determination and debugging of specific SCA service requests

Integrated with the base monitoring infrastructure, so data from SCA can be incorporated into views of service flows captured through other frameworks (JAX-RPC), other platforms (BEA, .NET) and appliances (Datapower)

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.



Dynamic Graphing: Displays service statistics
Service Operation Response Time - SENTINEL - SYSADMIN
File
Edit
View
Help
Physical
Enterprise
Windows Systems
SENTINEL
Services Management Agent
MessageArrival
Services Management Agent Enviro
.NET:SENTINEL
Performance Summary
Message Summary
Faults Summary
WebSphere:server1
Performance Summary
Message Summary
Faults Summary
Physical
Average Response Time by Service Operation
reserve
resFindCopies
getPatronStatus
getPatronId
findCopies
find
checkout
0
20
40
60
80
100
120
140
160
180
200
220
240
Catalog View: Displays discovered services
Threshold View: Displays service behavior against pre-set thresholds
Hub Time: Tue, 04/12/2005 04:14 PM
Server Available.
Service Operation Response Time - SENTINEL - SYSADMIN

Services/Operation Inventory

ServiceName	OperationName	AppServerEnv	LocalHostname	Local
Reserva[illegible]n Service	checkout	WAS	moria.middleearth.org	9.27.'
Card C[illegible]g Service	find	NET	rivendell.middleearth.org	9.27.'
Book [illegible] Service	findCopies	WAS	rivendell.middleearth.org	9.27.'
Pat[illegible]e	getPatronId	NET	rivendell.middleearth.org	9.27.'
[illegible]	getPatronStatus	NET	[illegible]	9.27.'
[illegible]	resFindCopies	NET	[illegible]	[illegible]
[illegible]	reserve	NET	[illegible]	[illegible]

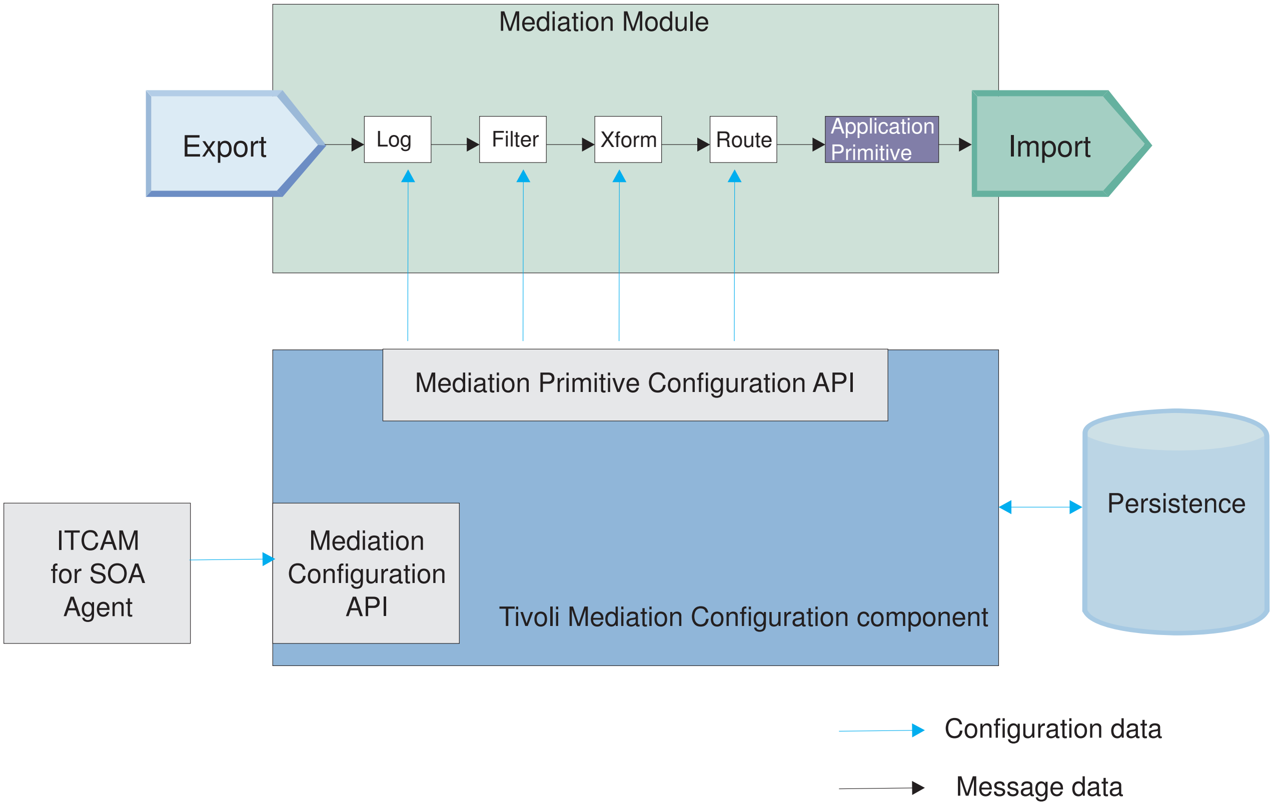
Service Operation Response time

EventTime	ElapsedTime	ServiceName	OperationName	Fault
04/05/05 13:21:45	1016	Reservation Service	reserve	FALS
04/05/05 13:21:45	1016	Patron Service	getPatronId	FALS
04/05/05 13:21:45	1016	Reservation Service	reserve	FALS
04/05/05 13:2[illegible]	1016	Reservation Service	checkout	FALS
04/05[illegible]45	507	Patron Service	getPatronStatus	FALS
[illegible]17:45	507	Patron Service	getPatronId	FALS
[illegible]3:17:45	507	Book Finder Service	findCopies	FALS
[illegible]05 13:17:45	507	Patron Service	getPatronStatus	FALS
[illegible]05/05 13:17:45	507	Patron Service	getPatronStatus	FALS
[illegible]05/05 13:17:45	507	Reservation Service	checkout	FALS
[illegible]05/05 13:17:45	507	Reservation Service	resFindCopies	FALS
[illegible]05/05 13:17:45	507	Patron Service	getPatronStatus	FALS
[illegible]05/05 13:21:45	486	Reservation Service	checkout	FALS
[illegible]05/05 13:21:45	486	Reservation Service	reserve	FALS
04/05/05 13:21:45	486	Reservation Service	resFindCopies	FALS

Value: Quickly drill down into the performance of Web Services

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 information

Generating an IBM Heap

Product version and history information

Enabling verbose garbage collection (verbosegc) in WebSphere Application Server

IBM Support Assistant

Enabling Cross-Component Trace

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, please 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 utilize for recovery purposes.

The architecture team that is building the solution must understand how to utilize 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 (Failed event manager, etc.) of WebSphere Process Server are leveraged appropriately.

Related tasks

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

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.

Related information

Fault handling and compensation handling in business processes

Dealing with faults in your business process

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 recreate 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 utilizes 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

Deploying applications using scripting

Sample Scripts for WebSphere Application Server

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
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

WebSphere Application Server Network Deployment V6: High Availability Solutions

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

Related information

Selecting your deployment pattern

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
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 impacted 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 prior to 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 impacted, and all of them at various stages of the process flow. We have describe how 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 J2EE 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.

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 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

Administering servers and clusters

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
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
The error prevention and recovery information describes how to avoid problems that might cause system failures, and provides or points to information on how 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
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, we must 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 2. 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 3. 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
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

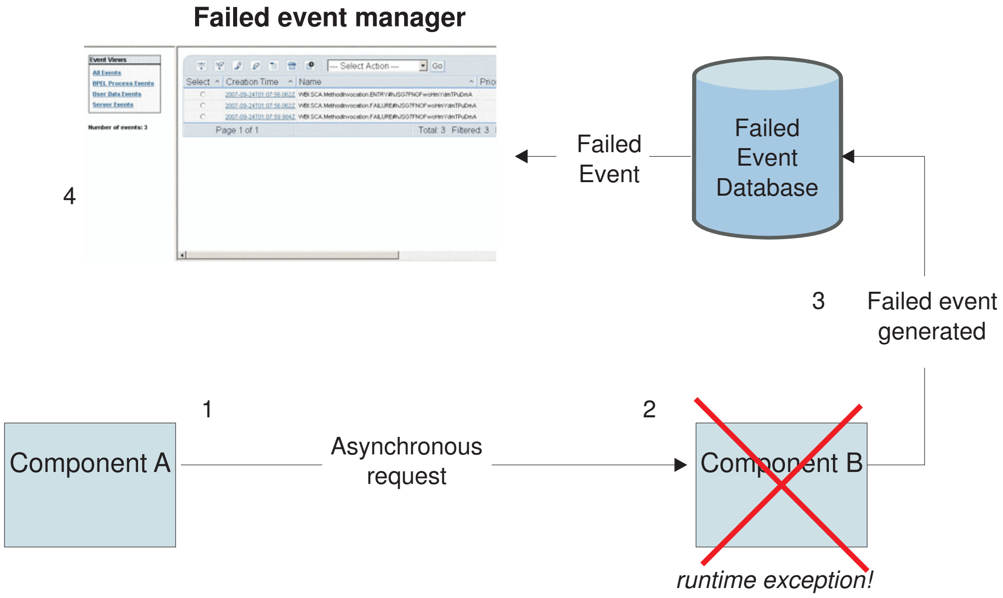
The failed event manager is a web-based client for working with and resubmitting the failed invocations.

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.



Select	Creation Time	Name	Prio
○	2007-09-24T01:07:56.062Z	WBI.SCA.MethodInvocation.ENTRY#hJSG7FNOFwoHmYdmTPuDmA	
○	2007-09-24T01:07:56.062Z	WBI.SCA.MethodInvocation.FAILURE#hJSG7FNOFwoHmYdmTPuDmA	
○	2007-09-24T01:07:59.984Z	WBI.SCA.MethodInvocation.FAILURE#hJSG7FNOFwoHmYdmTPuDmA	

Page 1 of 1 Total: 3 Filtered: 3

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

Notices

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

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

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

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

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

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

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

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

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

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

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

IBM Corporation
1001 Hillsdale Blvd., Suite 400
Foster City, CA 94404
U.S.A.

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows: (c) (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. (c) Copyright IBM Corp. _enter the year or years_. All rights reserved.

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

Programming interface information

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

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

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

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

Trademarks and service marks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (^{R} or ^{TM}), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at ″Copyright and trademark information″ at www.ibm.com/legal/copytrade.shtml.

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

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

Java is a trademark of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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

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

IBM WebSphere Process Server for Multiplatforms, Version 6.2

Printed in USA