WebSphere® Application Server for z/OS V5

Installation and Customization

GA22-7910-00

WebSphere® Application Server for z/OS V5

Installation and Customization

GA22-7910-00

Note
Before using this information and the product it supports, be sure to read the general information under Appendix D, "Notices", on page 233.

First Edition (April 2003)

This edition applies to WebSphere Application Server for z/OS V5 (5655–I35), and to all subsequent releases and modifications until otherwise indicated in new editions.

The most current versions of the WebSphere Application Server for z/OS V5 publications are at this Web site:
`http://www.ibm.com/software/webservers/appserv/zos_os390/`

Contents

Chapter 4. Performing post-installation tasks 123

Chapter 5. Performing advanced tasks 127

Figures

Tables

About this book

WebSphere Application Server for z/OS V5.0: Installation and Customization describes how to

- Plan for, install, and customize the WebSphere for z/OS run-time environment
- Upgrade code levels from one release or service level of the product to another.

  Note: The primary source for migration information for WebSphere for z/OS is the migration information in the z/OS view of the WebSphere Application Server InfoCenter, which you can access via the WebSphere for z/OS library Web site. You should begin your migration planning with that information.
- Set up WebSphere for z/OS in advanced system configurations, such as a cell.

Included are instructions for setting up requisite z/OS functions, such as eNetwork Communication Server (TCP/IP), the Security Server (RACF), and workload management (WLM), for use by WebSphere for z/OS.

Note: The full product name is "WebSphere Application Server for z/OS V5," referred to in this text as "WebSphere for z/OS."

Who should read this book

This book is intended for system programmers, security administrators, network administrators, or database administrators who configure z/OS subsystems and install WebSphere for z/OS.

How this book is organized

Planning for and installing WebSphere for z/OS includes those tasks you must perform prior to installing business applications. It includes such tasks as planning your system configuration and installing the WebSphere for z/OS run-time environment. Chapter 1, "Overview of installation and customization", on page 1 provides a quick introduction to the installation process.

To install the run-time environment, you must perform tasks in two general areas:

- The base z/OS system. You must prepare various z/OS subsystems and your network prior to setting up WebSphere for z/OS. For instance, you must perform such tasks as setting up security controls and defining workload management (WLM) workloads. See Chapter 2, "Preparing the base z/OS environment", on page 11 for details.
- The WebSphere for z/OS run-time environment itself. This includes loading the code, changing PARMLIB members, creating environment files, and running configuration jobs. See Chapter 3, "Installing and customizing your first run-time", on page 45 for details.

Chapter 4, "Performing post-installation tasks", on page 123 covers tasks, such as backing up your system, that you may want to do immediately after installation and customization.

You can get started with WebSphere for z/OS on a monoplex system, then implement advanced security, workload management, database, and sysplex operations later. For these advanced tasks, see Chapter 5, “Performing advanced tasks”, on page 127.

Chapter 6, “Installing new releases and maintenance levels of WebSphere for z/OS”, on page 211 provides general information and procedures for migrating WebSphere for z/OS from one release or service level to another.

Following the last chapter are three appendices of helpful charts, and Appendix D, “Notices”, on page 233, which contains various legal notices.

Note: The appendix on the WebSphere for z/OS WebSphere variables has been removed from this book because all the information is now covered in the Administrative Console and the InfoCenter.

“Glossary” on page 237 tells you where to find information on terms used in this manual.

“Index” on page 239 provides a topic page reference.

Where to find related information, tools, and supplements

Most of the information about WebSphere for z/OS appears in task-oriented articles in the WebSphere Application Server InfoCenter, which you can access through the WebSphere for z/OS library Web site:

`http://www.ibm.com/software/webservers/appserv/zos_os390/library.html`

The WebSphere for z/OS library Web site also includes the following books in PDF format:

- *WebSphere Application Server for z/OS V5.0: License Information*, GA22-7908, which describes the license information for WebSphere for z/OS.
- *WebSphere Application Server for z/OS V5.0: Program Directory*, GI11-2825, which describes the elements of and the installation instructions for WebSphere for z/OS.
- *WebSphere Application Server for z/OS V5.0: Installation and Customization*, GA22-7909, which describes the planning, installation, and customization tasks and guidelines for WebSphere for z/OS.
- *WebSphere Application Server for z/OS V5.0: Operations and Administration*, SA22-7912, which describes z/OS system operations and administration tasks for WebSphere for z/OS and other z/OS subsystems that are configured in the WebSphere for z/OS environment. This book also includes information about improving the performance of WebSphere for z/OS and the applications it hosts.
- *WebSphere Application Server for z/OS V5.0: Messages and Codes*, GA22-7915, which describes messages and codes associated with WebSphere for z/OS.
- *WebSphere Application Server for z/OS V5.0: Diagnosis*, GA22-7915, which provides diagnosis information associated with WebSphere for z/OS.

For additional WebSphere for z/OS tools and supplements, go to the following Web site and select the download link:

`http://www.ibm.com/software/webservers/appserv/zos_os390/`

You also might need to refer to information about other z/OS or OS/390 elements and products. All of this information is available through links at the following Internet locations:

`http://www.ibm.com/servers/eserver/zseries/zos/`
`http://www.ibm.com/servers/s390/os390/`

How to send your comments

Your feedback is important in helping to provide the most accurate and highest quality information.

- To send comments on articles in the WebSphere Application Server InfoCenter:
  1. Display the article in your Web browser and scroll to the end of the article.
  2. Click on the `Feedback` label at the bottom of the article, and a separate window containing an e-mail form appears.
  3. Fill out the e-mail form as instructed, and click on `Submit feedback`.
- To send comments on PDF books, you can e-mail your comments to:

  `wasdoc@us.ibm.com`

  or fax them to 919-254-0206.

  Be sure to include the document name and number, the WebSphere Application Server version you are using, and, if applicable, the specific page, table, or figure number on which you are commenting.

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.

Chapter 1. Overview of installation and customization

WebSphere Application Server for z/OS V5, hereafter referred to as WebSphere for z/OS, builds upon the function of WebSphere Application Server V4.0.1 for z/OS and OS/390 and WebSphere Application Server Advanced Edition.

This manual covers planning, installing, and customizing tasks for WebSphere for z/OS.

Planning for, installing, and customizing WebSphere for z/OS includes those tasks you must perform prior to installing business applications, such as planning your system configuration and installing the WebSphere for z/OS run-time environment. This chapter:

- Gives a general overview of the tasks you must do to initially install and customize WebSphere for z/OS.
- Provides pictures and descriptions of your run-time environment after the initial installation and customization of the base and network deployment systems. The initial base installation and customization is performed on a monoplex or a single system in a sysplex.
- Provides a checklist of items you should consider for your initial installation of WebSphere for z/OS, your application development and client systems, and advanced system configurations, such as WebSphere for z/OS in a cell.

To install the run-time environment initially, you must perform tasks in two general areas:

1. The base z/OS system. You must prepare various z/OS elements, products, and your network prior to setting up WebSphere for z/OS. For instance, you must perform such tasks as updating your TCP/IP network, setting up security controls, and defining workload management (WLM) workloads. See Chapter 2, "Preparing the base z/OS environment", on page 11 for details.
2. The WebSphere for z/OS run-time environment itself. This includes loading the code, changing parmlib members, creating environment files, and running configuration jobs. See Chapter 3, "Installing and customizing your first run-time", on page 45 for details.

If you already have a prior version of WebSphere for z/OS installed and customized, you can configure WebSphere Application Server for z/OS V5 to coexist with it. For more information, see the migration information in the z/OS view of the WebSphere Application Server InfoCenter, which you can access via the WebSphere for z/OS library Web site.

After installation and customization, you can install application development environments for your application developers and client environments for your business applications. For more information about this, see the assembling applications information in the WebSphere Application Server InfoCenter, which you can access via the WebSphere Application Server for distributed platforms library Web page.

When you have stabilized WebSphere for z/OS on the first system, you can enable WebSphere for z/OS in a sysplex. You may also implement other advanced system

configurations, such as organizing cells or connecting your business applications to an IMS or CICS database. These and other topics are in Chapter 5, “Performing advanced tasks”, on page 127.

Diagram of a WebSphere for z/OS run-time configuration

WebSphere Application Server for z/OS V5 is optimized to run in a network deployment configuration. The first step to getting there is to customize your base environment. Figure 1 depicts the typical WebSphere for z/OS base run-time configuration, and Figure 2 on page 3 depicts the typical WebSphere for z/OS run-time configuration after you fully configure the product for network deployment on a single system. The network deployment configuration is an extension of the base system, and introduces clusters, nodes and cells, arranged in any variety of configurations. See “WebSphere for z/OS terminology and system configuration breakdown” on page 4 for further description of each.

WebSphere for z/OS base system

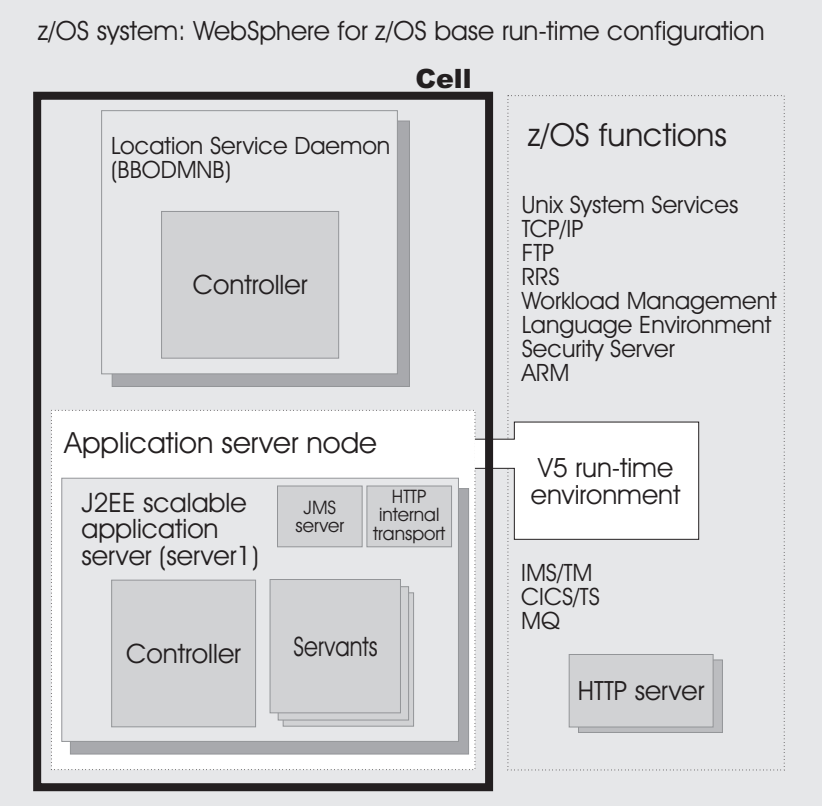


Figure 1. WebSphere for z/OS base system run-time configuration

As Figure 1 shows, a typical WebSphere for z/OS base run-time includes a location service daemon (BBODMNB) and one node which includes an Application Server (server1) with a controller and any number of servants.

The run-time servers use other z/OS functions, as indicated in Figure 1, such as z/OS UNIX and TCP/IP. Part of installing WebSphere for z/OS includes configuring these functions for use by the run-time (more about that in Chapter 2, “Preparing the base z/OS environment”, on page 11).

Java servers contain at least one Web container and one EJB container. The Web container manages Web applications (servlets and JavaServer Pages), while the EJB container manages enterprise beans.

The HTTP internal transport, which is part of the Java Application Server, is a functional component that acts as an HTTP protocol catcher for Web applications. The HTTP internal transport is depicted in the server1 Java Application Server in Figure 1 on page 2. For more information, see "The WebSphere for z/OS HTTP internal transport" on page 6.

The JMS server, which is also part of the Java Application Server, hosts the WebSphere for z/OS JMS function. The JMS server is depicted in the server1 Java Application Server in Figure 1 on page 2. For more information, see the the WebSphere Application Server InfoCenter, which you can access via the WebSphere Application Server for distributed platforms library Web page.

WebSphere for z/OS network deployment system

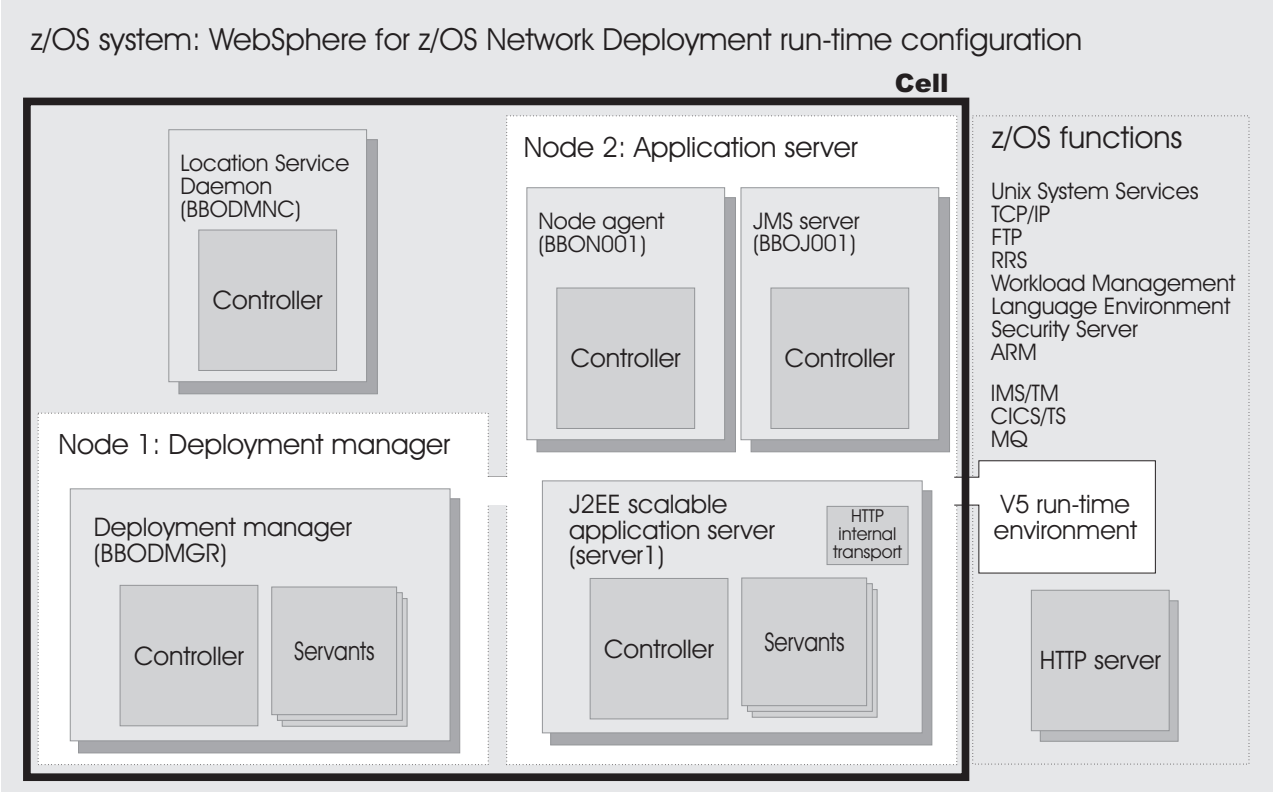


Figure 2. WebSphere for z/OS Network Deployment run-time configuration

As Figure 2 shows, a typical WebSphere for z/OS network depolyment run-time includes a location service daemon (BBODMNC and two nodes (both housed in a cell, which can span systems)—one for the Deployment Manager and one for the Application Server. The Deployment Manager node includes a Deployment Manager (BBODMGR) with a controller and any number of servants. The Application Server node includes a node agent (BBON001), a JMS server (BBOJ001), and an Application Server (server1) with a controller and any number of servants.

WebSphere for z/OS terminology and system configuration breakdown

In WebSphere for z/OS, the functional component on which applications run is called a *server*. Servers comprise address spaces that actually run code.

Within each server are two kinds of address spaces: controllers and servants. A *controller* runs system authorized programs and manages tasks, such as communication, for the server. Each server has one controller. A *servant* runs unauthorized programs, such as business applications. Depending on the workload, a server has one or more servants running at a time. When work builds up, additional servants are dynamically started to meet the demand.

Note: The location service daemon, node agent and JMS Server are specialized servers and have no servants.

Here is a quick breakdown of the different server types on your system:

Unmanaged (base) application server
The application server set up during base configuration that hosts your Java applications.

Managed (network deployment) application server
The application server set up during network deployment configuration that hosts your Java applications.

Location service daemon
A server which is the initial point of contact for client requests in either configuration.

JMS server
Hosts the WebSphere for z/OS JMS function, which controls the MQ broker and queue manager in either configuration.

Deployment Manager
A specialized Application Server that hosts the Administrative Console application and provides cell-level administrative function in a network deployment configuration.

Node agent
Provides node-level administrative function in a network deployment configuration.

Appendix A, “Default server values for WebSphere Application Server for z/OS V5”, on page 217 lists the default servers for each configuration and their corresponding names.

Note: The ″Server name″ is the server long name used in the HFS path. The ″Server short name″ is the platform-specific native alias. Every element of the configuration (servers, clusters, nodes and cells) has one of each.

A *cluster* is a *logical grouping* of like-configured servers. Clusters exist to promote scalability and availability; workload balancing occurs across the servers in a cluster. Clusters allow you to partition workloads into separate servers while still referring to them as a single unit. This is handy in cell environments, where, while each system in the cell might run a like-configured server, clients outside the cell collectively address the servers as a single cluster. The client does not know

specifically which server is actually processing its work; in fact, different servers in the cell may, due to workload balancing, process subsequent work requests from the client.

A node contains servers which may be part of a cluster. The cluster may span nodes as long as all involved nodes are in the same cell.

Here is a quick breakdown of clusters, nodes and cells:

cluster
A logical collection of like-configured servers. A cluster can span nodes and systems within the same cell.

node
A logical collection of managed servers on a particular system in the cell. A node can contain servers that are part of clusters that span other nodes, but the node itself is confined to a single system.

cell
A logical collection of nodes from the network of systems. You can have more than one cell on each system in a network (and, hence, more than one cell in a network), but a cell can't span networks. The cell is the largest unit of organization.

To help you understand the interaction between servers, clusters, nodes and cells, here is a diagram depicting various configurations you can set up in your network deployment sysplex:

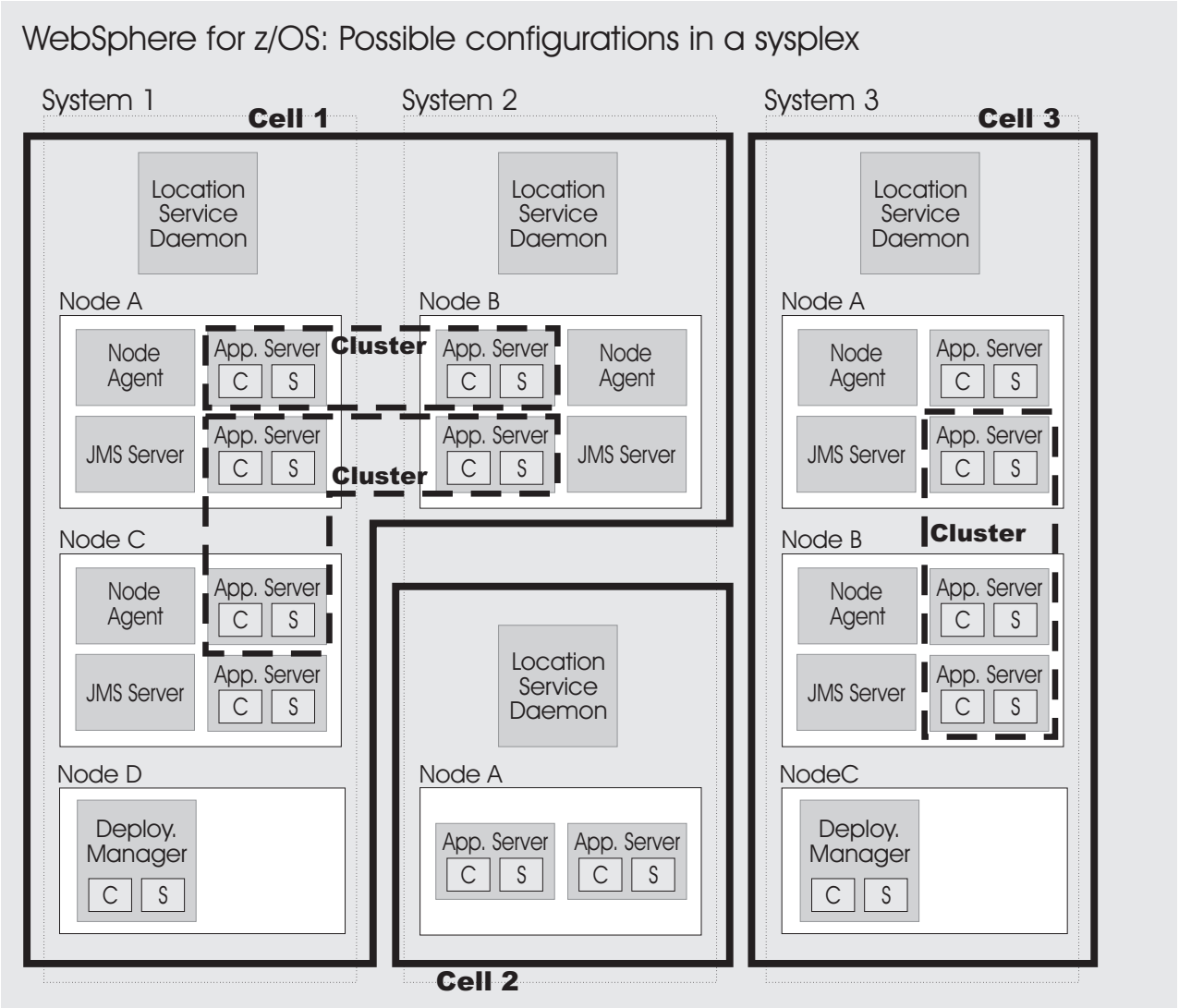


Figure 3. Various configurations of WebSphere for z/OS

In Figure 3, cells 1 and 3 depict network deployment configuration cells and cell 2 is a base configuration cell.

The WebSphere for z/OS HTTP internal transport

Web components, which are known as Web applications, may consist of any combination of the following parts:

- One or more Java servlets
- Any other Java classes that act as utility classes in support of the servlets
- Static files such as HTML pages and GIF or JPEG images
- JavaServer Pages (JSPs) that format dynamic output

To enable Web applications for use, your Web-serving environment requires an HTTP transport (to receive HTTP requests from a network of browsers using the HTTP access protocol) and an execution environment (to interpret the inbound request and run the appropriate servlet, based on the contents of the inbound request). The WebSphere for z/OS Java server includes a choice of two HTTP transports and execution environments:

1. The HTTP internal transport and/or HTTPS internal transport in combination with the Web container in the Java server, or
2. The IBM HTTP server for z/OS in combination with the WebSphere for z/OS Local Redirector plug-in shipped with the WebSphere for z/OS product, and/or Web container in the Java server.

Recommendation: Because the application server is not designed to directly host static content (for example, HTML pages), the recommended configuration places a Web server with a plug-in in front of the application server. See Figure 4.

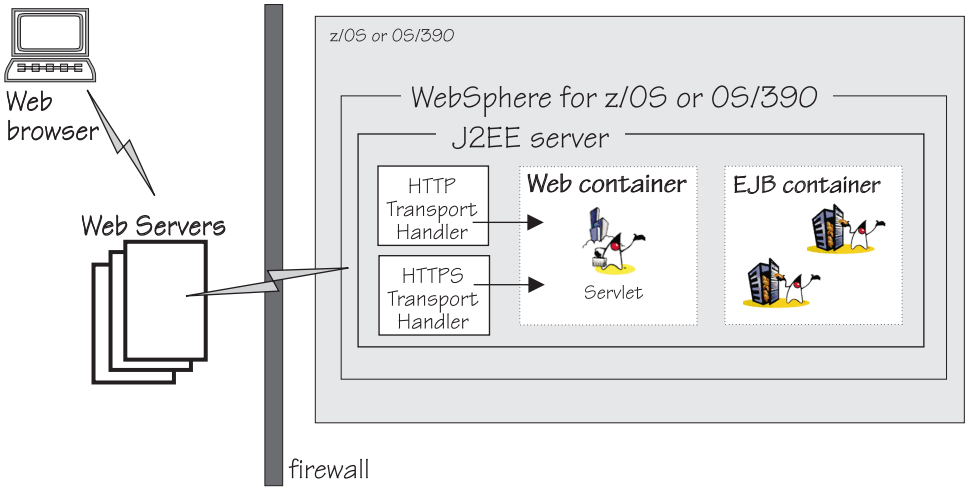


Figure 4. Recommended configuration of the Web-serving environment on z/OS or OS/390

Web applications running in the Web container have direct access to resources on z/OS or OS/390, or can access them through Enterprise beans running in any WebSphere for z/OS Java server. Web applications use the RMI/IIOP protocol to access Enterprise beans running in Java servers on the same or different z/OS or OS/390 images.

For more information about deploying Web applications, see the assembling applications information in the WebSphere Application Server InfoCenter, which you can access via the WebSphere Application Server for distributed platforms library Web page.

Creating a plan to implement WebSphere for z/OS

Successful deployment of WebSphere for z/OS requires that you plan for changes to your z/OS system and plan for the WebSphere for z/OS installation and customization. This section provides a checklist for tasks you should consider.

Steps for creating your implementation plan

To get started, plan to build all WebSphere for z/OS run-time servers on one system, then replicate them on other systems as you expand into a cell. This procedure guides you through initial planning and implementation of WebSphere for z/OS on a monoplex. Then it guides you through setting up your application development and client environments. Finally, the procedure guides you through planning for optional advanced system configurations.

Before you begin: We assume you have a z/OS system on which you will implement WebSphere for z/OS.

Perform the following steps to implement your plan:

1. Plan WebSphere for z/OS on a monoplex or a single system in a multi-system sysplex. Check off each item as you complete it:

Check off	Item	For more information, see . . .
	Determine the skills you need.	"Determining your skill needs" on page 11
	Determine WebSphere for z/OS system requirements.	"Determining WebSphere for z/OS system requirements" on page 12
	Understand and plan for customization changes you will need to do for your TCP/IP network.	"Updating your TCP/IP network" on page 16
	Understand security options and prepare for securing your system.	"Setting up security" on page 18
	Set up workload management environments for WebSphere for z/OS run-time servers.	"Setting up workload management (WLM)" on page 29
	Customize resource recovery services for use by WebSphere for z/OS.	"Recommendations for resource recovery services" on page 33
	Plan for your performance and monitoring systems.	"Guideline for RMF and other monitoring systems" on page 34
	Follow recommendations for memory utilization.	"Recommendations for using memory" on page 35
	Plan and define your problem diagnosis procedures.	"Planning for problem diagnosis" on page 36
	Consider automatic restart management before you install WebSphere for z/OS.	"Tip on automatic restart management (ARM)" on page 44

2. Install and customize WebSphere for z/OS (perform one of two tasks).

Check off	Item	For more information, see . . .
	Install and customize WebSphere for z/OS for the first time.	Chapter 3, "Installing and customizing your first run-time", on page 45
-or-		
	Upgrade code levels to a new service level of WebSphere for z/OS.	The migration information in the z/OS view of the WebSphere Application Server InfoCenter, which you can access via the WebSphere for z/OS library Web site, and Chapter 6, "Installing new releases and maintenance levels of WebSphere for z/OS", on page 211

3. Perform various post-installation tasks.

Check off	Item	For more information, see . . .
	Plan and define your system backup procedures.	"Guidelines for backup of the WebSphere for z/OS system" on page 123
	Plan and define your software service procedures.	"Overview of product service" on page 123
	Implement automation controls and set up automatic restart management for WebSphere for z/OS, if desired.	"Setting up automation and automatic restart management" on page 125

4. Plan for your application development and client environments.

Check off	Item	For more information, see . . .
	Review WebSphere for z/OS requirements for application development and client environments.	The assembling applications information in the WebSphere Application Server InfoCenter, which you can access via the WebSphere Application Server for distributed platforms library Web page

5. (Optional) Plan and implement advanced system configurations.

Check off	Item	For more information, see . . .
	Plan to deploy WebSphere for z/OS in a deployment manager cell.	"Setting up WebSphere for z/OS on multiple systems in a sysplex" on page 127
	Plan to have multiple TCP/IP stacks, use connection optimization, use an IBM Network Dispatcher, or use bind-specific support.	"Implementing an advanced TCP/IP network" on page 137
	Implement advanced security controls such as SSL and Kerberos.	"Implementing advanced security" on page 141
	Set up RACF protection for DB2, if desired.	"Setting up RACF protection for DB2" on page 124
	Tune system performance.	"Implementing advanced performance controls" on page 201
	Access IMS resources: a. Use the IMS Connector for Java. b. Use the IMS JDBC Connector.	The the WebSphere Application Server InfoCenter, which you can access via the WebSphere Application Server for distributed platforms library Web page
	Access CICS resources with the CICS Transaction Gateway ECI connector.	The the WebSphere Application Server InfoCenter, which you can access via the WebSphere Application Server for distributed platforms library Web page
	Plan for testing and production systems.	Configuring your systems for test and production on page 207

6. Plan and implement release and maintenance upgrades.

Check off	Item	For more information, see . . .
	Review code upgrade methods.	Chapter 6, "Installing new releases and maintenance levels of WebSphere for z/OS", on page 211

You are done when you have checked all the applicable items.

Chapter 2. Preparing the base z/OS environment

Some z/OS function customization steps you need to do for WebSphere for z/OS can be done before you install and customize WebSphere for z/OS itself. We have put those tasks into this chapter, allowing you to segment your work.

Other z/OS function customization steps must occur along with customizing WebSphere for z/OS itself. You will find those steps in Chapter 3, "Installing and customizing your first run-time", on page 45.

In either case, this chapter gives you background information about how WebSphere for z/OS uses z/OS functions, and it provides planning guidelines and tips for implementing WebSphere for z/OS.

Determining your skill needs

In assembling your project team, you should consider the skills you need to implement WebSphere for z/OS. Below are the function skill areas you need depending on your desired environment and where you are in the deployment process.

Skills to get started

You can get started with WebSphere for z/OS by assembling a team with system skills in the following areas:

For basic configurations:

- z/OS UNIX System Services and the hierarchical file system (HFS) — to set up a functional HFS and UNIX environment
- eNetwork Communications Server (TCP/IP) or equivalent — to configure connectivity for WebSphere for z/OS clients and servers
- Resource recovery services (RRS) — to implement resource recovery services and support two-phase commit transactions
- Security Server (RACF), or the security product you use — to authenticate WebSphere for z/OS clients and servers, and authorize access to resources
- Workload management (WLM)
- SMP/E and JCL
- DB2

For advanced configurations:

When dealing with advanced configurations, you need all the same skills as for basic configurations, plus the following:

- System logger — to set up logstreams for RRS and the WebSphere for z/OS error log
- Parallel sysplex — to implement multi-system configurations
- CICS
- IMS
- MQ

Skills for a production environment

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

- Automatic restart management (ARM)

- System Automation, if you have it installed, or whichever automation you prefer to use
- Syplex, if you plan to use WebSphere for z/OS in a cell
- Secure Sockets Layer (SSL) or Kerberos, if you plan to have security in a distributed network
- Advanced environment SSL if you enable security
- RMF or other performance measurement systems
- Webserver, if you plan to support HTTP clients
- Java

Skills for an application development environment

For the application development environment, you need the following skills:
- Object-oriented application programming skills
- If you plan to use Java-based components, knowledge of the Java infrastructure and the Enterprise JavaBeans (EJB) component architecture
- Windows skills

Determining WebSphere for z/OS system requirements

The following are system requirements for WebSphere for z/OS.

z/OS hardware requirements

The hardware requirements for this product are any hardware that supports OS/390 Version 2 Release 10 or any version of z/OS. However, there are significant performance advantages for those applications doing floating point arithmetic if the machine has binary floating point hardware, such as S/390 Parallel Enterprise Server–Generation 5 and later systems.

The LPAR in which the WebSphere for z/OS run-time and initial application servers run requires a minimum of 512 MB of real storage. You may need to increase the real storage size depending on the size and number of application servers you deploy. In addition, you may want to increase your JES spool space if you use WebSphere for z/OS tracing options to the SYSPRINT DD dataset.

You should plan on an extra 3390-3 volume for the distribution and configuration HFS datasets.

Recommendation: We recommend you increase your paging subsystem by one 3390-3 volume if your storage is constrained.

z/OS software requirements for WebSphere for z/OS

You must install, enable, and configure the following z/OS or OS/390 elements, features, and components. Consult the Program Directory or PSP bucket for the required corrective service.

- OS/390 Version 2 Release 10 or z/OS configured as a sysplex (at minimum, you need a monoplex). For details, see *z/OS MVS Setting Up a Sysplex*, SA22-7625.

  Note: If you want to utilize the dynamic application environment functionality of WebSphere Application Server for z/OS V5, you need to run at least z/OS Version 1.2.
- z/OS UNIX System Services (z/OS UNIX) with a hierarchical file system (HFS). For details, see *z/OS UNIX System Services Planning*, GA22-7800.

- eNetwork Communications Server (TCP/IP) or equivalent. In this manual, we refer to eNetwork Communications Server, but you may substitute an equivalent product. For details, see *z/OS Communications Server: IP Migration*, GC31-8773.
- Workload management (WLM) set up in goal mode. For details, see *z/OS MVS Planning: Workload Management*, SA22-7602.

  Note: If you are not running z/OS V1.2 or above with the WLM-DAE support PTF (APAR OW54622), you need to complete some additional steps for WLM. See "Setting up workload management (WLM)" on page 29 for details.
- z/OS system logger. For details, see *z/OS MVS Setting Up a Sysplex*, SA22-7625.
- Resource recovery services (RRS). For details, see *z/OS MVS Programming: Resource Recovery*, SA22-7616.
- A security product such as SecureWay Security Server (RACF). In this manual we refer to Security Server in examples, but you may substitute an equivalent security product. For details, see *z/OS Security Server RACF Migration*, GA22-7690.
- Cryptographic Services System SSL, a component of Cryptographic Services Base, an element of z/OS. For details, see *z/OS System Secure Sockets Layer Programming*, SC24-5901.
- IBM Developer Kit for OS/390 Java 2 Technology Edition Version 1.1, an element of WebSphere for z/OS, but also available separately. The SDK level supported by this product is 1.3.1.

  Note: Later releases of the IBM Developer Kit for OS/390 Java 2 Technology Edition are not supported.

Regarding optional functions, consult the following table:

Table 1. Software requirements for optional functions

If you plan to use . . .	Then you need . . .	Notes . . .
Kerberos security	OS/390 SecureWay Security Server Network Authentication and Privacy Service for OS/390	For OS/390 V2R10 and z/OS, this support is part of SecureWay Security Server.
EJB roles to secure Web components or enterprise beans	z/OS V1R2 Security Server (RACF) or equivalent	For releases of z/OS earlier than z/OS V1R2, install the appropriate APAR for this support.
Java Message Service (JMS)	• Integrated JMS provider • Either of the following: – MQ stack from WebSphere Application Server for z/OS V5 – Full function MQ using MQ 5.3.1	MQSI, a pub sub product that supports z/OS, does not support new function like MDBs.
WebSphere for z/OS IMS Connect V8 support	IMS/TM 6.1.0	
WebSphere for z/OS CICS Transaction Gateway 5.0.1 support	CICS/TS 1.3	
DB2	DB2 V7.1	

Table 1. Software requirements for optional functions (continued)

If you plan to use . . .	Then you need . . .	Notes . . .
DB2 SQLJ	DB2 V7.1 PTF UQ59527	
Connectors		
	For the CICS Transaction Gateway ECI connector: • CICS Transaction Gateway V5.0.1 • CICS Transaction Server V1.3 • WebSphere Studio Application Developer IE V5.0 • WebSphere for z/OS Administrative Console Version 4.01.011	For configuration details, see the the WebSphere Application Server InfoCenter, which you can access via the WebSphere Application Server for distributed platforms library Web page.
	For the IMS Connector for Java: • IMS Connect for z/OS V2.1 • IMS V8 • WebSphere Studio Application Developer IE V5.0.1 • WebSphere for z/OS Administrative Console Version 4.01.011	For configuration details, see the the WebSphere Application Server InfoCenter, which you can access via the WebSphere Application Server for distributed platforms library Web page.
	For the IMS JDBC Connector: • IMS V8 • WebSphere for z/OS Administrative Console Version 4.01.011	For configuration details, see the the WebSphere Application Server InfoCenter, which you can access via the WebSphere Application Server for distributed platforms library Web page.

Software requirements for developing WebSphere for z/OS applications

If you are developing Java components, you need the following on your workstation:

Table 2. Software requirements for application components

Application component	Software to use
Enterprise beans	**For development:** One of the following: • The IBM WebSphere Studio Application Developer V5, which is the preferred method of deploying applications. • WebSphere Studio Application Developer Integration Edition • Non-IBM tools, such as JBuilder or Visual Cafe, for application development. Use the documentation for those products to determine hardware and software requirements.
	For testing: One of the following: • IBM WebSphere Studio Application Developer V5 • WebSphere Studio Application Developer Integration Edition • This combination of products: – IBM or Sun Microsystems Java 2 Standard Edition (J2SE) Software Development Kit (SDK) V1.3.1 – WebSphere Application Server, V5 (Optional) DB2 Universal Database Version 7.1, required only for testing beans that require the use of a persistent datastore.
	For assembly: One of the following: • The WebSphere Application Server, V5, Application Assembly tool (read the recommendations following this table for further information about using the Application Assembly tool) • The IBM WebSphere Studio Application Developer V5
	For installation in a Java server: • The WebSphere for z/OS Administrative Console
Servlets and JavaServer Pages (JSPs)	**For development and testing:** One of the following: • WebSphere Studio Application Developer, which is the preferred method of deploying applications. • The IBM WebSphere Studio Application Developer V5 • IBM or Sun Microsystems Java 2 Standard Edition (J2SE) Software Development Kit (SDK) V1.3.1
	For assembly: One of the following: • The WebSphere for z/OS Application Assembly tool (read the recommendations following this table for further information about using the Application Assembly tool) • The IBM WebSphere Studio Application Developer V5
	For installation in a Java server: The WebSphere for z/OS Administrative Console

Recommendations:

- The preferred method of deploying applications is to use the IBM WebSphere Studio Application Developer, which you also use to develop and test beans, servlets, and JSPs. This product enables developers to fully test entity and session beans, including JNDI lookups, remote method calls, and method calls on the home interface. It also has a servlet engine, so you can serve up servlets and JSPs to a Web browser as if they were going through an HTTP and Application Server.

Additionally, WebSphere Studio Application Developer enables you to automatically package servlets or JSPs into Web application archive (WAR) files. (If you use other tools, you might have to create the WAR files manually.)

- If you access the Administrative Console using Netscape on Windows, you must use Netscape Version 4.7.9 or later. Go to `http://www.ibm.com/software/webservers/appserv/doc/v50/prereqs/prereq50.html` for more informaiton.

Updating your TCP/IP network

WebSphere for z/OS follows the CORBA standard, Internet Inter-ORB Protocol (IIOP), for communications. Accordingly, you must consider changes to your TCP/IP network and modify the TCP/IP configuration.

This section provides background information about changes you will need to make to your Domain Name Server (DNS) and TCP/IP. The actual steps to perform are in the customized instructions provided by the customization dialog (see "Using the customization dialog" on page 49).

TCP/IP and DNS port specifications

Here is a chart of TCP/IP and DNS port defaults you will encounter when configuring WebSphere Application Server for z/OS V5. Please see Appendix B, "z/OS port assignments", on page 219 for a complete listing of port assignments.

Table 3. TCP/IP and DNS port specifications

Server	Default port	Same value as WebSphere Application Server Advanced Edition?
Location service daemon	5655	
Application Server		
IIOP	(dynamic)	—
IIOPS	(dynamic)	—
HTTP	9080	Yes
HTTPS	9443	Yes
Node agent		
IIOP	2089	No
IIOPS	(dynamic)	—
HTTP	9080	Yes
HTTPS	9443	Yes

Tips on TCP/IP and WebSphere for z/OS

Consider the following for your TCP/IP network on z/OS.

- You can get started with a simple Domain Name Service (DNS) name server and a single z/OS image, but you should design your initial configuration with growth in mind. You may, for instance, intend to expand your business applications beyond the monoplex to a full sysplex configuration for performance reasons or to prevent a single point of failure. Several considerations come to bear here.

  Several DNS implementations and network router implementations allow the use of a generic location service daemon IP name while dynamically routing

network traffic to like-configured servers. If you intend to expand your system beyond a monoplex, it might be worthwhile to use one of these implementations from the start. Non-round-robin DNS name servers limit your ability to expand without retrofitting a name server that allows dynamic network traffic routing.

Recommendation: The IBM-recommended implementation if you are running in a sysplex is to set up your TCP/IP network with Sysplex Distributor. This makes use of dynamic virtual IP addresses (DVIPAs), which increase availability and aid in workload balancing. For more information, see "Implementing an advanced TCP/IP network" on page 137.

Beyond Sysplex Distributor, you have your choice of the following DNS and router implementations on or off z/OS:

- Non-round-robin DNS name servers.
- Round robin DNS name servers.
- Connection optimization, a technique used by z/OS that uses DNS and workload management (WLM). WebSphere for z/OS uses connection optimization to prevent a single point of failure. To use connection optimization, you must run the DNS name server on z/OS. For more information, see "Connection optimization" on page 138.
- Network routers, such as the IBM Network Dispatcher. For more information, see "IBM Network Dispatcher" on page 139.

- Select the location service daemon IP name **carefully**. You can choose any name you want, but, once chosen, it is difficult to change.

  You must define the location service daemon host IP name during installation and customization, before you start the location service daemon. Use the location service daemon IP name you chose. See the WebSphere variables in the Administrative Console or the InfoCenter.
- Select the port for the location service daemon server and do not change it. Object references also include the port—if you change the port, you can no longer access existing objects. WebSphere Application Server for z/OS V5 uses port 5655 as a default.

  Note: If you install WebSphere Application Server for z/OS V5 on a system that already contains WebSphere Application Server V4.0.1 for z/OS and OS/390, double-check your current location service daemon port value (default for WebSphere Application Server V4.0.1 for z/OS and OS/390 is 5555).
- You can set location service daemon port numbers and IP addresses, found in sysplex-level WebSphere variables, to enable you to configure your servers behind a firewall. If you need to use the Internet Inter-ORB Protocol (IIOP) through a firewall, ensure that your firewall supports IIOP.

  To configure the ports for a firewall, set up the following WebSphere variables:
  - protocol_iiop_daemon_listenIPAddress
  - protocol_iiop_daemon_port (default value is 5655)
  - protocol_iiop_daemon_port_ssl (default value is 5656)

  Notes:

  1. When recovering a server somewhere other than its configured system, ensure that you configure it with a unique port to avoid a conflict with a port that is in use on the system on which it is recovering.

2. When configuring different WebSphere Application Server products on the same system, realize that WebSphere Application Server for z/OS V5 supports dynamic allocation of IIOP ports while WebSphere Application Server Advanced Edition does not.

HTTP and HTTPS ports are found in individual servers under the Web container transports.

Notes:

1. Watch for HTTP transport port conflicts if you previously installed WebSphere Application Server V4.0.1 for z/OS and OS/390.

Also ensure you set up the following properties on servers that require them in the Administrative Console:

- SSL Firewall port
- Web container HTTP transport
- Web container HTTPS transport

Note: See the Administrative Console and the InfoCenter for more information on the WebSphere variables and how to set their values.

- Some ports are obtained dynamically.
- Other TCP/IP-related activities include setting up NFS, WebServer (optional) and Kerberos (optional).
- If you use the DNS on z/OS, you may wish to change the refresh timer interval (-t value) associated with the named location service daemon. The -t value specifies the time (nn, in seconds) between refreshes of cell names and addresses and of the weights associated with those names and addresses. The default is sixty seconds. Reducing the -t value will shorten the lapse time required to register the location service daemon IP name with the DNS, but will also increase DNS processing overhead. In our testing, we used an interval of 10 seconds. For details, see *z/OS Communications Server: IP Configuration Reference*, SC31-8776.

Setting up security

WebSphere for z/OS supports access to resources by clients and clusters in a distributed network, so part of your security strategy should be to determine how to control access to these resources and prevent inadvertant or malicious destruction of the system or data.

These are the pieces in the distributed network that you must consider:

- You must authorize clusters to the base operating system services in z/OS or OS/390. These services include SAF security, database management, and transaction management.
  - For the clusters, you must distinguish between controllers and servants. Controllers run authorized system code, so they are trusted. Servants run application code and are given access to resources, so you should carefully consider the authorizations you give servants.
  - You must also distinguish between the level of authority run-time clusters and your own application clusters have. For example, the node needs the authority to start other clusters, while your own application clusters do not need this authority.
- You must authorize clients (users) to clusters and objects within clusters. The characteristics of each client requires special consideration:

  - Is the client on the local system or is it remote? The security of the network becomes a consideration for remote clients.
  - Will you allow unidentified (unauthenticated) clients to access the system? Some resources on your system may be intended for public access, while others need to be protected. In order to access protected resources, clients must establish their identities and have authorization to use those resources.
- Authentication is the process of establishing whether a client is valid in a particular context. A client can be either an end user, a machine, or an application. An authentication mechanism in WebSphere Application Server typically collaborates closely with a User Registry. When configuring a cell, you must select a single authentication mechanism. The choices for authentication mechanism include:
  - Simple WebSphere Authorization Mechanism (SWAM) — only on Base Application Server, not available on the Network Distributed configuration
  - Lightweight Third Party Authentication (LTPA)
  - Integrated Cryptograqphic Service Facility (ICSF)
- Information about users and groups reside in a user registry. In WebSphere Application Server, a user registry authenticates a user and retrieves information about users and groups to perform security-related functions, including authentication and authorization. Implementation is provided to support multiple operating system or operating environment-based user registries. When configuring a cell, you must select a single user registry. The user registry can be local or remote. The choices for user registry include:
  - SAF-based local registry (default)
  - Lightweight Directory Access Protocol (LDAP) — LDAP can be either a local or remote registry
  - Custom user registry — a Custom Regisity is provided by 3rd party. WebSphere provides a simple user registry sample called the FileBasedRegistrySample.

If you need to protect resources, identifying who accesses those resources is critical. Thus, any security system requires client (user) identification, also known as authentication. In a distributed network supported by WebSphere for z/OS, clients can be accessing resources from:

- Within the same system as a cluster
- Within the same sysplex as the cluster
- Remote z/OS or OS/390 systems
- Heterogeneous systems, such as WebSphere on distributed platforms, CICS, or other Java-compliant systems.

Additionally, clients may request a service that requires a cluster to forward the request to another cluster. In such cases, the system must handle delegation, the availability of the client identity for use by intermediate clusters and target clusters.

Finally, in a distributed network, how do you ensure that messages being passed are confidential and have not been tampered? How do you ensure that clients are who they claim to be? How do you map network identities to z/OS or OS/390 identities? These issues are addressed by the following support in WebSphere for z/OS:

- The use of SSL and digital certificates
- Kerberos

- Common Secure Interoperability Version 2 (CSIv2)

Because network security is not required for your initial installation and customization of WebSphere for z/OS, details on these topics are reserved for the topic Chapter 5, "Performing advanced tasks", on page 127. This current topic is designed to introduce you to WebSphere for z/OS security and allow you to make early planning decisions about system security. In Chapter 3, "Installing and customizing your first run-time", on page 45, there are specific instructions for setting up initial RACF security controls through the use the customization dialog IBM provides with the product.

The following topics describe how WebSphere for z/OS supports security. The descriptions are organized under the following subtopics:

- Authorization checking
- User identification, authentication, and network security issues
- User registries

Note: We use Security Server (RACF) as an example, but you can use an equivalent product.

Included are notes on support for security auditing and security administration.

Authorization checking

Each controller, servant, and client must have its own MVS user ID (more about user identification and authentication later). When a request flows from a client to the cluster or from a cluster to a cluster, WebSphere for z/OS passes the user identity (client or cluster) with the request. Thus each request is performed on behalf of the user identity and the system checks to see if the user identity has the authority to make such a request.

Summary of controls

Table 4 is a summary of the controls used to grant authorizations to resources. By understanding and using these controls, you can control all resource accesses in WebSphere for z/OS.

Table 4. Summary of controls and SAF authorizations

Control	Authorization
CBIND class	Access to a cluster
DATASET class	Access to data sets
DSNR class	Access to DB2
EJBROLE or GEJBROLE class	Access to methods in enterprise beans
FACILITY class (IRR.DIGTCERT.LIST and IRR.DIGTCERT.LISTRING)	SSL key rings, certificates, and mappings
FACILITY class (IMSXCF.OTMACI)	Access to OTMA for IMS access
FACILITY Class (IRR.RUSERMAP)	Kerberos credentials
GRANTs (DB2)	DB2 access to plans and database
HFS file permissions	Access to HFS files
LOGSTRM class	Access to log streams
OPERCMDS class	Start and stop clusters by location service daemon
PTKTDATA class	Passticket enabling in the sysplex

Table 4. Summary of controls and SAF authorizations (continued)

Control	Authorization
SERVER class	Access to <controller by a servant
Set OS Thread Identity to RunAs Identity	Java cluster property used to enable the execution identity for non-Java resources
STARTED class	Associate user ID (and optionally group ID) to start procedure
SURROGAT class (*.DFHEXCI)	Access to EXCI for CICS access

Table 5 is a summary of the controls used to grant non-SAF authorizations to resources.

Table 5. Summary of controls and non-SAF authorizations

Control	Authorization
LDAP access control lists	Access to LDAP user registry only

Cluster authorizations

Figure 5 shows the kinds of authorization checking WebSphere for z/OS does for aclusters.

Figure 5. Cluster authorization checking

The following explains the numbered items in Figure 5.

1. Servants must have access to profiles in the RACF SERVER class. This controls whether a servant can call authorized routines in the controller.

   controllers do not require such access control. Only authorized programs, loaded from Authorized Program Facility (APF) libraries, run in controllers.
2. Resource managers such as DB2, IMS, and CICS have implemented their own resource controls, which control the ability of clusters to access resources.

   When resource controls are used by DB2, all controllers and servants need to be granted access to the relevant resources. You can do this by using the DSNR RACF class (if you have RACF support) or by issuing the relevant DB2 GRANT statements.

   Access to OTMA for IMS access is through the FACILITY Class (IMSXCF.OTMACI). Access to EXCI for CICS is through the SURROGAT class (*.DFHEXCI).

   You can control access to data sets through the DATASET class and HFS files through file permissions.

Specifics about cluster authorization checking: To control access to WebSphere for z/OS resources:

- As a general rule, give greater authority to controllers and less authority to servants.

Table 6. Level of trust and authority for regions

Region	Level of trust and access authority
Controller	Contains WebSphere for z/OS system code. Trusted, deals with multiple users. Greater authorization. Runs APF-authorized.

Table 6. Level of trust and authority for regions (continued)

Region	Level of trust and access authority
Servant	Contains application code. Untrusted. Other than having authorization to get work and to attach to data stores, should run unauthorized.Supports the option to use Java Authorization in the servant region.

- Regarding the WebSphere for z/OS run-time clusters, the general rule is to give less authority to the location service daemon, and greater authority to the node, as explained in the table below:

Table 7. Assigning authorities to WebSphere for z/OS run-time cluster control and servants

Run-time Cluster	Region	Required Authorities
location service daemon	Control	STARTED class, access to WLM services, access to DNS, OPERCMDS access to START, STOP, CANCEL, FORCE and MODIFY other clusters
node	Control	STARTED class
	Cluster	STARTED class, READ authority to the SERVER class, OPERCMDS access to START, STOP, CANCEL, FORCE and MODIFY other servers

- Remember to protect the RRS log streams. By default, UACC is READ.
- Protect the WebSphere for z/OS properties xml files, especially if they contain passwords. For more information about the properties xml files, see .

SAF-based client authorizations

Figure 6 shows the kinds of SAF-based authorization checking WebSphere for z/OS does for clients.

Figure 6. SAF-based client authorization checking

The following explains the numbered items in Figure 6.

1. You can use the CBIND class in RACF (optional) to restrict a client's ability to access clusters, or you can deactivate the class if you do not require this kind of access control. There are two types of profiles WebSphere for z/OS uses in the CBIND class:
   - One that controls whether a local or remote client can access clusters. The name of the profile has this form:

     `CB.BIND.`**`cluster_name`**

     where *cluster_name* is the name of the cluster.
   - One that controls whether a client can use components in a cluster. The name of the profile has this form:

     `CB.`**`cluster_name`**

     where *cluster_name* is the name of the cluster.

   Note: When you add a new cluster, you must authorize all systems management user IDs (for example, WSADMIN) to have read access to the CB.**cluster_name** and CB.BIND.**server_name** RACF profiles.

   Example: WSADMIN needs read authority to the CB.BBOC001 and CB.BIND.BBOC001 profiles:

```
PERMIT CB.BBOC001      CLASS(CBIND) ID(WSADMIN) ACCESS(READ)
PERMIT CB.BIND.BBOC001 CLASS(CBIND) ID(WSADMIN) ACCESS(READ)
```

2. EJBROLE and SOMDOBJS classes:
   - Use the EJBROLE (or GEJBROLE) class in RACF to control a client's access to enterprise beans. There are two distinct sets of tasks that are required to protect an application using EJB roles.
     a. The security administrator must define the roles and set up access rights in RACF.
        - Define a profile name using the EJBROLE (or GEJBROLE) class.

          Example:

          ```
          RDEF EJBROLE role_name UACC(NONE)
          ```

          where *role_name* matches the security role attribute specified either in the jar file or for the application. A role name cannot contain blanks, and cannot exceed 245 characters. Role names, however, may be in mixed case.
        - Create membership in the role by granting MVS user IDs or groups permission to the defined EJBROLE profile.

          Example:

          ```
          PERMIT role_name CLASS(EJBROLE)  ID(mvsid_gp) ACCESS(READ)
          ```
        - Activate and RACLIST the EJBROLE class.

          Example:

          ```
          SETROPTS CLASSACT(EJBROLE)
          SETROPTS RACLIST(EJBROLE) GENERIC(EJBROLE)
          ```
     b. The application assembler must assign method permissions to the bean or method using the Application Assembly Tool.
        - Define the roles relevant to the application. These role names must match the profile names assigned to RACF.
        - Once defined, the role can be assigned to access an application (as a method permission).
        - After the application assembly is complete, the application must be reinstalled using the Administration application.

        For details about assigning method permissions, refer to the WebSphere Application Server InfoCenter. Topics relating to assigning method permissions are located in the Application section.
   - Use the SOMDOBJS class in RACF to control a client's access to CORBA objects. Profile names in SOMDOBJS have the form:

     cluster_name.home.method

     where

     cluster_name
     Is the cluster name. It must be 8 characters or less.

     home
     Is the home name. It must be 192 characters or less.

     method
     Is the method name. It can be up to the length of the remainder of 244 minus the sum of the cluster and home name lengths.

     Example: If the cluster name is 8 characters, and the home name is 128 characters, the method name can be 108 (244 – (8 + 128)).

If a method is protected by SOMDOBJS and:

- A client program is using the method to update an attribute of an object, give the client UPDATE authorization for the method.
- A client program is using the method to read an attribute of an object, give the client READ authorization for the method.

All names are folded into uppercase characters, regardless of how you enter them. Thus, there is no difference between MY_server.MY_home.MY_method and MY_SERVER.MY_HOME.MY_METHOD.

In addition to the RACF SOMDOBJS definitions, you must specify method-level access checking through the WebSphere for z/OS Administration application. Check the box for method-level access checking when you define your application's container.

3. Resource managers such as DB2, IMS, and CICS have implemented their own resource controls, which control the ability of clients to access resources.

   When resource controls are used by DB2, use the DSNR RACF class (if you have RACF support) or by issuing the relevant DB2 GRANT statements.

   Access to OTMA for IMS access is through the FACILITY Class (IMSXCF.OTMACI). Access to EXCI for CICS is through the SURROGAT class (*.DFHEXCI).

   You can control access to data sets through the DATASET class and HFS files through file permissions.

User identification, authentication, and network security issues

Proper security for any system requires that users or programs identify themselves and prove they are who they claim to be (authenticate themselves). Figure 7 shows the kinds of user identification and authentication WebSphere for z/OS uses within and across systems.

Figure 7. Identification and authentication

The following explains the numbered items in Figure 7.

1. Local clients and clusters use their user IDs to identify themselves when requesting a service. WebSphere for z/OS uses a transportable form of the user's Accessor Environment Element (ACEE), called a RACO, for local clients and clusters running in the same sysplex. The RACO is used throughout the WebSphere for z/OS system and ensures that any task is performed under the requestor's identity. No authentication is required because the user's identity is already established by the operating system. Just like other OS/390 applications, WebSphere for z/OS uses the operating system to keep track of the user identities and makes calls to the security service during the execution of a piece of work.
2. Unless you can be sure all messages exchanged flow exclusively within a trusted network, authenticity of clients and clusters, message confidentiality, and message integrity become important issues. A client may want to be sure that it is receiving a service from a legitimate cluster and a cluster may want to be sure who the client is. Each party also wants to be sure that messages exchanged are protected from tampering or snooping by a malicious third party, so security in the transportation medium (message protection) is a concern. WebSphere for z/OS provides several authentication mechanisms, some of which involve message protection. You need to decide, based on the

nature of your network, which authentication mechanism you need. WebSphere Application Server for z/OS V5 supports the following authentication mechanisms:

- The Simple WebSphere authentication mechanism (SWAM) is intended for simple, non-distributed, single application server run-time environments. The single application server restriction is due to the fact that SWAM does not support forwardable credentials. If a servlet or enterprise bean in application server process 1, invokes a remote method on an enterprise bean living in another application server process 2, the identity of the caller identity in process 1 is not transmitted to server process 2. What is transmitted is an unauthenticated credential, which, depending on the security permissions configured on the EJB methods, can cause authorization failures. Since SWAM is intended for a single application server process, single signon (SSO) is not supported. The SWAM authentication mechanism is suitable for simple environments, software development environments, or other environments that do not require a distributed security solution.
- Lightweight Third Party Authentication (LTPA) is intended for distributed, multiple application server and machine environments. It supports forwardable credentials and single signon (SSO). LTPA can support security in a distributed environment through cryptography. This supports permits LTPA to encrypt, digitally sign, and securely transmit authentication-related data, and later decrypt and verify the signature. The Lightweight Third Party Authentication (LTPA) protocol enables the WebSphere Application Server to provide security in a distributed environment using cryptography. If you need to interoperate with network distributed servers, you will need to use LTPA.
- Integrated Cryptographic Service Facility (ICSF) uses hardware encryption available on zSeries processors. ICSF also generates security tokens for authenticated users which can be propagated over to other servers. The main advantage of ICSF is that the keys are stored in secure hardware and only the key label is provided.

3. Within the sysplex, all security protocols (except for RACO) are supported between clients and clusters within the sysplex. Additionally, PassTickets are supported, in which the client's user ID is used for identification and a PassTicket for authentication. A PassTicket is a one-time-use password that is dynamically generated.

   Because communications within a sysplex flow directly over a protected network, WebSphere for z/OS avoids the overhead of message encryption for these communications. In other words, when systems in a sysplex are directly connected, WebSphere for z/OS determines that the communication is guaranteed to be secure, and does not use encryption.

When a client connects to a cluster, part of the connection includes a negotiation between the client and cluster about what security protocol is to be used. This is an advance topic. Details about security protocol negotiation are in the topic "How clients and clusters negotiate security protocols" on page 150.

Specifics about identification and authentication

For identification, each controller and servant start procedure must have its own user ID and you must define it in the STARTED class. Controllers are trusted, while servants are not—we explain that in "Authorization checking" on page 20. Because you should give differing resource authorizations to each, you should give differing user IDs to controllers and servants.

Additional user IDs are required for installation. We provide the definitions for these user IDs in our RACF sample. See the customized instructions produced when you run the customization dialog.

- User IDs for controllers and servants.
- A user ID for the Installation Verification Test (IVT) and its application cluster. Our RACF sample uses WSIVT.
- A user ID called WSADMIN used by the Administration application.
- A default local and remote user ID associated with each cluster through the Administrative Console. We use WSGUEST.

Necessary user IDs and RACF definitions for the WebSphere for z/OS run time are provided by our RACF sample.

Regarding authentication, an operator starts a cluster by using the START command and the controller start procedure. Authentication of the start procedure's user ID is made by virtue of the fact that an operator started the start procedure—that is, no password is required. If you want to restrict an operator's ability to start clusters, do so through the OPERCMDS class in RACF.

Setting permission for files created by applications

Files created by applications running in the servant will have permission bits set according to the default umask. To change the default umask for the servant, specify the _EDC_UMASK_DFLT environment variable in the JCL procedure for the servant.

On the JCL EXEC statement, specify:

```
PARM='ENVAR("_EDC_UMASK_DFLT=xxx")
```

where xxx is the umask value to use.

Recommendation: A umask value of 007 will cause files to be created with permission bits set to 770. This is the IBM recommended value.

Note: See the following documents for more information:

- *z/OS Language Environment Programming Reference*, SA22-7562, for more information on ENVAR.
- *z/OS C/C++ Programming Guide*, SC09-4765, for more information on how to change the UMASK defaults.
- *z/OS UNIX System Services Command Reference*, SA22-7802.

Security auditing

Security auditing is handled in the usual way by the security product. WebSphere for z/OS uses the System Authorization Facility (SAF), which provides an auditing mechanism consistent with other functions in z/OS or OS/390.

Security administration

Security administration should be handled in the usual way by the security product.

Choosing the system security you need

Determine the security you need and the components you must install and customize. You need to determine your security based on your application, the interaction between clusters, and network topology before you decide which security mechanisms best fit your needs.

Steps for choosing the system security you need

Before you begin: You need to know how WebSphere for z/OS uses the underlying security systems during run time. "Setting up security" on page 18 provides an overview of WebSphere for z/OS security.

Follow these steps to choose the security you need:

1. Decide whether or not your applications require protection.

   If your applications do not exchange confidential data and the identities of participants are not required, then you can avoid most security controls and ignore the rest of this topic.

   Note: You must enable clusters to allow unauthenticated requests through the Administrative Console and set up a z/OS or OS/390 user ID that will be used to process unauthenticated requests through RACF.

---

2. If your applications operate in an untrusted network and they deal with confidential or mission-critical data, then you should choose one of the security mechanisms that support message integrity and/or confidentiality (Table 8).

Table 8. Recommended security mechanisms based on your trust in the network

	Non-SSL Security			SSL-based Security[2a]			
Type of network	**local**	**Pass Ticket**	**User ID/ Pass– word**	**Basic Auth– tication**	**Kerb– eros**	**Client certifi– cates**	**Aserted identity**
Trusted	X	X	X	X	X	X	X[2b]
Untrusted		[2c]	[2d]	X	X	X	

Notes:

a. While SSL generally causes encryption to be done, the level of encryption is negotiated by cluster and client, and integrity of the messages without confidentiality is a possible outcome. If you want to ensure the confidentiality of messages, specify this while setting up the cluster. See "Setting up SSL security for WebSphere for z/OS" on page 151.

b. The management of asserted identities, for both the CsIv2, and zSAS implementations, requires trust to be conferred administratively on intermediate clusters..

c. Generally, communication within a sysplex is protected through an XCF connection. Because PassTicket security is used only among members of a sysplex, the configuration of the rest of the network is not relevant.

d. **Never** send user IDs and passwords over an untrusted network. Note that when security is enabled the Administrative Console connects from the workstation to WebSphere for z/OS through user ID and password.

---

3. If your application has a cluster component (enterprise beans) that issues requests to remote clusters, consider a security mechanism that provides for an

authenticated identity to be transmitted to the remote clusters. Some mechanisms enable the client identity to be propagated (delegated) to a remote cluster and some mechanisms transmit the intermediate cluster's identity (Table 9).

Table 9. Recommended security mechanisms based on the need to propagate a user identity

	Non-SSL Security			SSL-based Security			
Type of propagation	**local**	**Pass Ticket**	**User ID/ Pass– word**	**Basic Auth– tication**	**Kerb– eros**	**Client certifi– cates**	**Asserted identity**
Cluster can forward client identity	**X**	**X**			**X**		**X**

4. Finally, determine the type of security mechanism to use according to the software configuration you have and the type of client that is interacting with your clusters (Table 10).

Table 10. Recommended security mechanisms based on the software configuration and client characteristics

	Non-SSL Security			SSL-based Security			
Client characteristics	**local**	**Pass Ticket**	**User ID/ Pass– word**	**Basic Auth– tication**	**Kerb– eros**	**Client certifi– cates**	**Asserted identity**
On the same z/OS or OS/390 system	**X**						
In the same sysplex		**X**	**X**	**X**	**X**	**X**	**X**
Registered in a remote shared RACF database			**X**	**X**	**X**	**X**	**X**
Registered in a remote RACF database that is not shared					**X**	**X**	
WebSphere Application Server Advanced Edition V4.0				**X**[4a]			
WebSphere Application Server Enterprise Edition (distributed) C++						**X**	
WebSphere Application Server Enterprise Edition (distributed) Java				**X**			
CICS						**X**	
OEM ORBs						**X**	

Note:

a. Using SSL basic authentication with WebSphere Application Server Advanced Edition is limited to the interaction between a client (or cluster) and a WebSphere for z/OS cluster. A WebSphere for z/OS client (or cluster) cannot use SSL basic authentication in its interaction with a WebSphere Application Server Advanced Edition cluster.

You can now implement the security controls for the components you chose.

Example of choosing system security

Example: This is an example of how you would consider selecting security mechanisms for a system.

In this example, you deploy two Java clusters (WSSRV1 and WSSRV2) in a sysplex. Clients communicate with the system through WSSRV1 and WSSRV1 propagates client identities to WSSRV2 across the sysplex, which is secure. Clients run on WebSphere Application Server Enterprise Edition (distributed) and their interaction with the sysplex is on a network that is not trusted. The data the application uses must be protected and kept confidential.

1. Since you must protect the confidentiality of the data and know the client identities, your first decision is clear: since your network is untrusted, you must use a security mechanism that supports message integrity and confidentiality (see Table 8 on page 27).
2. Your application requires that the client identity be propagated to other clusters. You may use PassTicket, asserted identities, CSIv2 GSSUP authentication, or Kerberos (see Table 9 on page 28).
   - PassTicket security is generally the simplest mechanism to set up within a sysplex, but is restricted in that an address space can only have one PassTicket per second.
   - For both the CSIv2 and zSAS implementations of asserted identity, security requires that the client's MVS identity be defined on both MVS systems. You must define SSL certificates and key rings for WSSRV1 and WSSRV2 through RACF. Also, you must define a trust relationship between WSSRV1 and WSSRV2 by giving WSSRV1 RACF CONTROL authority for the CB.BIND.WSSRV2.* profile.
   - Choose CSIv2 GSSUP authentication for network interactions because WebSphere Application Server Enterprise Edition (distributed) supports that security mechanism.
   - zSAS Kerberos is scalable and delegates Kerberos network identities securely. However, you must install and configure Kerberos and SSL, which is a significant task.

   You choose PassTicket security because you know your application will have a low volume of transactions and you want to minimize security tasks and administration.
3. Finally, you choose zSAS SSL basic authentication for network interactions because you will have clients or servers that will be running prior releases of WebSphere. These releases do not support CSIv2 security. A z/OS server may be configured to support both CSIv2 and zSAS security.

In this example, you would define PassTicket and SSL Type 1 (basic authentication) for WSSRV1 and PassTicket security for WSSRV2.

Setting up workload management (WLM)

WebSphere for z/OS uses the workload management (WLM) function in z/OS to manage workloads. This section helps you get started and is sufficient to get a functioning WebSphere for z/OS system. Advanced workload management topics are in Chapter 5, "Performing advanced tasks", on page 127.

Setting up workload management (WLM) in goal mode

WebSphere for z/OS requires that z/OS run workload management in goal mode. If your system runs in compatibility mode, you must implement goal mode. For details on workload management, see *z/OS MVS Planning: Workload Management*, SA22-7602.

Setting up workload management for run-time servers

In addition to setting up workload management in goal mode, you need to define workload management policies for WebSphere for z/OS servers and your business application servers. This section discusses specifics for the run-time servers. For details on workload management and business applications, see the assembling applications information in the WebSphere Application Server InfoCenter, which you can access via the WebSphere Application Server for distributed platforms library Web page.

Overview of workload management and servers

Tip: If you are running z/OS V1.2 or above with the WLM-DAE support PTF (APAR OW54622), you can skip this section.

Note: To get started, you do not need to define special classification rules and work qualifiers, but you may want to do this for your production system. For more information, see "Implementing advanced performance controls" on page 201.

Because the Installation Verification Test needs servers, you must also define an application environment for the Java application server. We include that server in Appendix A, "Default server values for WebSphere Application Server for z/OS V5", on page 217.

Just like servers for your business applications, the WebSphere for z/OS run-time servers (with the exception of the location service daemon and node agent) have a controller and one or more servants. The regions are started by the start procedures shown in Appendix A, "Default server values for WebSphere Application Server for z/OS V5", on page 217.

You have to start the controllers for the WebSphere for z/OS run-time servers and business application servers yourself. This in turn starts the location service daemon. Workload manager dynamically starts the servants as work requests arrive. Thus, you must create WLM application environments that name servant start procedures to start, as shown in Appendix A, "Default server values for WebSphere Application Server for z/OS V5", on page 217. For example, specify BBO5ASR as the start procedure name that workload management starts for the server.

Each new server that you create for a business application also must be defined to workload management. For more information, see the assembling applications information in the WebSphere Application Server InfoCenter, which you can access via the WebSphere Application Server for distributed platforms library Web page.

Step for defining workload management policies for the run-time servers

Tip: If you are running z/OS V1.2 or above with the WLM-DAE support PTF (APAR OW54622), you can skip this section.

Before you begin: You must have access to the IWMARIN0 application and be able to update the workload management policies.

Perform the following step to define the workload management policies:

1. Use the ISPF application IWMARIN0 to define WLM application environments according to Appendix A, "Default server values for WebSphere Application Server for z/OS V5", on page 217.

   Tip: Appendix A, "Default server values for WebSphere Application Server for z/OS V5", on page 217 uses IBM default names for the servers and procedures as examples. The customization dialog allows you to change server and procedure names, which means you would need to use the values you supply through the customization dialog. The dialog also tailors the workload management definition task to include the names you supply, so you may prefer to follow the customized instructions from the dialog. For more on the customization dialog, see Chapter 3, "Installing and customizing your first run-time", on page 45.

   For details on defining the application environments to workload manager, see *z/OS MVS Planning: Workload Management*, SA22-7602.

You are done when you activate the service policy and exit IWMARIN0.

The following example shows how to create an application environment for BBOC001. You must perform the steps in the example for both your Application Server and Deployment Manager.

Example of using IWMARIN0: The following shows the panels you use in IWMARIN0 to define an application environment.

Before you begin: Workload management must be running in goal mode, and you must have access to a WLM definition, either saved in a WLM definition data set, or active in the WLM couple data set.

The user of IWMARIN0 must have update access to the RACF FACILITY class profile MVSADMIN.WLM.POLICY.

Perform the following steps to create the BBOC001 application environment:

1. Open the main panel by issuing IWMARIN0. Either load a WLM goal mode definition from a WLM definition data set, or extract a working goal mode definition from the WLM couple data set. Then choose option 9:

```
  File  Utilities  Notes  Options  Help
------------------------------------------------------------------------------
Functionality LEVEL003         Definition Menu         WLM Appl LEVEL004
Command ===> ____________________________________________________________

Definition data set  . . : 'CB.MYCB.WLM'

Definition name  . . . . . CB390      (Required)
Description  . . . . . . . WLM Setup for WebSphere for z/OS

Select one of the
following options. . . . . 9__  1.  Policies
                               2.  Workloads
                               3.  Resource Groups
                               4.  Service Classes
                               5.  Classification Groups
                               6.  Classification Rules
                               7.  Report Classes
                               8.  Service Coefficients/Options
                               9.  Application Environments
                              10.  Scheduling Environments
```

2. Fill in the field on the next panel as shown:

```
  Application-Environment  Notes  Options  Help
------------------------------------------------------------------------------
                    Create an Application Environment
Command ===> ____________________________________________________________

Application Environment  . . . BBOC001_____________________________  Required
Description  . . . . . . . . . WebSphere for z/OS IVT server___
Subsystem Type . . . . . . . . CB__                                 Required
Procedure Name . . . . . . . . BBO5ASR
Start Parameters . . . . . . . JOBNAME=&IWMSSNM.S,ENV=CELL1.&SYSNAME..
                               &IWMSSNM_____________________________
                               _____________________________________

Limit on starting server address spaces for a subsystem instance:
2   1.  No limit
    2.  Single address space per system
    3.  Single address space per sysplex



   .-------------------------------------------------------------------.
   | Selection List empty. Define an application environment. (IWMAM600) |
   '-------------------------------------------------------------------'
```

Note: In the previous panel, change `CELL1` to your cell_short_name and `&SYSNAME` to your node_short_name as needed. Otherwise, the defaults will be assumed.

3. Save the application environment. The following panel appears:

```
  Application-Environment  Notes  Options  Help
------------------------------------------------------------------------------
                  Application Environment Selection List     Row 1 to 12 of 12
Command ===> ____________________________________________________________

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar

Action  Application Environment Name      Description
  __    BBOC001                           WebSphere for z/OS IVT server
****************************** Bottom of data ********************************
```

4. From the Utilities menu, select Install definition.

5. From the Utilities menu, select Activate service policy.

6. From the File menu, select exit.

You are done when BBOC001 has an application environment.

Recommendations for resource recovery services

WebSphere for z/OS requires the use of the RRS Attach Facility (RRSAF) of DB2, which in turn requires that resource recovery services (RRS) be set up. If you do not have RRS set up, the customization dialog helps you do this. See Chapter 3, "Installing and customizing your first run-time", on page 45.

When setting up RRS, consider the following:

1. You may have already configured RRS for z/OS to exploit WLM-managed DB2 Stored Procedures address spaces. However, if DB2 is the only RRS-compliant resource manager participating in transactional commits, optimizations will cause the system to bypass RRS usage of the system logger. This means that, while your installation may have configured RRS, your log streams might have just minimal activity. WebSphere for z/OS is an RRS-compliant resource manager and will participate in transactional commits with DB2. Thus, WebSphere for z/OS will require RRS to start writing data to its system logger log streams. You might need to adjust the size of your log streams.
   - WebSphere for z/OS has no significant impact on the RM.DATA log.
   - Depending on the transaction policies of both the client and container, you may not see any activity in the MAIN.UR log. This lack of activity is not a problem.
   - Depending on the transactional policy defined for your containers, you may see much more activity in your DELAYED.UR log stream than in the MAIN.UR log stream.

     All RRS transaction logging for WebSphere for z/OS will occur solely in the DELAYED.UR log stream. Such logging may change in future releases of WebSphere for z/OS, so you may still want to configure your MAIN.UR log stream so that it can handle a production workload, in case you deploy a new container or the WebSphere for z/OS infrastructure changes.
   - WebSphere for z/OS has no significant impact on the RESTART log.

- There is no reason to change your policy about the ARCHIVE log. Though optional, we suggest you use the ARCHIVE log. It has a small negative effect on performance. Set the retention period for the log as you would normally.

2. The Object Transaction Service in WebSphere for z/OS cannot detect when it has been restarted in a different logging group, which affects transaction recovery. We recommend you use automatic restart management (ARM) to control restart locations.
3. For structure sizes, we recommend the following for initial setup values. Through experience, you may need to adjust these:

Table 11. Recommended size of log streams

Log stream	Initial size	Size
RM.DATA	1 MB	1 MB
MAIN.UR	5 MB	50 MB
DELAYED.UR	5 MB	50 MB
RESTART	1 MB	5 MB
ARCHIVE	5 MB	50 MB

Check the MAXBUFSIZE on your log streams. If the size is too small, you may encounter DB2 failures.

Details about resource recovery are in *z/OS MVS Programming: Resource Recovery*, SA22-7616. Details about the RRS Attach Facility are in *DB2 for OS/390 Application Programming and SQL Guide*, SC26-8958.

Guideline for RMF and other monitoring systems

You can use any performance and monitoring system you choose.

Guidelines for Java Database Connectivity

Java Database Connectivity (JDBC) provides an interface for Java application programs to access relational data in a database by using dynamic SQL. DB2 supports this application programming interface. For complete information about JDBC and DB2, see *DB2 for OS/390 Application Programming Guide and Reference for Java*. This topic covers guidelines related to WebSphere for z/OS's use of JDBC.

- You may use JDBC (dynamic SQL) in your server applications.
- Record the location of the run-time properties file, db2sqljjdbc.properties. You will use the location during the WebSphere for z/OS customization process. If you customize this file, you may want to keep the customized version in a separate directory such as `/etc` and record its location.
- All Java servers and the node must be granted EXECUTE authority on the DSNJDBC plan. If your installation allows public access to the DSNJDBC plan, all you need to do is issue:

  ```
  GRANT EXECUTE ON PLAN DSNJDBC TO PUBLIC
  ```

  If your installation does not allow public access to the DSNJDBC plan, then you must grant EXECUTE authority to all Java servers and the node. If you use DB2 secondary authorization IDs, then you can grant the authority to the groups to which the server IDs belong.
- You must use the RRSAF attachment interface (not CAF).

For more information about setting up JDBC and the implications for application programs, see *DB2 for OS/390 Application Programming Guide and Reference for Java*.

Guidelines for DB2 settings for WebSphere concurrency control management

If your installation uses typical DB2 defaults for U-lock management and lock size, certain WebSphere applications that use container-managed Enterprise beans (CMP beans) may encounter deadlocks. The likelihood of encountering deadlocks is entirely dependent on the design and execution pattern of the application. The potential for deadlocks increases with the number and frequency of applications driving concurrent transactions that update the same areas of the DB2 database. If, given your installation's application workload, the potential for deadlocks is high, consider using the following DB2 settings:

- RRULOCK(YES)
- LOCKSIZE(ROW)

For additional details, see the information about settings for the internal resource lock manager (IRLM) in *DB2 Installation Guide*, GC26-9936.

Alternative: Your applications may qualify for the optimistic approach to concurrency control management. To determine whether your applications can use optimistic concurrency control, see the topic about controlling concurrent access to persistent data in the assembling applications information in the WebSphere Application Server InfoCenter, which you can access via the WebSphere Application Server for distributed platforms library Web page.

Recommendations for using memory

WebSphere for z/OS differs from previous application servers in its use of memory. WebSphere for z/OS's implementation takes advantage of z/OS's efficient memory management, but, like many of today's newer application servers and languages, it is a large consumer of memory. You may experience some changes from your existing memory usage patterns. This section outlines changes you might need to make. Follow these recommendations:

- For real storage requirements, see "z/OS hardware requirements" on page 12.
- In your production cell, we recommend you dynamically load the run-time in the link pack area (LPA) because the size of the load modules is large, and many address spaces need to refer to those load modules. The load modules for the run-time comprise about 40 MB in size.

  Note: WebSphere Application Server for z/OS V5 allows for multiple cells to run on the same system at the same time. With such a configuration, you need to identify the SBBOLPA data set as part of STEPLIB within all the WebSphere for z/OS PROCs. WebSphere for z/OS will then automatically load these load modules into LPA without you placing them there yourself. One exception is BBORTSS5; you must place this load module into LPA or link list yourself because that is the only way it can get control in master.

  Because you are using dynamic LPA, you may run out of ECSA after an IPL if you do not increase CSA at IPL time. Add 40 MB to ECSA in support of

WebSphere for z/OS. You should monitor ECSA after dynamically loading the run-time into LPA. Remember to increase the size of your CSA page data set accordingly.

Note: Do not dynamically load the run-time in the LPA in your development cell. Load the DLLs out of private memory instead of using ECSA.

- If you choose to place the load modules in steplib or in the link list, you must allow for the additional 40 MB as part of each address space's region. A typical WebSphere for z/OS base configuration runs 3 address spaces, and a typical WebSphere for z/OS network deployment configuration runs 8 address spaces. Each of these reference most of the 40 MB of load modules.
- In addition to placing the load modules in the link pack area, give each address space a dynamic area of at least 128 MB.
- Check to see whether your installation limits region sizes through the IEFUSI exit, JES exits, or TSO segment defaults. All of the WebSphere for z/OS JCL procedures are shipped with a default REGION=0M, which means you should give them as large a region as possible. If you choose to run from the link pack area, you will need a minimum of 128 MB for the dynamic area. If you choose to run from the link list you will need a minimum of 168 MB (40 MB for load modules and 128 MB for the dynamic area).

  If your IEFUSI exit routine limits the maximum region to a size smaller than what you need (128 MB minimum when you run from the link pack area or 328 MB minimum when you run from the link list), you will get an abend. To fix the problem, either change the IEFUSI exit routine to allow a larger default region, or change the JCL REGION= parameter to the size needed.

  Your installation may limit (control) the specification of REGION=, usually through the JES2 EXIT06 exit or the JES3 IATUX03 exit. If so, relax this restriction for the WebSphere for z/OS JCL procedures.

  Finally, check your TSO segment default region size and, if necessary, change it.

Additional information about tuning your application's memory usage is in "Implementing advanced performance controls" on page 201.

Planning for problem diagnosis

This section describes:

- WebSphere for z/OS's use of Component Trace
- The WebSphere for z/OS error log stream
- Dump data sets

Overview of problem diagnosis

WebSphere for z/OS uses component trace (CTRACE) to capture and display trace data in trace data sets. WebSphere for z/OS identifies itself to CTRACE with a dynamic component name determined by the short cell name. CTRACE allows you to:

- Merge multiple traces through the browse tool, including other components such as TCP/IP and z/OS UNIX.
- Write trace data to a data set rather than sysprint, keeping spool space free.
- Better manage system resources by allowing trace data to wrap or not wrap.
- Use CTRACE to funnel trace data from multiple address spaces to one data set, or have CTRACE send the trace data from each address space to separate data sets.

- Start and stop tracing without stopping and restarting WebSphere for z/OS address spaces.
- Use one or more data sets for capturing trace data, thus allowing you to manage I/O more effectively.

WebSphere for z/OS also has an error log stream that records error information when WebSphere for z/OS detects an unexpected condition or failure within its own code, such as:

- Assertion failures
- Unrecoverable error conditions
- Vital resource failures, such as memory
- Operating system exceptions
- Programming defects in WebSphere for z/OS code

Use the error log stream in conjunction with other facilities available to capture error or status information, such as an activity log, trace data, system logrec, and job log.

The WebSphere for z/OS error log stream is a system logger application. Because the error log stream uses the system logger, you can:

- Have error information written to a coupling facility log stream, which provides sysplex-wide error logging, or to a DASD-only log stream, which provides single system-only error logging.

  Note: There is a significant performance penalty when using DASD-only error logging.
- Set up either a common log stream for all of WebSphere for z/OS or individual log streams servers. Local z/OS or OS/390 client ORBs can also log data in log streams. Because the system logger APIs are unauthorized, any application can use them. You should control access to the log streams through a security product such as RACF.

WebSphere for z/OS provides a REXX EXEC (BBORBLOG) that allows you to browse the error log stream. By default, the EXEC formats the error records to fit a 3270 display.

This manual describes the error log stream and how to set it up. Information about using the error log stream to diagnose problems is in *WebSphere Application Server for z/OS V5.0: Diagnosis*, GA22-7915. General information and guidance about the system logger is in *z/OS MVS Setting Up a Sysplex*, SA22-7625. Table 12 shows where to find information pertinent to the error log stream:

Table 12. Finding WebSphere for z/OS Error Log Stream Information

What is your goal?	You should read:
Learn about the system logger and understand its requirements	*z/OS MVS Setting Up a Sysplex*, SA22-7625
Learn about the WebSphere for z/OS error log stream	"Overview of problem diagnosis" on page 36
Plan for and set up the WebSphere for z/OS error log stream	*z/OS MVS Setting Up a Sysplex*, SA22-7625 and Table 17 on page 62
Size the coupling facility structure space needed for the WebSphere for z/OS error log stream	*z/OS MVS Setting Up a Sysplex*, SA22-7625

Table 12. Finding WebSphere for z/OS Error Log Stream Information (continued)

What is your goal?	You should read:
Define access authorization to system logger resources for the WebSphere for z/OS error log stream	Table 17 on page 62
Define the WebSphere for z/OS error log stream	Table 17 on page 62
View the WebSphere for z/OS error log stream	*WebSphere Application Server for z/OS V5.0: Diagnosis*, GA22-7915
Learn about how Java applications can log messages and trace data in the error log stream	The assembling applications information in the WebSphere Application Server InfoCenter, which you can access via the WebSphere Application Server for distributed platforms library Web page

For details about problem diagnosis, see *WebSphere Application Server for z/OS V5.0: Diagnosis*, GA22-7915.

Post-installation notes on the error log

After installation is complete, use the Administrative Console to change the log stream name or create new log stream names for servers or servants.

Notes:

1. A servers error log stream setting overrides the general WebSphere for z/OS setting, and a servant setting overrides a server setting. Thus, you can set up general error logging, but direct error logging for servers or servants to specific log streams.
2. If you create a new log stream name through the Administrative Console, you must configure a new log stream on z/OS and, if using the coupling facility, define a corresponding new coupling facility log stream.
3. If you changed an existing log stream, or created a new one, you probably need to restart WebSphere for z/OS. When the name of a log stream is changed through the Administrative Console, in most cases a restart of WebSphere for z/OS is required before the change becomes effective. The only case when the change takes effect automatically is when the log stream name is changed for a server along with other changes that cause the server to be restarted.

If you want WebSphere for z/OS messages that occur during execution of a z/OS client to be recorded in an error log stream, code the client_ras_logstreamname WebSphere variable in its environment file, then initialize the client. For more information about client_ras_logstreamname, see the WebSphere variables in the Administrative Console or the InfoCenter.

Our RACF samples BBOWBRAC and BBODBRAC give UPDATE authority to the run-time control and servant user IDs for the log stream you created (they require that you supply a log stream name). After installation and customization, if you want to grant access to the log stream:

- For each server identity that writes to the log stream (or client identity, if you allow clients to write to the error log stream), assign UPDATE access to the log stream.
- For each user who browses the error log stream, assign READ access.

Follow the sample RACF commands in BBOWBRAC or BBODBRAC.

Ensuring problem avoidance

To implement WebSphere for z/OS, you must implement the necessary features, subsystems, and resources required for the run-time environment. This section provides checklists for tasks you should verify before running your WebSphere for z/OS system in order to prevent the most common errors encountered during the installation.

Steps for ensuring problem avoidance

Before you begin: Perform the following steps to ensure problem avoidance, checking off each item as you complete it:

1. Plan to prepare your z/OS environment:

Check off	Item
	Check that all the maintenance suggested in the PSP bucket WASAS500 subset WAS500 has been applied.
	Make certain your address space is large enough. Some WebSphere for z/OS servers must be able to get a 1GB virtual region to run any workload. Make sure that your installation exits (IEFUSI) do not limit the virtual region size. We recommend that you specify REGION=0M so as not to limit their size.
	Add another local page data set, two if your system does any paging of the WebSphere for z/OS server address spaces.

2. Plan to prepare your DB2 subsystem (if you will use DB2):

Check off	Item
	Increase the MAX USERS (CTHREAD) and MAX BATCH CONNECT (IDBACK) in your DB2 environment settings. Use the sample job in DSN710.SDSNSAMP(DSNTEJ6Z) to display the ″ZPARMS″ settings of the running system. (An alternative is to use the DB2 Control Center to display these parameters.)
	Define at least 200 buffers to the DB2 BP32K buffer pool. Use this command to display the current bufferpool allocations: `-dis bpool(active)detail`. Verify JDBC 2.0 functionality. The JDBC IVT sample01 JAVA application does not exercise JDBC 2.0 drivers nor the RRS attach facility. A modified version that tests these functions can be found in the DB2 Conundrum whitepaper on Techdocs at http://www.ibm.com/support/techdocs/atsmastr.nsf/PubAllNum/WP100217 (This will also verify that the DSNJDBC plan is bound correctly and that it matches the .ser file.)
	Verify the level of DB2 code running on your system with the DSNTEJ6U sample job or run the DSNUTILB utility with the `DIAGNOSE DISPLAY MEPL` command. The module names, dates, and PTF number on the right of the report are in EBCDIC.
	Make sure that any updates to the DB2 ERLY code are installed, and that you have IPLed your system to activate them.

Check off	Item

	Check the JDBC service installed on your system. Use the following java program to display the service level:
	`>export LIBPATH=/usr/lpp/db2/db2710/lib:$LIBPATH`
	`>java -cp /usr/lpp/db2/db2710/classes/db2j2classes.zip COM.ibm.db2os390.sqlj.util.DB2DriverInfo`
	
	The typical output message looks like this:
	`DB2 for OS/390 SQLJ/JDBC Driver build version is:DB2 7.1 PQ54756`

3. Plan to verify your USS/HFS configuration:

Check off	Item
	Specify enough threads, files, and processes in your BPXPRMxx member of parmlib. Here is a starting list if you don't have it set up yet: • MAXTHREADS:10000 • MAXTHREADTASKS:5000 • MAXFILEPROC:10000 • MAXSOCKETS in the AF_INET domain:12000
	If you have an exit that checks for valid accounting codes, you may need to specify an accounting value for spawned address spaces. Use the `_BPX_ACCT_DATA=` variable in the current.env file.

4. Plan your SMP/E tasks:

Check off	Item
	You can install WebSphere for z/OS into an SMP/E environment (SMP/E 3.1 or later) separate from the one you use for z/OS. This includes target and distribution zones, as well as HFS data sets. We recommend you use a separate environment, but you should enable the cross-zone checking so that any prerequisite service requirement can be checked between the WebSphere for z/OS and z/OS SMP zones.
	Verify that the DDDEF for the LTS data set describes a PDSE format data set. This will avoid LINK-EDIT errors during the SMP/E processing.
	We strongly recommend that you carefully read *WebSphere Application Server for z/OS V5.0: Program Directory*, GI11-2825. This is a very large product and you have to make sure that there is sufficient space in all target and temporary data sets for receive and apply processing.
	If you are applying maintenance using a staging HFS file, remember to copy the updated HFS file to the mounted HFS file or mount the updated HFS file on the correct mount point.
	After maintenance has been applied, verify that the new code has been loaded into linklist and LPA.

5. Plan your ISPF dialog consideration:

Check off	Item
	Do not use ISPF dialog in Split Screen mode. It may happen that you don't see everything on one screen. Keep in mind that there might be parameter values outside of the visible portion of your screen and that this may cause some of the WebSphere for z/OS installation options to be set to default values, which may not be what you want.
	Turn off PFSHOW on the ISPF dialog screen. As with the split screen problem, there might be parameter values outside the visible portion of your screen, which may cause some of the WebSphere for z/OS installation options to be set to default values, which may not be what you want.
	You need to use a screen size of at least 32 lines to be able to invoke ISPF dialog.
	You might want to consider starting with new CNTL and DATA data sets as output of the ″Generate the Jobs Stream″ function. During this task, you get a large number of confirmation messages for every successful generated job. It is very easy to overlook an error message during this process. Unfortunately this sometimes causes the job generation process to stop without recreating all JCL. When you run the jobs you may run ″old″ JCL and this will result in unpredictable problems.

6. Plan to verify your WLM environment:

Check off	Item
	Verify that WLM is running in goal mode with the `D WLM,SYSTEMS` command **Tip:** If you are running z/OS V1.2 or above with the WLM-DAE support PTF (APAR OW54622), you can skip this item.

	Verify that the two base and one IVT WLM application environments are defined and available with the `D WLM,APPLENV=*` command.
	
	Example: This example shows you how to check the WLM application environment.
	`d wlm,applenv=BBOC001`
	`RESPONSE=SC42`
	`IWM029I 13.09.47 WLM DISPLAY 075`
	`APPLICATION ENVIRONMENT NAME STATE STATE DATA`
	`BBOC001 AVAILABLE`
	`ATTRIBUTES:PROC=BBO5ASR SUBSYSTEM TYPE:CB`

7. Plan to check your TCP/IP configuration:

Check off	Item
	Make sure that Java applications can successfully access getlocalhost, gethostbyaddress, and gethostbyname. This helps ensure that most TCP/IP functions are set up correctly for WebSphere for z/OS.

Check off	Item

	Telnet into UNIX Systems Services and issue these commands to verify that you can find your host name by IP address or IP host-name:
	• – Get the local host name: `hostname`
	– You will get a response such as: wtsc49.itso.ibm.com
	– Use the output from the hostname command for the following `nslookup` command.
	• Get host address by name: `nslookup sc49.itso.ibm.com`
	– You will get a response such as this:
	`Server:sc49.itso.ibm.com`
	`Address:9.12.6.15`
	`Name:sc49.itso.ibm.com`
	`Addresses:9.12.6.15`
	– Use the dotted IP address from this display for the following command.
	• Get host name by address: `nslookup 9.12.6.15`
	– You will get a response such as in the previous nslookup display.
	There is also a small java program, InetInfo.java, which you can run to verify the same TCP/IP configuration. See techdocs for the program at http://www.ibm.com/support/techdocs/atsmastr.nsf/PubAllNum/TD100609.
	Example: This example shows you how to run the InetInfo java code.
	`JAVA4 @SC42:/u/java4>export PATH=/usr/lpp/java/IBM/J1.3/bin`
	`JAVA4 @SC42:/u/java4>java InetInfo`
	`get Local Host`
	`IP Address:9.12.6.27`
	`get Host Name By Address using 9.12.6.27`
	`Host Name:wtsc42oe.itso.ibm.com`
	`get Host Address By Name using wtsc42oe.itso.ibm.com`
	`Host Address:9.12.6.27`

	Issue the `hometest` command from TSO. It should show the correct TCP Host name, corresponding IP address(es), and HOME IP addresses. If it doesn't produce the correct results, then TCP/IP is not configured correctly.
	If the fully-qualified TCP/IP HostName is greater than 24 characters, then a DNS will be required. Otherwise, the /etc/hosts file can provide the naming lookup.
	The following TCP/IP servers should also be up to provide access: You must have an FTP server that can access the HFS for deploying applications.
	Check if you protect the standard ports below 1024 (TCPCONFIG - RESTRICTLOWPORTS). If this is the case, then you must add one line for each node to the PROFILE.TCPIP configuration file. This entry will allow the userid of the node to use the reserved port: `900 TCP SZSMGT02`
	Verify that the dns name you are using is definitive in your installation.

8. Plan to verify that security is in place:

Check off	Item
	Check that the location service daemon has access to parmlib concatenation to retrieve CTRACE settings in the CTIBBOxx member.
	Verify that all WebSphere for z/OS servers must have READ access to any datasets or files in their JCL procedures.

Check off	Item
	Verify that your installation has the RACF list-of-groups turned on. (SETROPS LIST will show you if turned on or off.) Without this list of groups turned on, an ID cannot belong to more than one group and ASSR1 associates with only WSSR1 instead of both WSSR1 and WSCFG1. Use the command `SETROPTS GRPLIST` to turn on the list of groups.
	Define the profile BPX.SAFFASTPATH in the FACILITY class to enable SAF fastpath support.
	If you do not load SBBOLOAD into LPA, you must add it to the program control list in the RACF PROGRAM class. (With z/OS V1.2, you can use the FACILITY class profile BPX.DAEMON.HFSCTL class. This will cause only HFS files to be checked for program control.)
	Verify that the /usr/lpp/java/IBM/J1.3/lib HFS file permission bits are set up correctly to allow the read capability to other (644) and that /usr/lpp/java/IBM/J1.3/bin and /usr/lpp/java/IBM/J1.3/bin/classic with the execute permission bit on (755) and APF authorized.
	Verify that the authorization bits for the WebSphere for z/OS HFS (default name is /usr/lpp/usr/lpp/zWebSphere/V5R0M0) file are correctly set up for the WebSphere for z/OS System Management userid/group

9. Plan your installation phases:

Check off	Item
	After any maintenance has been applied, verify that the code loaded in LPALIB or LNKLST is in sync with the code in the HFS. Check the location service daemon joblog to verify that the correct maintenance level is in use.

You are done when you have checked all the applicable items.

Planning for Component Trace

To use CTRACE, you:

- Specify trace options for identifying trace data sets and connecting WebSphere for z/OS address spaces to the data sets in parmlib members.
- Update WebSphere for z/OS WebSphere variables to allow for initial trace parameters.
- Use IPCS-CTRACE to view the trace data because you cannot read the trace data in an ordinary editor.

For more information about setting up CTRACE for WebSphere for z/OS, see *WebSphere Application Server for z/OS V5.0: Diagnosis*, GA22-7915.

Recommendation for dumps

Plan as you would normally for system dumps. Due to the size of WebSphere for z/OS address spaces, you may need to re-size your system dump data sets.

Tip on automatic restart management (ARM)

If you have automatic restart management (ARM) enabled on your system, you may wish to disable ARM for the WebSphere for z/OS address spaces before you install and customize WebSphere for z/OS. During customization, job errors may cause unnecessary restarts of the WebSphere for z/OS address spaces. After installation and customization, consider enabling ARM. For more information, see "Setting up automation and automatic restart management" on page 125.

Chapter 3. Installing and customizing your first run-time

You should follow this chapter in the order in which it is presented.

1. "Preparing for installation and customization" on page 46 tells you about steps you must complete before you start customizing WebSphere for z/OS and configuring the run-time servers.
2. "Installing the code through SMP/E" on page 48 tells you where to find information about installing the product code.
3. "Using the customization dialog" on page 49 explains how to run the customization dialog and follow the generated instructions.

If you encounter problems during installation and customization, refer to *WebSphere Application Server for z/OS V5.0: Diagnosis*, GA22-7915, for trouble-shooting information.

Overview of installing and customizing WebSphere for z/OS

This topic explains the installation and customization process at a high level.

Installing and customizing WebSphere for z/OS requires that you prepare the operating system and subsystems, install the product code through SMP/E, run the customization dialog, configure native products (for example, WLM, RACF and TCP/IP), follow the customized instructions and run the jobs, including the Installation Verification Test, from the dialog, and bring up your server.

You can find background information about preparing z/OS subsystems in Chapter 2, "Preparing the base z/OS environment", on page 11.

For information about installing the product code through SMP/E, see *WebSphere Application Server for z/OS V5.0: Program Directory*, GI11-2825.

The customization dialog is an ISPF dialog that eliminates the need to hand-tailor sample jobs supplied with the product. You define the customization options once in the dialog panels, then the dialog generates the jobs with your options, eliminating the need to define them in several places. The benefit to you is reduced typos and inconsistencies, and a quicker customization.

IBM provides an Installation Verification Test that tests Web applications and server components, such as enterprise beans. At the end of installation and customization, you will run this program.

The following table outlines the installation and customization process:

Stage	Description
1	Install prerequisite products. Configure z/OS subsystems, such as resource recovery services (RRS) and workload management.
2	Install WebSphere for z/OS using SMP/E according to the *WebSphere Application Server for z/OS V5.0: Program Directory* (if you use CBPDO) or *ServerPac: Installing Your Order* (if you use ServerPac).

Stage	Description
3	Run the customization dialog. Through a series of panels, you choose options and define variables. Using your values, the dialog tailors the WebSphere for z/OS customization jobs but does not execute them. Rather, the dialog provides a custom set of instructions for you to follow. When you finish the dialog, you have a set of instructions and tailored jobs ready to complete the product customization.
4	Follow the instructions created by the customization dialog. When you finish, you have a complete WebSphere for z/OS run-time configuration.
5	With the Administrative Console, create the server definition. This server is used by the Installation Verification Test and is an example of an application server you will create for your own applications.
6	Run the Installation Verification Test to verify that your WebSphere for z/OS system is working properly.

When you finish the entire installation and customization process, you have WebSphere for z/OS running in a system. As you gain experience, you can roll out WebSphere for z/OS across your sysplex to gain the advantages of z/OS sysplex operations.

Preparing for installation and customization

You must prepare z/OS subsystems and do other tasks in this section before you start installation and customization. Additionally, you must determine important information about WebSphere for z/OS and z/OS subsystems before you start customization.

Steps for preparing your z/OS subsystems

Before you begin: Read Chapter 1, “Overview of installation and customization”, on page 1.

Follow these steps:

1. Prepare your z/OS subsystems (see Chapter 2, “Preparing the base z/OS environment”, on page 11). In particular, be sure you have followed instructions and tips for the following:
   - System requirements. See “Determining WebSphere for z/OS system requirements” on page 12.
   - TCP/IP. See background information and tips in “Updating your TCP/IP network” on page 16.
   - Security Server (RACF). See “Setting up security” on page 18.
   - Workload manager (WLM). See “Setting up workload management (WLM)” on page 29.
   - Resource Recovery Services. See “Recommendations for resource recovery services” on page 33.

   ---

2. If you do not already have one, set up a RACF user ID and authorize it to have read/write access to the WebSphere for z/OS files (BBO.* data sets and HFS files).

Note: In this book we cite product data set names without high-level qualifiers, unless a full data set name is required for clarity, in which case we use `BBO` as the qualifier.

You are done when you have successfully finished these preparations.

Installing the code through SMP/E

To install the code through SMP/E, follow one of two documents depending on what you use:

- If you use CBPDO, follow the *WebSphere Application Server for z/OS V5.0: Program Directory*, GI11-2825.
- If you use ServerPac, follow *ServerPac: Installing Your Order*.

You can find further information on the eSupport Web site at http://www.ibm.com/software/webservers/appserv/zos_os390/support.html, or check the PSP buckets or contact the IBM Software Support Center.

Notes:

1. You can change the high-level qualifier of the installed data sets (not recommended) or the middle-level qualifier.
2. If you are installing from a driving system, make sure the maintenance level of the target system meets requirements for WebSphere for z/OS.
3. Make sure the product code HFSes are mounted at `/usr/lpp/java` and `/usr/lpp/usr/lpp/zWebSphere/V5R0M0`, or at similar mount points of your choice.

Using the customization dialog

The customization dialog is intended for the system programmer or administrator responsible for installing and customizing WebSphere for z/OS. It lets you separately configure a base Application Server, integral JMS provider, Deployment Manager, and Web Services. In order to use the dialog, you must know or be able to find the system characteristics for the system on which WebSphere for z/OS will run.

The dialog covers a portion of WebSphere for z/OS customization. Specifically, it creates tailored jobs to:

- Copy the generated jobs into your system libraries.
- Create the run-time HFS structure and the initial environment file
- Set up WebSphere for z/OS security controls (RACF)
- Define the WebSphere for z/OS run-time configuration (Application Server, integral JMS provider, Deployment Manager, Web Services, location service daemon)
- Run the Installation Verification Test (IVT)

  Note: For information on running the Installation Verification Test at times other than during initial customization, see "Running the Installation Verification Test (IVT) after initial customization" on page 134.

The dialog also produces a set of instructions for you to follow in order to effectively run these tailored jobs.

Note: Be aware of other versions of WebSphere for z/OS that you have running on your system, as the customization dialog does not detect them for you. Keep this in mind when going through the dialog and, if you are running other versions of WebSphere for z/OS, watch out for such problems as potential location service daemon port collision or LPA issues.

Steps for starting the customization dialog

Before you begin: You must have the product code installed and have access to the product data sets.

Rules: Regarding your display:

- Your logon display must support 3270 emulation and be set to a minimum of 32 rows by 80 columns (32 x 80) in order for the ISPF customization dialog to run.
- If you have a 32-row display and use the ISPF split screen function, deselect "Always show split line" on the ISPF Settings panel and split the screen at the extreme top or bottom of the display. This prevents the split screen line from displaying and lines in the customization dialog from being obscured. Other uses of split screen will obscure lines in the customization dialog.
- If you have a 32-row display, you cannot display the PF key settings. Displaying the PF key settings will obscure lines at the bottom of the dialog panels. Issue `PFSHOW OFF`.

You should complete the worksheets in this section.

Perform the following steps to run the customization dialog:

1. From the ISPF command line, enter the following:

   ```
   ex 'hlq.sbboclib(bbowstrt)' 'options'
   ```

where

hlq
Is the high-level qualifier for the SBBOCLIB data set.

options
Are command options. Enclose any and all options in a single set of quotes.

appl(*value*)
Specifies the ISPF application name. This option creates unique ISPF profiles, usually stored in the ″userid.ISPF.ISPFPROF″ dataset, that are useful when you want to keep saved variables separate from those in other target environments. The default value is `BBO5`.

lang(*value*)
Specifies the national language. Values can be either `ENUS` (English) or `JAPN` (Japanese). The default is `ENUS`.

Example:

```
ex 'bbo.sbboclib(bbowstrt)' 'appl(bbo5) lang(enus)'
```

Result: You see the splash screen:

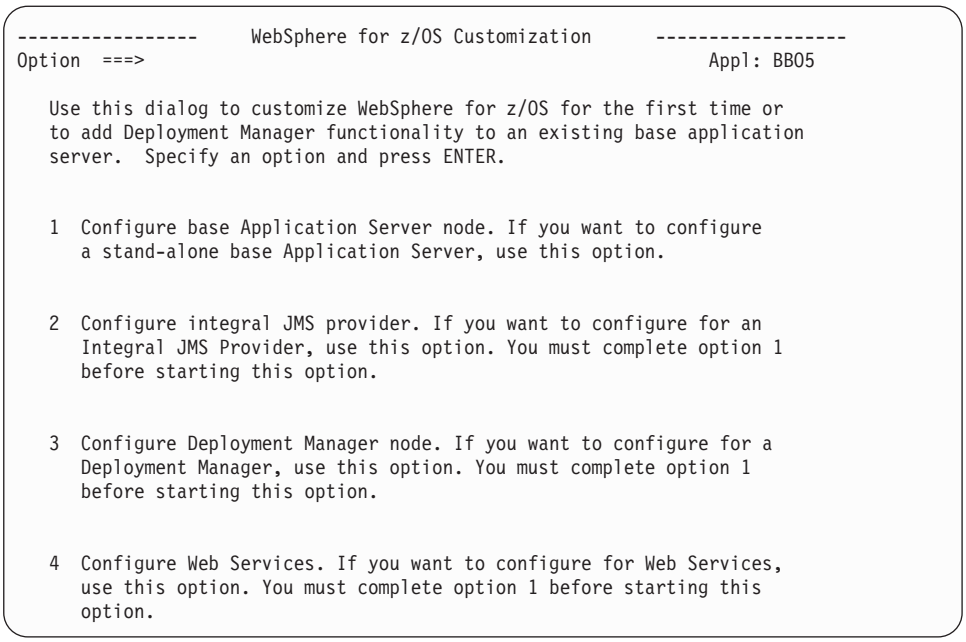
```
-----------------     WebSphere for z/OS Customization      ------------------
Option  ===>


     WebSphere Application Server for z/OS V5

     Licensed Material - Property of IBM
     5655-I35 (C) Copyright IBM Corp. 2000, 2003

     All Rights Reserved.
     U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
     Disclosure restricted by GSA-ADP schedule contract with IBM Corp.

     Status  = H28W500

     Version = 5.00.000


                       Press ENTER to continue.
```

---

2. Press Enter.

   Result: You see the ″THIRD PARTY LICENSE TERMS AND CONDITIONS, NOTICES AND INFORMATION″ panel, which is too large to reprint here.
3. Press PF3.

   Result: You see the following panel:



You have finished starting the customization dialog and can now choose the option for the path you would like to follow.

Steps for allocating the target data sets

This section tells you how to complete the "Allocate target data sets" option that is in all of the four main tasks you can perform.

Before you begin: You must start the customization dialog and select your desired task option.

Perform the following steps to allocate the target data sets for each option:

1. On the main dialog panel, type your high-level qualifier in the `HLQ for WebSphere product data sets` field if you did not specify one when you started the customization dialog.

   _____

2. Type 1 in the `Option` field to select "Allocate target data sets".

   _____

3. Press Enter.

   Result: You see:

```
-----------------     WebSphere for z/OS Customization      ------------------
Option  ===>

 Allocate Target Data Sets

 Specify a high level qualifier (HLQ) and press ENTER to allocate the
 data sets to contain the generated WebSphere jobs and instructions.
 You can specify multiple qualifiers, up to 39 characters.

 High Level Qualifier:                                          .CNTL
                                                                .DATA

 The dialog will display data set allocation panels. You can make
 changes to the default allocations, however you should not change
 the DCB characteristics of the data sets.

    .CNTL  - a PDS with fixed block 80-byte records to
             contain WebSphere customization jobs.

    .DATA  - a PDS with variable length data to contain
             other data produced by the customization dialog.
```

4. On the Allocate Target Data Sets panel, type in the information from "1 Allocate Target Data Sets for the base Application Server node" on page 57, "1 Allocate Target Data Sets for the Integral JMS Provider" on page 80, "1 Allocate Target Data Sets for the Deployment Manager node" on page 94, or "1 Allocate Target Data Sets for Web Services" on page 114, then press Enter.

You are done when the data set allocation succeeds.

Steps for defining variables

This section tells you how to complete the "Define variables" option that is in all of the four main tasks you can perform.

Before you begin: You must start the customization dialog and select your desired task option.

Perform the following steps to define variables for each option:

1. On the main dialog panel, type 2 in the `Option` field to select "Define variables".
2. Press Enter.

   Result: You see:

```
-----------------     WebSphere for z/OS Customization      ------------------
Option  ===>

Define Variables to configure base Application Server node

   Specify a number and press ENTER to define the WebSphere variables.
   You should review all of the variables in each of the sections, even
   if you are using all of the IBM-supplied defaults.
   Once you complete all sections, press PF3 to return to the main menu.


                                                    Completed?
  1 - System Locations (directories, HLQs, etc)
  2 - System Environment Customization
  3 - Server Customization
  4 - Security Customization
```

Note: This example is from ″Option 1: Configure base Application Server node,″ but the steps are the same no matter which option you choose.

---

3. Follow the options in order and enter information from “2 Define Variables to configure base Application Server node” on page 58, “2 Define Variables to configure Integral JMS Provider” on page 81, “2 Define Variables to configure Deployment Manager node” on page 96, or “2 Define variables for Web Services” on page 116.

You are done when you finish all the ″Define Variables to configure...″ panels.

Steps for generating customization jobs

This section tells you how to complete the ″Generate customization jobs″ option that is in all of the four main tasks you can perform.

Before you begin: You must complete Option 2, Define variables.

Recommendation: When you have finished entering all your customization data, before you generate the customization jobs, use the S option to save your customization variables for future reference. See “Steps for saving the customization variables” on page 55.

Perform the following steps to generate the customization jobs:

1. On the main dialog panel, type 3 in the `Option` field to select ″Generate customization jobs″.

---

2. Press Enter.

   Result: If all variables are defined correctly, you see the Specify Job Cards panel:

```
-----------------     WebSphere for z/OS Customization      ------------------
Option  ===>

Generate Customization Jobs

 This portion of the Customization Dialog generates the jobs you must
 run after you complete this dialog process. You must complete the
 customization process before you generate the jobs with this step.
 If you have not done this, please return to that step.

 Jobs and data files will get generated into data sets:
   'hlq.CNTL'
   'hlq.DATA'
 If you wish to generate using other data sets, then END from this
 panel and select option 1 (Allocate target data sets).

 All the jobs that will be tailored for you will need a jobcard.
 Please enter a valid jobcard for your installation below. The
 file tailoring process will update the jobname for you in all the
 generated jobs, so you need not be concerned with that portion of
 the job cards below. If continuations are needed, replace the
 comment cards with continuations as needed.

 Specify the job cards.  Press ENTER to continue.

 //jobname  JOB (ACCTNO,ROOM),'jobname',CLASS=A,REGION=0M
 //*
 //*
 //*
```

3. Fill in the job card information according to "3 Generate customization jobs for the base Application Server node" on page 76, "3 Generate customization jobs for the Integral JMS Provider" on page 90, "3 Generate customization jobs for the Deployment Manager node" on page 110, or "3 Generate customization jobs for Web Services" on page 117, then press Enter.

You are done when all the jobs are generated.

Steps for viewing the generated customization instructions

This section tells you how to complete the "View instructions" option that is in all of the four main tasks you can perform.

Before you begin: You must complete Option 3, Generate customization jobs.

Perform the following steps to view the generated customization instructions:

1. On the main dialog panel, type 4 in the `Option` field to select "View instructions".
2. Press Enter. You will then see the generated instructions file.
3. View the instructions. You may print the instructions according to your local print procedures.

You are done when you view or print the instructions. You can now go on to follow them, using the worksheets for your task as guides.

Steps for saving the customization variables

This section tells you how to complete the ″Save customization variables″ option that is in all of the four main tasks you can perform.

Before you begin: You must complete Option 2, Define variables.

Perform the following steps to save the customization variables:

1. On the main dialog panel, type S in the `Option` field to select ″Save customization variables″.

---

2. Press Enter.

   Result: You see the Save Customization Variables panel.:

```
-----------------     WebSphere for z/OS Customization      ------------------
Option  ===>

Save Customization Variables

  Specify the name of a sequential data set to contain the customization
  variables. If the data set does not exist, the dialog displays the
  Allocate New Data Set panel, through which you can allocate a data set.
  Press Enter to continue.


 Data set name:
```

---

3. Type in the information from “S Save customization variables for the base Application Server node” on page 77, “S Save customization variables for the Integral JMS Provider” on page 91, “S Save customization variables for the Deployment Manager node” on page 111, or “S Save customization variables for Web Services” on page 118, then press Enter.

   Attention: Be sure to enclose your data set name in single quotes.

You are done when you successfully save the variables.

Steps for loading customization variables

This section tells you how to complete the ″Load customization variables″ option that is in all of the four main tasks you can perform.

When you first run the customization dialog, the dialog loads initial default customization variables. If you have previously run the dialog and saved your variables, you can follow these instructions to re-load those saved variables.

Before you begin: You must start the customization dialog and select your desired task option.

Perform the following steps to load customization variables:

1. On the main dialog panel, type L in the `Option` field to select ″Load customization variables″.

---

2. Press Enter.

   Result: You see the Load Customization Variables panel.:

```
-----------------     WebSphere for z/OS Customization      ------------------
Option  ===>

Load Customization Variables

 Specify the name of a data set containing the customization variables.
 IBM-supplied defaults are in 'BBO.SBBOEXEC(BBOWVARS)'
 Press Enter to continue.

 Data set name:


 If this data set is not cataloged, specify the volume.

 Volume:
```

3. Type in the information from "L Load customization variables for the base Application Server node" on page 78, "L Load customization variables for the Integral JMS Provider" on page 92, "L Load customization variables for the Deployment Manager node" on page 112, or "L Load customization variables for Web Services" on page 119, then press Enter.

   Attention: Be sure to enclose your data set name in single quotes.

You are done when you successfully load the variables.

1 Configure base Application Server node — worksheets, definitions and instructions

After following the steps in "Steps for starting the customization dialog" on page 49, choose option 1 and press Enter to configure your base Application Server node.

Result: You see the following dialog panel:

```
-----------------     WebSphere for z/OS Customization      ------------------
Option  ===>                                                     Appl: BBO5

 Configure base Application Server node

   Use this dialog to define WebSphere for z/OS variables and generate
   customization jobs for your installation. Specify an option and press ENTER.

   HLQ for WebSphere product data sets: BBO


   1  Allocate target data sets. The data sets will contain the WebSphere
      customization jobs and data generated by the dialog.

   2  Define variables. Define your installation-specific information for
      WebSphere customization.

   3  Generate customization jobs. Validate your customization variables
      and generate jobs and instructions.

   4  View instructions. View the generated customization instructions.


   Options for WebSphere for z/OS Customization Variables

   S  Save customization variables. Save your WebSphere customization
      variables in a data set for later use.

   L  Load customization variables. Load your WebSphere customization
      variables from a data set.
```

1 Allocate Target Data Sets for the base Application Server node

Table 13. Allocate target data sets

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
High Level Qualifier	(null)	

This panel asks you to specify the high-level qualifiers (hlq) for the target data sets. Target data sets are those into which the customization dialog places the customized jobs and other data. The data sets are:

hlq.CNTL
: A partitioned data set of fixed block, 80-byte records, that contains WebSphere for z/OS customization jobs.

hlq.DATA
: A partitioned data set of variable length records that contains other data produced by the customization dialog.

Worksheet

2 Define Variables to configure base Application Server node

1 System locations (directories, HLQs, etc):

System Locations (1 of 2):

Table 14. System Locations (1 of 2)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
System name	(System on which the customization dialog is running)	
Sysplex name	(Sysplex on which the cutsomization dialog is running)	
PROCLIB	SYS1.PROCLIB	
PARMLIB	SYS1.PARMLIB	
SYSEXEC	(blank)	
SCEEUN	CEE.SCEERUN	
		In link list or LPA?
SBBOLOAD	BBO.SBBOLOAD	
		In link list or LPA?
SBBOLD2	BBO.SBBOLD2	
		In link list or LPA?
SBBOMIG	BBO.SBBOMIG	
		In link list or LPA?
SBBOLPA	BBO.SBBOLPA	
		In link list or LPA?
SBBOEXEC	BBO.SBBOEXEC	
SBBOMSG	BBO.SBBOMSG	

This panel asks you for information about your base operating system and HFS-resident components.

System name
: The system name for the target z/OS system on which WebSphere for z/OS is installed.

Sysplex name
: The sysplex name for the target z/OS system on which WebSphere for z/OS is installed.

  Tip: If you are not sure what the system name (&SYSNAME) and sysplex name (&SYSPLEX) are, use the console command `D SYMBOLS` on the target z/OS system to display them.

For the following, specify the fully-qualified data set names without quotes.

PROCLIB
: An existing procedure library where the WebSphere for z/OS cataloged procedures are to be added.

PARMLIB
: An existing parameter library for system definitions to support WebSphere for z/OS. This data set must be in the parmlib concatenation for the target z/OS system.

SYSEXEC
: A variable-block (RECFM=VB, LRECL=255) data set into which the customization process places REXX EXECs to be called from TSO, such as the WebSphere for z/OS error log browser, BBORBLOG. You must allocate this data set and concatenate it as part of the SYSEXEC DD allocation in your installation–wide TSO logon PROC or allocation exec.

  If your existing SYSEXEC DD data set concatenation consists of fixed-blocked (RECFM=FB) data sets, you must make a **copy** of the *hlq*.DATA data set (produced by the customization dialog) after the customization process is complete, and place the copy in the SYSEXEC concatenation.

  If you do not specify a data set name, the customization process does not place any REXX EXECs in any data set.

Specify the following Language Environment and WebSphere for z/OS data sets and whether they are (“Y”) or are not (“N”) in the link list or the link pack area (LPA). “N” indicates the generated JCL will contain STEPLIB statements for these data sets. Refer to your SMP/E installation for the location of these data sets listed by their DD Name.

SCEERUN
: Your existing Language Environment run-time load module library.

SBBOLOAD
: WebSphere for z/OS load module library that you installed through SMP/E. It has members that should go into the link list or LPA.

SBBOLD2
: WebSphere for z/OS load module library that you installed through SMP/E. It has members that should go into the link list. **DO NOT** place them in LPA.

SBBOMIG
: WebSphere for z/OS IPCS data set that you installed through SMP/E. It is used not during normal operations, but for dump formatting in IPCS only. **DO NOT** place them in LPA.

SBBOLPA
: WebSphere for z/OS data set that you installed through SMP/E. It has members that should go into the link list or LPA.

Specify the following WebSphere for z/OS libraries so they can be accessed by the customized job streams the dialog produces. These data sets must be cataloged.

SBBOEXEC
: WebSphere for z/OS variable length file distribution PDS you installed through SMP/E.

SBBOMSG
: WebSphere for z/OS message skeletons for language translation you installed through SMP/E.

Worksheet

System Locations (2 of 2):

Table 15. System Locations (2 of 2)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Locations of HFS resident components		
WebSphere SMP/E home directory	/usr/lpp/zWebSphere/V5R0M0	
WebSphere JMS Client Java Feature SMP/E home directory	/usr/lpp/mqm/V5R3M1	
java home directory	/usr/lpp/java/IBM/J1.3	

Locations of HFS resident components:

WebSphere SMP/E home directory
: The name of the directory where WebSphere for z/OS files reside after SMP/E installation.

WebSphere JMS Client Java Feature SMP/E home directory
: The name of the directory where the WebSphere JMS Client Java Feature files reside after SMP/E installation.

java home directory
: The name of the directory where the Java SDK files reside after SMP/E installation.

2 System Environment Customization:

System Environment Customization (1 of 4):

Table 16. System Environment Customization (1 of 4)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
WebSphere HFS information		
Mount point	/WebSphere/V5R0M0	
Name	OMVS.WAS.CONFIG.HFS	
Volume, or '*' for SMS	*	
Primary allocation in cylinders	250	
Secondary allocation in cylinders	100	

WebSphere configuration HFS Information

Mount point
: Read/write HFS directory mount point where application data and environment files are written. The customization process creates this mount point, if it didn't already exist.

Name
: Hierarchical File System data set mounted at the above mount point.

  Rule: You can specify up to 42 characters for the data set name.

Volume, or '*' for SMS
: Specify either the DASD volume serial number containing the above data set or "*" to let SMS select a volume. Using "*" requires that SMS automatic class selection (ACS) routines be in place to select the volume. If you do not have SMS set up to handle data set allocation automatically, list the volume explicitly.

Primary allocation in cylinders
: Initial size allocation in cylinders for the above data set.

  Recommendation: The minimum suggested size is 250 cylinders (3390).

Secondary allocation in cylinders
: Size of each secondary extent in cylinders.

  Recommendation: The minimum suggested size is 100 cylinders.

Worksheet

System Environment Customization (2 of 4):

Table 17. System Environment Customization (2 of 4)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
WebSphere error log stream information		
Name	WAS.ERROR.LOG	
Data class	STANDARD	
Storage class	(null)	
HLQ for data sets	LOGGER	
Is logstream CF resident (Y|N)	Y	
If yes, structure name	WAS_STRUCT	
If no, specify:		
logstream size	3000	
staging size	3000	
RRS log stream information		
Group name	(Cell on which the customization dialog is running)	
Data class	STANDARD	
Storage class	(null)	
HLQ for data sets	LOGGER	
Is logstream CF resident (Y|N)	Y	
Create RRS PROC (Y|N)	Y	

WebSphere Error Logstream Information

Name Name of your WebSphere error log stream that is created.

Rules:

- The name must be 26 characters or fewer.
- Do NOT put quotes around it.

Data class
An existing DFSMS data class for the log stream data set allocation.

Storage class
An existing DFSMS storage class for allocation of the DASD staging data set for this log stream.

HLQ for data sets
The high-level qualifier for your log stream data set name and staging data set name that is created.

Is logstream CF resident (Y|N)
If you want the log stream to be created on a coupling facility, specify “Y”. If on DASD, specify “N”.

If yes, specify structure name
If using the coupling facility, specify the coupling facility structure to be used for the log stream.

Rule: The name can be 1 to 16 characters, including alphanumeric characters, national characters, and an underscore, where the first character is uppercase alphabetic.

If no, specify: logstream size
: Specifies the size, in 4K blocks, of the log stream DASD data sets for the log stream being defined.

If no, specify: staging size
: Specifies the size, in 4K blocks, of the DASD staging data set for the log stream being defined.

RRS Logstream Information

If you do not have the RRS log streams set up, the customization dialog will create the jobs you can use to set up the log streams.

Group name
: Specify the XCF group name.

  Recommendation: Use your cell name.

Data class
: Specify an existing DFSMS Data Class for the log stream data set allocation.

Storage class
: An existing DFSMS storage class for allocation of the DASD staging data set for this log stream.

HLQ for data sets
: The high-level qualifier for your log stream data set name and staging data set name.

Is logstream CF resident (Y|N)
: If the log stream is to be created on a coupling facility, specify “Y”. If on DASD, specify “N”.

Create RRS PROC (Y|N)
: If you answer “Y”, the dialog copies the ATRRRS cataloged procedure into SYS1.PROCLIB so that RRS can be started.

  If you already have RRS set up, specify “N”.

Worksheet

System Environment Customization (3 of 4):

Table 18. System Environment Customization (3 of 4)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
WebSphere Configuration Group Information		
Group	WSCFG1	
GID	2500	
WebSphere Administrator Information		
User ID	WSADMIN	
UID	2403	
Password	WSADMIN	
WebSphere Common Groups and User IDs		
Servant group for base servers	WSSR1	
Servant GID for base servers	2501	
Unauthenticated User Definitions for Base servers		
User ID	WSGUEST	
UID	2402	
Group	WSCLGP	
GID	2502	

This panel asks you to supply some RACF groups and user IDs that are common throughout WebSphere for z/OS. The dialog creates the RACF commands to define these new user IDs and groups for your security system.

To minimize the number of RACF definitions, RACF authorizations are at the group level rather than the user ID level. In a later panel, the dialog asks for user IDs for the run-time servants. These user IDs will be connected to their proper RACF groups.

For controllers, which run system authorized code, you can create a single group. Thus, the dialog creates a single RACF group for all controllers.

On the other hand, servants may have differing authorizations because they run application code and need access to differing resources. This dialog creates a RACF group for the WebSphere for z/OS base server.

Rules:

- User IDs and groups must be unique names (one to eight characters).
- UIDs (user identifiers) must be unique numbers within the system between 1 and 2,147,483,647.
- Do not assign a UID of 0 (Superuser) to any of these users.
- GIDs (group identifiers) should be unique numbers between 1 and 2,147,483,647.

WebSphere Configuration Group Information

Group The default group name for the WebSphere for z/OS administrator and base server. This group allows you to more easily control who can and can not make configuration changes to your servers.

Rule: These two users must have the same default group.

GID The group identifier for the WebSphere for z/OS configuration group.

WebSphere Administrator Information

User ID
: The user ID you use to log onto telnet to perform administrative actions against your server.

UID The user identifier for the WebSphere for z/OS user ID.

Password
: The password for the WebSphere for z/OS user ID.

WebSphere common groups and user IDs

Servant group for base servers
: Specifies an additional group name to which the base server's user ID connects. This is used to control access to resources that are external to the Application Server (for example, DB2).

Servant GID for base servers
: A group identifier that the dialog uses for the WebSphere for z/OS base server's servant group.

Unauthenticated user definitions for base servers

User ID
: The default user ID under which the unauthenticated client requests run.

UID The user identifier for the unauthenticated user.

Group The group for unauthenticated users.

GID The group identifier for unauthenticated users.

Worksheet

System Environment Customization (4 of 4):

Table 19. System Environment Customization (4 of 4)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
CTRACE Writer Definitions		
Procedure name	BBOWTR	
User ID	STCRACF	
Group	SYS1	
Trace Data Set Information		
Name	SYS1.*systemname*.WAS390.CTRACE	
Volume, or "*"for SMS	*	
Primary space in cylinders	10	
Secondary space in cylinders	0	

WebSphere for z/OS uses component trace (CTRACE) to capture and to display trace data in trace data sets. WebSphere for z/OS identifies itself to CTRACE with the with the component name ″SYSBBOSS″.

CTRACE Writer Definitions

Procedure name
: This is the CTRACE external writer start procedure to be created. It is identified in the WebSphere for z/OS CTRACE member (CTIBBOxx) in PARMLIB.

  Rule: The name can be 1 to 7 characters.

Userid
: RACF user ID to be created and associated with the CTRACE external writer start procedure.

Group
: RACF group name to be created and associated with this user.

Trace Data Set information

Name
: Specify a fully-qualified data set name, such as WAS390.CTRACE1, for the data set to be created. The default includes the system name of the system on which the customization dialog is running.

  Rules:

  - You can specify up to 42 characters for the data set name.
  - Do not use quotes.

Volume, or ″*″ for SMS
: Specify either the DASD volume serial number containing the above data set or "*" to let DFSMShsm select a volume. Using "*" requires SMS. Using "*" requires that SMS automatic class selection (ACS) routines be in place to select the volume. If you do not have SMS set up to handle data set allocation automatically, list the volume explicitly.

Primary space in cylinders
: The primary space for the trace data set.

Secondary space in cylinders
: The secondary space for the trace data set.

3 Server Customization: The WebSphere for z/OS run-time requires four base system servers: Application Server, Deployment Manager, node agent, location service daemon. The panels corresponding to the following tables set up the names, network configuration, start procedures, and user IDs for a base server.

Recommendation: Use the IBM default names the first time you install WebSphere for z/OS to make the installation instructions easier to follow.

For identification, each controller and servant start procedure must have a user ID and will be defined in the STARTED class. For more information, see "Cluster authorizations" on page 21.

Server Customization (1 of 4):

Table 20. Server Customization (1 of 4)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Application Server definitions		
WAS home directory	/WebSphere/V5R0M0/AppServer	
Cell name (short)	*sysplex*	
Cell name (long)	*sysplex*	
Node name (short)	*system*	
Node name (long)	*system*	
Server name (short)	BBOS001	
Server name (long)	server1	
Cluster transition name	BBOC001	

WAS home directory
: Directory in which the Application Server resides.

Cell name (short)
: Parameter passed to the server's start procedures that specifies the location of the cell's configuration files and identifies the cell to certain WebSphere for z/OS-exploited z/OS facilities (for example, SAF).

  Rule: Name must be 8 or fewer characters and all uppercase.

Cell name (long)
: Primary external identification of this WebSphere for z/OS cell. This name identifies the cell as displayed through the Administrative Console and also appears as part of the directory path under the WebSphere for z/OS home's configuration directory.

  Rule: Name must be 60 or fewer characters and can be of mixed case.

Node name (short)
: Parameter passed to the server's start procedures that specifies the location of the node's configuration files and identifies the node to certain WebSphere for z/OS-exploited z/OS facilities (for example, SAF).

  Rules:

  - Name must be 8 or fewer characters and all uppercase.
  - Name must be unique within the cell. The Application Server must be defined on its own node; no other server may exist on the same node as the Application Server.

Worksheet

Node name (long)
: Primary external identification of this WebSphere for z/OS node. This name identifies the node as displayed through the Administrative Console and also appears as part of the directory path under the WebSphere for z/OS home's configuration directory.

  Rules:

  - Name must be 60 or fewer characters and can be of mixed case.
  - Name must be unique within the cell. The Application Server must be defined on its own node; no other server may exist on the same node as the Application Server.

Server name (short)
: Name of the Application Server server. This is the server's jobname, as specified in the MVS START command JOBNAME parameter. This value is also passed as a parameter to the server's start procedures to specify the location of the server's configuration files and identify the server to certain WebSphere for z/OS- exploited z/OS facilities (for example, SAF).

  Rule: Name must be 8 or fewer characters and all uppercase.

Server name (long)
: Name of the Application Server server and the primary external identification of this WebSphere for z/OS server. This name identifies the server as displayed through the Administrative Console and also appears as part of the directory path under the WebSphere for z/OS home's configuration directory.

  Rule: Name must be 60 or fewer characters and can be of mixed case.

Cluster transition name
: WLM APPLENV name for this server.

  Note: If this server is converted to a cluster, this name becomes the cluster short name. The cluster short name is the WLM APPLENV name for all servers that are part of the same cluster.

  Rule: Name must be 8 or fewer characters and all uppercase.

Server Customization (2 of 4):

Table 21. Server Customization (2 of 4)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Application Server definitions		
Controller information		
Jobname	BBOS001	
Procedure name	BBO5ACR	
Userid	ASCR1	
UID	2431	
Servant information		
Jobname	BBOS001S	
Procedure name	BBO5ASR	
Userid	ASSR1	
UID	2432	

Controller information

Jobname
: The jobname, specified in the MVS START command JOBNAME parameter, associated with the Application Server controller.

Procedure name
: Name of member in your procedure library to start the Application Server controller.

User ID
: The user ID associated with the Application Server controller.

  Note: If you are using a non-IBM security system, the user ID may have to match the procedure name. Please refer to your security system's documentation.

UID
: The user identifier associated with this user ID.

  Rule: UIDs must be unique numbers, between 1 and 2,147,483,647, within the system.

Servant information

Jobname
: The jobname specified in the IWMSSNM parameter of the WLM Application Environment for the server.

Procedure name
: Name of member in your procedure library to start the Application Server servant.

User ID
: The user ID associated with the Application Server servant.

  Note: If you are using a non-IBM security system, the user ID may have to match the procedure name. Please refer to your security system's documentation.

Worksheet

UID	The user identifier associated with this user ID. **Rule:** UIDs must be unique numbers, between 1 and 2,147,483,647, within the system.

Server Customization (3 of 4):

Table 22. Server Customization (3 of 4)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Application Server definitions		
Node host name	(null)	
SOAP JMX Connector port	8880	
DRS Client Address port	7873	
ORB Listener host name	*	
ORB port	2809	
ORB SSL port	0	
HTTP transport host name	*	
HTTP port	9080	
HTTP SSL port	9443	

Application Server definitions

Node host name
: IP name or address of the system on which the server is configured. This value is used by other WebSphere for z/OS functions to connect to this server.

SOAP JMX Connector port
: Port number for the JMX HTTP connection to this server based on the SOAP protocol. JMX is used for remote administrative functions, such as invoking scripts through wsadmin.sh.

DRS Client Address port
: Port address for access to the server's data replication service. This is important for configurations that define replication groups.

ORB Listener host name
: IP address on which the server's ORB listens for incoming IIOP requests. The default is "*", which instructs the ORB to listen on all available IP addresses.

ORB port
: Port for IIOP requests which acts as the bootstrap port for this server and also as the port through which the ORB accepts IIOP requests. Port value cannot be 0.

ORB SSL port
: Port for secure IIOP requests. The default is "0", which allows the system to choose this port.

HTTP transport host name
: IP address on which the server's Web container should listen for incoming HTTP requests. The default is "*", which instructs the Web container to listen on all available IP addresses.

HTTP port
: Port for HTTP requests. Port value cannot be 0.

HTTP SSL port
: Port for secure HTTP requests. Port value cannot be 0.

Worksheet

Server Customization (4 of 4):

Table 23. Server Customization (4 of 4)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
location service daemon definitions		
Daemon Home Directory	/WebSphere/V5R0M0/Daemon	
Daemon job name	BBODMNB	
Procedure name	BBO5DMN	
Userid	WSDMNCR1	
UID	2411	
IP name	(null)	
Port	5655	
SSL Port	5656	

location service daemon definitions

The location service daemon is the initial point of contact in WebSphere for z/OS for clients and the server contains the location service agent to place sessions in a cell.

Daemon Home Directory
: Directory in which the location service daemon resides.

Daemon Job Name
: Specifies the jobname of the location service daemon, specified in the JOBNAME parameter of the MVS start command used to start the location service daemon.

  Caution: When configuring a second cell, ensure you change the daemon job name from the default or value you used for the first cell.

  Note: A server automatically starts the location service daemon if it is not already running.

Procedure name
: Name of the member in your procedure library to start the location service daemon.

User ID
: The user ID associated with the location service daemon.

UID
: The user identifier associated with this user ID.

  Rule: UIDs must be unique numbers, between 1 and 2,147,483,647, within the system.

IP Name
: The fully-qualified IP name, registered with the Domain Name Service (DNS), that the location service daemon will use.

  Note: In a sysplex, you should consider using a virtual IP address (VIPA) for the location service daemon IP name. Review the WebSphere for z/OS Installation and Customization Guide for information about configuring this value.

Port
: The port number on which the location service daemon listens.

SSL Port
: The port number on which the location service daemon listens for SSL connections.

Note: Select the IP name and port number for the location service daemon carefully. You can choose any name you want, but, once chosen, it is difficult to change, even in the middle of customization.

Worksheet

4 Security Customization: This panel allows you to specify authentication and authorization options for your run-time resources. For more information about security and WebSphere for z/OS, see “Setting up security” on page 18.

Table 24. Security Customization (1 of 1)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Use EJBROLE class to control EJB method access	N	
Use OPERMCDS to control commands	N	
Use Kerberos over SSL	N	
Use SSL basic authorization	N	
Use SSL client certificates	N	
Use passtickets	N	
Use CLASS(APPL)	N	
Test certificate authority label	WAS TestCertAuth	
PassTicket Profile name	CBS390	(cannot change)
PassTicket KEYMASK value	(null)	

In the following, specifying “Y” (yes) tells the dialog to define the profile or enable an option in RACF. Specifying “N” (no) tells the dialog to not define the profile or enable the option.

Use EJBROLE class to control EJB method access
: Specify “Y” to create the RACF EJBROLE class. EJBROLE controls method accesses for enterprise beans.

Use OPERCMDS to control commands
: Specify “Y” to create the RACF OPERCMDS class. OPERCMDS controls the ability of an operator to start servers.

Use Kerberos over SSL
: Specify “Y” to create Kerberos security. With this option, SSL provides message security and authenticates the server to the client. Kerberos provides the ability for the server to authenticate the client.

Use SSL basic authorization
: Specify “Y” to create SSL basic authorization security. With this option, the server proves its identity by passing a digital certificate to the client. The client proves its identity by passing a user identity and password known by the target server.

Use SSL client certificates
: Specify “Y” to create SSL client certificate security. With this option, both the server and client pass digital certificates to prove their identities to each other.

Use passtickets
: Specify “Y” to use passtickets.

Use CLASS(APPL)
: If the APPL class is activated in your installation, specify “Y” to create a CBS390 profile in that APPL class to represent WebSphere for z/OS and permit the WebSphere for z/OS server identities to that profile. If your installation does not require use of the APPL class, specify “N”.

Test certificate authority label
: The dialog uses this label to create a test certificate authority certificate.

  Recommendation: Use this certificate for testing purposes only.

PassTicket Profile name
: Name of the PassTicket Profile

PassTicket KEYMASK value
: Specify any string of 16 hexadecimal characters as a mask for PassTickets.

Worksheet

3 Generate customization jobs for the base Application Server node

Table 25. Generate customization jobs

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Job card information	//jobname JOB (ACCTNO,ROOM),'*userid*',CLASS=A,REGION=0M //* //* //*	

Specify the job card according to your installation requirements.

Note: The dialog generates a job name and the ″JOB″ keyword for each job.

S Save customization variables for the base Application Server node

Table 26. Save customization variables

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Dsname	(null)	

Specify the name of the data set into which you want to save the customization variables.

Rules:

- The data set must be a sequential data set. Do not specify a member name.
- Place quotes around the data set name.

Worksheet

L Load customization variables for the base Application Server node

Table 27. Load customization variables

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Dsname	(null)	

Specify the data set from which you will prime the variables.

Following the customized Application Server node instructions

The major stages of the Application Server node customization process are:

Stage	Description
1	Make a variety of configuration changes to your z/OS system configuration (PARMLIB, TCP/IP, workload management, and so forth). The customized instructions provide details and pointers to relevant documentation.
2	If you require message translation, set up MMS to translate messages for WebSphere for z/OS with job BBOMSGC.
3	Define log streams used by WebSphere for z/OS and RRS through jobs BBOERRLG and BBORRSLS. You do not need to run BBORRSLS if RRS is already running on your target MVS system.
4	Allocate the CTRACE data set through job BBOWCTR.
5	Create a customized set of RACF commands for initial WebSphere for z/OS security setup through job BBOCBRAJ. The RACF commands are saved in member BBOCBRAK of the *hlq*.DATA data set. Job BBOCBRAK executes these RACF commands. Later, you can use the RACF commands saved in BBOCBRAK to help in defining security for additional servers or users.
6	Allocate and mount the WebSphere for z/OS run-time HFS through job BBOWCHFS. If your root HFS is mounted read-only, you may need to define one or more mount points manually. See the instructions for details. Job BBOMCFG creates subdirectories in the WebSphere for z/OS run-time HFS and job BBOMCFGU, which is optional, configures the runtime HFS to include the directory for UDDIReg.
7	Copy customized PARMLIB and PROCLIB members into their proper locations using job BBOWCPY1.
8	Copy customized HFS files into their proper locations using job BBOWCPY2.
9	Create the was.env file for the location service daemon and the application servers through job BBOWC2N.
10	Install the Administrative Console and the IVT application through job BBOWIAPP.
11	Complete the HFS initialization with job BBOMCFG2.
12	Activate Resource Recovery Services (RRS), if it is not already active, with the MVS command `START ATRRRS,SUB=MSTR`.
13	Start the CTRACE writer used by WebSphere for z/OS with the MVS command `TRACE CT,WTRSTART=BBOWTR`.

14	Start the Application Server with the following MVS command:
	`S <controller_procname>,JOBNAME=server_shortname,`
	`  ENV=<cell_shortname.node_shortname.server_shortname>`
	
	This process causes information to be stored in the WebSphere for z/OS databases and HFS. Once the process is complete, you have a working WebSphere for z/OS run-time that you will use to run the Installation Verification Test (using job BBOWIVT) and your own applications.

15	Use BBOW5SH to execute certain shell scripts from the WebSphere for z/OS bin directory in order to carry out certain administrative tasks.

2 Configure integral JMS provider — worksheets, definitions and instructions

Rule: You must configure your base Application Server node before you configure your integral JMS provider. See "1 Configure base Application Server node — worksheets, definitions and instructions" on page 57 for more information.

After following the steps in "Steps for starting the customization dialog" on page 49, choose option 2 and press Enter to configure your integral JMS provider.

Result: You see the following dialog panel:

```
-----------------     WebSphere for z/OS Customization      ------------------
Option  ===>                                                     Appl: BBO5

 Configure integral JMS provider.

   Use this dialog to define WebSphere for z/OS variables and generate
   customization jobs for your installation. Specify an option and press ENTER.

   HLQ for WebSphere product data sets: BBO


   1  Allocate target data sets. The data sets will contain the WebSphere
      customization jobs and data generated by the dialog.

   2  Define variables. Define your installation-specific information for
      WebSphere customization.

   3  Generate customization jobs. Validate your customization variables
      and generate jobs and instructions.

   4  View instructions. View the generated customization instructions.


   Options for WebSphere for z/OS Customization Variables

   S  Save customization variables. Save your WebSphere customization
      variables in a data set for later use.

   L  Load customization variables. Load your WebSphere customization
      variables from a data set.
```

1 Allocate Target Data Sets for the Integral JMS Provider

Table 28. Allocate target data sets

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
High Level Qualifier	(null)	

This panel asks you to specify the high-level qualifiers (hlq) for the target data sets. Target data sets are those into which the customization dialog places the customized jobs and other data. The data sets are:

hlq.CNTL
: A partitioned data set of fixed block, 80-byte records, that contains WebSphere for z/OS customization jobs.

hlq.DATA
: A partitioned data set of variable length records that contains other data produced by the customization dialog.

2 Define Variables to configure Integral JMS Provider

1 System locations (directories, HLQs, etc):

System Locations (1 of 2):

Table 29. System Locations (1 of 2)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
System name	(System on which the customization dialog is running)	
Sysplex name	(Sysplex on which the cutsomization dialog is running)	
Jes3 (Y/N)	N	
MACLIB	SYS1.MACLIB	
PROCLIB	SYS1.PROCLIB	
PARMLIB	SYS1.PARMLIB	
SCSQAUTH	CSQ531.SCSQAUTH	
		In link list or LPA?
SCSQANLx	CSQ531.SCSQANLE	
		In link list or LPA?
SCSQLINK	CSQ531.SCSQLINK	
		In link list or LPA?
SCSQMVR1	CSQ531.SCSQMVR1	
		In link list or LPA?
SCEEUN	CEE.SCEERUN	
		In link list or LPA?
SCSQLOAD	CSQ531.SCSQLOAD	
		In link list or LPA?
SCSQMACS	CSQ531.SCSQMACS	
		In link list or LPA?
SCSQPROC	CSQ531.SCSQPROC	
		In link list or LPA?
SCSQSNLx	CSQ531.SCSQSNLE	
		In link list or LPA?
User load	CSQ531.USER.LOAD	
		In link list or LPA?

This panel asks you for information about your base operating system, HFS-resident components, and MQ subsystem.

System name
: The system name, designated in your base system customization, for the target z/OS system on which WebSphere for z/OS is installed.

Sysplex name
: The sysplex name, designated in your base system customization, for the target z/OS system on which WebSphere for z/OS is installed.

Tip: If you are not sure what the system name (&SYSNAME) and sysplex name (&SYSPLEX) are, use the console command `D SYMBOLS` on the target z/OS system to display them.

Jes3 (Y/N)
Indicate ″Y″ if you are using JES 3 or ″N″ if you are using JES 2.

For the following, specify the fully-qualified data set names without quotes.

Rule: You can specify up to 42 characters for the data set names.

MACLIB

PROCLIB
An existing procedure library where the WebSphere for z/OS cataloged procedures are added.

PARMLIB
An existing parameter library for system definitions to support WebSphere for z/OS.

Specify the following Language Environment and WebSphere for z/OS data sets and whether they are (“Y”) or are not (“N”) in the link list or the link pack area (LPA). “N” indicates the generated JCL will contain STEPLIB statements for these data sets. Refer to your SMP/E installation for the location of these data sets listed by their DD Name.

SCSQAUTH
The main repository for all MQSeries product load modules. It also contains the default parameter modules CSQZPARM and CSQXPARM. Ensure this library is APF-authorized.

SCSQANLx
Contains the load modules for various versions of MQSeries. The different versions, designated by letters in place of the ″x″, are U.S. English: mixed case (″E″), U.S. English: uppercase (″U″), Simplified Chinese (″C″), and Japanese (″K″).

SCSQLINK
The early code library. Contains the load modules that must reside in the link list because they are loaded at system initial program load (IPL). Ensure this library is APF-authorized and in the link list.

SCSQMVR1
Contains the load modules for distributed queuing when using LU 6.2 or TCP/IP with either the OpenEdition sockets or IUCV interface. Ensure this library is APF-authorized.

SCEERUN
The LE runtime library. Access is required. If it is not in your link list, concatenate it in the STEPLIB DD statement. You need to stop and restart your queue manager to do this.

SCSQLOAD
The load library. Contains load modules for non-APF code, user exits, utilities, samples, installation verification programs, and adapter stubs. The library does not need APF-authorization nor must it reside in the link list.

SCSQMACS
Contains Assembler macros including sample macros, product macros, and system parameter macros.

SCSQPROC
: Contains sample JCL and default system initialization data sets.

SCSQSNLx
: Contains the load modules for various versions of the MQSeries modules that are required for special purpose function (for example, the early code). The different versions, designated by letters in place of the ″x″, are U.S. English: mixed case (″E″), U.S. English: uppercase (″U″), Simplified Chinese (″C″), and Japanese (″K″).

Specify and catalog the following parameter linkedit target data sets.

User load
: Controls the logging, archiving, tracing, and connection environments that MQSeries uses in its operation.

Worksheet

System Locations (2 of 2):

Table 30. System Locations (2 of 2)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
High Level Qualifier of IJP Operational Datasets		
CSQHLQ	CSQ531	
Integral Provider Output Volumes		
BSDS Vol1	(blank)	
BSDS Vol2	(blank)	
LogCopy 1 Vol1	(blank)	
LogCopy 1 Vol2	(blank)	
LogCopy 2 Vol1	(blank)	
LogCopy 2 Vol2	(blank)	
Integral Provider Page Volumes		
PageSet Vol0	(blank)	
PageSet Vol1	(blank)	
PageSet Vol2	(blank)	
PageSet Vol3	(blank)	
PageSet Vol4	(blank)	
PageSet Vol5	(blank)	
PageSet Vol6	(blank)	
PageSet Vol7	(blank)	
PageSet Vol8	(blank)	

Each MQSeries subsystem in WebSphere for z/OS contains two bootstrap data sets (BSDSs), two sets of log data sets, and nine page data sets. Along with the volume 0 page data set, which is fixed and untailorable, there are eight other page data sets that you can organize to suit your needs.

Note: You must define all the BSDS, LogCopy, and PageSet volumes, but you can choose to make the values different or all the same. You might specify the same value for all the volumes if, for example, you run only a test system, but the system performs better if they are spread out.

High Level Qualifier of IJP Operational Data sets:

CSQHLQ
The high level qualifier of your IJP operational data sets.

Integral Provider Output Volumes

BSDS Vol1
The bootstrap data set volume 1.

BSDS Vol2
The bootstrap data set volume 2.

LogCopy 1 Vol1
Volume 1 of copy 1 of the log data set.

LogCopy 1 Vol2
Volume 2 of copy 1 of the log data set.

LogCopy 2 Vol1
Volume 1 of copy 2 of the log data set.

LogCopy 2 Vol2
Volume 2 of copy 2 of the log data set.

Integral Provider Page Volumes

PageSet Vol0
The volume 0 page data set, which is fixed and untailorable.

PageSet Vol1 - PageSet Vol8
Eight volumes of page data sets. You must define them all, but you can choose to make the values different or all the same.

Worksheet

2 Server Customization: These panels allow you to define values for your Integral JMS Provider. The values you initially set for your base application server and location service daemon are displayed for reference purposes only. If you wish to change them, you need to go back and reconfigure your base Application Server.

Recommendation: Use the IBM default names the first time you install WebSphere for z/OS to make the installation instructions easier to follow.

For identification, each controller and servant start procedure must have a user ID and will be defined in the STARTED class. For more information, see "Cluster authorizations" on page 21.

Server Customization (1 of 2):

Table 31. Server Customization (1 of 2)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
WebSphere HFS Information		
Mount point	/WebSphere/V5R0M0	(cannot change)
Application Server definitions		
WAS home directory	/WebSphere/V5R0M0/AppServer	(cannot change)
Cell name (short)	*sysplex*	(cannot change)
Cell name (long)	*sysplex*	(cannot change)
Node name (short)	*system*	(cannot change)
Node name (long)	*system*	(cannot change)
Server name (short)	BBOS001	(cannot change)
Server name (long)	server1	(cannot change)
Integral JMS definitions		
JMS Server name (short)	WMQX	
Command Prefix	+	

WebSphere HFS Information

Mount point
: Read/write HFS directory mount point where application data and environment files are written. The customization process creates this mount point, if it didn't already exist.

Application Server definitions

WAS home directory
: Directory in which the Application Server resides.

Cell name (short)
: Parameter passed to the server's start procedures that specifies the location of the cell's configuration files and identifies the cell to certain WebSphere for z/OS-exploited z/OS facilities (for example, SAF).

Cell name (long)
: Primary external identification of this WebSphere for z/OS cell. This name identifies the cell as displayed through the Administrative Console and also appears as part of the directory path under the WebSphere for z/OS home's configuration directory.

Node name (short)
: Parameter passed to the server's start procedures that specifies the location of the node's configuration files and identifies the node to certain WebSphere for z/OS-exploited z/OS facilities (for example, SAF).

Node name (long)
: Primary external identification of this WebSphere for z/OS node. This name identifies the node as displayed through the Administrative Console and also appears as part of the directory path under the WebSphere for z/OS home's configuration directory.

Server name (short)
: Name of the Application Server server. This is the server's jobname, as specified in the MVS START command JOBNAME parameter. This value is also passed as a parameter to the server's start procedures to specify the location of the server's configuration files and identify the server to certain WebSphere for z/OS- exploited z/OS facilities (for example, SAF).

Server name (long)
: Name of the Application Server server and the primary external identification of this WebSphere for z/OS server. This name identifies the server as displayed through the Administrative Console and also appears as part of the directory path under the WebSphere for z/OS home's configuration directory.

Integral JMS definitions

JMS Server name (short)
: MQSeries subsystem base server name. This value defines the Integral JMS Queue Manager as a z/OS subsystem and forms the name of the Integral JMS Broker.

  Rule: The JMS Server name must contain only 4 characters.

Command Prefix
: Character string value that WebSphere System Manager uses, in conjunction with the JMS Server short name, to form system commands that direct both the operation and administration of the Integral JMS Queue Manager.

Worksheet

Server Customization (2 of 2):

Table 32. Server Customization (2 of 2)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Integral JMS Server Parameters		
Archive retention (days)	9999	
Stat Interval Time (secs)	30	
JMS Server Queued Address port	5558	
JMS Server Direct Address port	5559	
JMS Server Security port	5557	

Integral JMS Server Parameters

Archive retention (days)
: Specifies the retention period, in days, set when you create the archive log data set. The default is 9999.

  Rule: The parameter must be in the range 0 through 9999.

Stat Interval Time (secs)
: The interval, in seconds, between listener restart attempts. The default is 60.

JMS Server Queued Address port

JMS Server Direct Address port

JMS Server Security port

3 Security Customization: This panel allows you to specify authentication and authorization options for your run-time resources. For more information about security and WebSphere for z/OS, see “Setting up security” on page 18.

Table 33. Security Customization (1 of 1)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Queue Manager Security		
Default User ID for Commands	(blank)	
Resource Auditing (Y/N)	Y	
SAF Authentication (Y/N)	Y	

Default User ID for Commands
: Specifies the default user ID used for command security checks. Ensure this user ID is defined to the ESM.

  Rule: The name must be between 1 and 8 alphanumeric characters and start with a letter.

In the following, specifying “Y” (yes) tells the dialog to define the profile or enable an option in RACF. Specifying “N” (no) tells the dialog to not define the profile or enable the option.

Resource Auditing (Y/N)
: If you want to enable the creation of an audit trail, specify ″Y″. Otherwise, specify ″N″. Creating an audit trail may impact performance.

SAF Authentication (Y/N)
: If you want your queue manager to have SAF authentication, specify ″Y″. Otherwise, specify ″N″. Specifying ″Y″ will authenticate userids and passwords and perhaps impact performance.

Worksheet

3 Generate customization jobs for the Integral JMS Provider

Table 34. Generate customization jobs

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Job card information	//jobname JOB (ACCTNO,ROOM),'*userid*',CLASS=A,REGION=0M //* //* //*	

Specify the job card according to your installation requirements.

Note: The dialog generates a job name and the ″JOB″ keyword for each job.

S Save customization variables for the Integral JMS Provider

Table 35. Save customization variables

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Dsname	(null)	

Specify the name of the data set into which you want to save the customization variables.

Rules:

- The data set must be a sequential data set. Do not specify a member name.
- Place quotes around the data set name.

Worksheet

L Load customization variables for the Integral JMS Provider

Table 36. Load customization variables

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Dsname	(null)	

Specify the data set from which you will prime the variables.

Following the customized integral JMS provider instructions

The major stages of the integral JMS provider customization process are:

Stage	Description
1	Make a variety of configuration changes to your z/OS system configuration. The customized instructions provide details and pointers to relevant documentation.
2	Establish security identity for the MQ started tasks.
3	Creates dual bootstrap data sets for usage by the Integral JMS Provider WebSphere MQ Queue Manager with job BBOJBSDS.
4	Create page data sets for usage by the Integral JMS Provider WebSphere MQ Queue Manager with job BBOJPAGR.
5	Assemble your parameters with jobs BBOJXPRR and BBOJZPRR. BBOJXPRR assembles the X parameters for usage by the Integral JMS Provider WebSphere MQ Channel Initiator, and BBOJZPRR assembles the Z parameters for usage by the Integral JMS Provider WebSphere MQ Queue Manager.
6	Copy the tailored start procedures and parameters to the run-time libraries with jobs BBOWCPYJ and BBOWCPYZ.
7	Stop the Application Server with the MVS command `STOP BBOS001`.
8	Run job BBOWCPJ2 to write z/OS UNIX resident files into the HFS for use by the application server.
9	Initialize the Integral JMS Provider runtime environment in the HFS for the application server with job BBOWJWC.

10	Start the Application Server with the following MVS command:
	`S <controller_procname>,JOBNAME=server_shortname,`
	`ENV=<cell_shortname.node_shortname.server_shortname>`
	
	This process causes information to be stored in the WebSphere for z/OS databases and HFS. Once the process is complete, you have a working WebSphere for z/OS run-time that you will use to run the Installation Verification Test (using job BBOWIVT) and your own applications.

3 Configure Deployment Manager node — worksheets, definitions and instructions

Rule: You must configure your base Application Server node and integral JMS provider before you configure your Deployment Manager node. See "1 Configure base Application Server node — worksheets, definitions and instructions" on page 57 and "2 Configure integral JMS provider — worksheets, definitions and instructions" on page 80 for more information.

After following the steps in "Steps for starting the customization dialog" on page 49, choose option 3 and press Enter to configure your Deployment Manager node.

Result: You see the following dialog panel:

```
-----------------     WebSphere for z/OS Customization      ------------------
Option  ===>                                                      Appl: BBO5

 Configure Deployment Manager node

   Use this dialog to define WebSphere for z/OS variables and generate
   customization jobs for your installation. Specify an option and press ENTER

   HLQ for WebSphere product data sets: BBO


   1  Allocate target data sets. The data sets will contain the WebSphere
      customization jobs and data generated by the dialog.

   2  Define variables. Define your installation-specific information for
      WebSphere customization.

   3  Generate customization jobs. Validate your customization variables
      and generate jobs and instructions.

   4  View instructions. View the generated customization instructions.


   Options for WebSphere for z/OS Customization Variables

   S  Save customization variables. Save your WebSphere customization
      variables in a data set for later use.

   L  Load customization variables. Load your WebSphere customization
      variables from a data set.
```

1 Allocate Target Data Sets for the Deployment Manager node

Table 37. Allocate target data sets

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
High Level Qualifier	(null)	

This panel asks you to specify the high-level qualifiers (hlq) for the target data sets. Target data sets are those into which the customization dialog places the customized jobs and other data. The data sets are:

hlq.CNTL
: A partitioned data set of fixed block, 80-byte records, that contains WebSphere for z/OS customization jobs.

hlq.DATA
A partitioned data set of variable length records that contains other data produced by the customization dialog.

Worksheet

2 Define Variables to configure Deployment Manager node

1 System locations (directories, HLQs, etc):

System Locations (1 of 2):

Table 38. System Locations (1 of 2)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
System name	(System on which the customization dialog is running)	
Sysplex name	(Sysplex on which the cutsomization dialog is running)	
PROCLIB	SYS1.PROCLIB	
PARMLIB	SYS1.PARMLIB	
SYSEXEC	(blank)	
SCEEUN	CEE.SCEERUN	
		In link list or LPA?
SBBOLOAD	BBO.SBBOLOAD	
		In link list or LPA?
SBBOLD2	BBO.SBBOLD2	
		In link list or LPA?
SBBOMIG	BBO.SBBOMIG	
		In link list or LPA?
SBBOLPA	BBO.SBBOLPA	
		In link list or LPA?
SBBOEXEC	BBO.SBBOEXEC	
SBBOMSG	BBO.SBBOMSG	

This panel asks you for information about your base operating system and HFS-resident components.

System name
: The system name for the target z/OS system on which WebSphere for z/OS is installed.

Sysplex name
: The sysplex name for the target z/OS system on which WebSphere for z/OS is installed.

  Tip: If you are not sure what the system name (&SYSNAME) and sysplex name (&SYSPLEX) are, use the console command `D SYMBOLS` on the target z/OS system to display them.

For the following, specify the fully-qualified data set names without quotes.

PROCLIB
: An existing procedure library where the WebSphere for z/OS cataloged procedures are to be added.

PARMLIB
: An existing parameter library for system definitions to support WebSphere for z/OS. This data set must be in the parmlib concatenation for the target z/OS system.

SYSEXEC
: A variable-block (RECFM=VB, LRECL=255) data set into which the customization process places REXX EXECs to be called from TSO, such as the WebSphere for z/OS error log browser, BBORBLOG. You must allocate this data set and concatenate it as part of the SYSEXEC DD allocation in your installation–wide TSO logon PROC or allocation exec.

  If your existing SYSEXEC DD data set concatenation consists of fixed-blocked (RECFM=FB) data sets, you must make a **copy** of the *hlq*.DATA data set (produced by the customization dialog) after the customization process is complete, and place the copy in the SYSEXEC concatenation.

  If you do not specify a data set name, the customization process does not place any REXX EXECs in any data set.

Specify the following Language Environment and WebSphere for z/OS data sets and whether they are (“Y”) or are not (“N”) in the link list or the link pack area (LPA). “N” indicates the generated JCL will contain STEPLIB statements for these data sets. Refer to your SMP/E installation for the location of these data sets listed by their DD Name.

SCEERUN
: Your existing Language Environment run-time load module library.

SBBOLOAD
: WebSphere for z/OS load module library that you installed through SMP/E. It has members that should go into the link list or LPA.

SBBOLD2
: WebSphere for z/OS load module library that you installed through SMP/E. It has members that should go into the link list. **DO NOT** place them in LPA.

SBBOMIG
: WebSphere for z/OS IPCS data set that you installed through SMP/E. It is used not during normal operations, but for dump formatting in IPCS only. **DO NOT** place them in LPA.

SBBOLPA
: WebSphere for z/OS data set that you installed through SMP/E. It has members that should go into the link list or LPA.

Specify the following WebSphere for z/OS libraries so they can be accessed by the customized job streams the dialog produces. These data sets must be cataloged.

SBBOEXEC
: WebSphere for z/OS variable length file distribution PDS you installed through SMP/E.

SBBOMSG
: WebSphere for z/OS message skeletons for language translation you installed through SMP/E.

Worksheet

System Locations (2 of 2):

Table 39. System Locations (2 of 2)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Locations of HFS resident components		
WebSphere SMP/E home directory	/usr/lpp/zWebSphere/V5R0M0	
WebSphere JMS Client Java Feature SMP/E home directory	/usr/lpp/mqm/V5R3M1	
java home directory	/usr/lpp/java/IBM/J1.3	

Locations of HFS resident components:

WebSphere SMP/E home directory
: The name of the directory where WebSphere for z/OS files reside after SMP/E installation.

WebSphere JMS Client Java Feature SMP/E home directory
: The name of the directory where the WebSphere JMS Client Java Feature files reside after SMP/E installation.

java home directory
: The name of the directory where the Java SDK files reside after SMP/E installation.

2 System Environment Customization:

System Environment Customization (1 of 1):

Table 40. System Environment Customization (1 of 4)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
WebSphere HFS information		
Mount point	/WebSphere/V5R0M0	
Name	OMVS.WAS.CONFIG.HFS	
Volume, or '*' for SMS	*	
Primary allocation in cylinders	250	
Secondary allocation in cylinders	100	

WebSphere configuration HFS Information

Mount point
: Read/write HFS directory mount point where application data and environment files are written. The customization process creates this mount point, if it didn't already exist.

Name
: Hierarchical File System data set mounted at the above mount point.

  Rule: You can specify up to 42 characters for the data set name.

Volume, or '*' for SMS
: Specify either the DASD volume serial number containing the above data set or "*" to let SMS select a volume. Using "*" requires that SMS automatic class selection (ACS) routines be in place to select the volume. If you do not have SMS set up to handle data set allocation automatically, list the volume explicitly.

Primary allocation in cylinders
: Initial size allocation in cylinders for the above data set.

  Recommendation: The minimum suggested size is 250 cylinders (3390).

Secondary allocation in cylinders
: Size of each secondary extent in cylinders.

  Recommendation: The minimum suggested size is 100 cylinders.

Worksheet

3 Server Customization: The WebSphere for z/OS run-time requires four base system servers: Application Server, Deployment Manager, node agent, location service daemon. The panels corresponding to the following tables set up the names, network configuration, start procedures, and user IDs for a Deployment Manager server.

Recommendation: Use the IBM default names the first time you install WebSphere for z/OS to make the installation instructions easier to follow.

For identification, each controller and servant start procedure must have a user ID and will be defined in the STARTED class. For more information, see "Cluster authorizations" on page 21.

Server Customization (1 of 4):

Table 41. Server Customization (1 of 4)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Deployment Manager definitions		
WAS home directory	/WebSphere/V5R0M0/DeploymentManager	
Cell name (short)	*sysplex*	
Cell name (long)	*sysplex*	
Node name (short)	*system*	
Node name (long)	*system*	
Server name (short)	BBODMGR	
Server name (long)	dmgr	
Cluster transition name	BBODMGR	

WAS home directory
: Directory in which the Deployment Manager resides.

Cell name (short)
: Parameter passed to the server's start procedures that specifies the location of the cell's configuration files and identifies the cell to certain WebSphere for z/OS-exploited z/OS facilities (for example, SAF).

  Rule: Name must be 8 or fewer characters and all uppercase.

Cell name (long)
: Primary external identification of this WebSphere for z/OS cell. This name identifies the cell as displayed through the Administrative Console and also appears as part of the directory path under the WebSphere for z/OS home's configuration directory.

  Rule: Name must be 60 or fewer characters and can be of mixed case.

Node name (short)
: Parameter passed to the server's start procedures that specifies the location of the node's configuration files and identifies the node to certain WebSphere for z/OS-exploited z/OS facilities (for example, SAF).

  Rules:

  - Name must be 8 or fewer characters and all uppercase.

- Name must be unique within the cell. The Deployment Manager must be defined on its own node; no other server may exist on the same node as the Deployment Manager.

Node name (long)
: Primary external identification of this WebSphere for z/OS node. This name identifies the node as displayed through the Administrative Console and also appears as part of the directory path under the WebSphere for z/OS home's configuration directory.

  Rules:

  - Name must be 60 or fewer characters and can be of mixed case.
  - Name must be unique within the cell. The Deployment Manager must be defined on its own node; no other server may exist on the same node as the Deployment Manager.

Server name (short)
: Name of the Deployment Manager server. This is the server's jobname, as specified in the MVS START command JOBNAME parameter. This value is also passed as a parameter to the server's start procedures to specify the location of the server's configuration files and identify the server to certain WebSphere for z/OS- exploited z/OS facilities (for example, SAF).

  Rule: Name must be 8 or fewer characters and all uppercase.

Server name (long)
: Name of the Deployment Manager server and the primary external identification of this WebSphere for z/OS server. This name identifies the server as displayed through the Administrative Console and also appears as part of the directory path under the WebSphere for z/OS home's configuration directory.

  Rule: Name must be 60 or fewer characters and can be of mixed case.

Cluster transition name
: WLM APPLENV name for this server.

  Note: The Deployment Manager is not clusterable, so this value never actually becomes the cluster short name of this server's cluster. However, like an Application Server, the Deployment Manager still needs an APPLENV, so the cluster transition name is used for this purpose.

  Rule: Name must be 8 or fewer characters and all uppercase.

Worksheet

Server Customization (2 of 4):

Table 42. Server Customization (2 of 4)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Deployment Manager definitions		
Controller information		
Jobname	BBODMGR	(cannot change)
Procedure name	BBO5DCR	
Userid	DMCR1	
UID	2421	
Servant information		
Jobname	BBODMGRS	(cannot change)
Procedure name	BBO5DSR	
Userid	DMSR1	
UID	2422	

Controller information

Jobname
: The jobname, specified in the MVS START command JOBNAME parameter, associated with the Deployment Manager controller.

Procedure name
: Name of member in your procedure library to start the Deployment Manager controller.

User ID
: The user ID associated with the Deployment Manager controller.

  Note: If you are using a non-IBM security system, the user ID may have to match the procedure name. Please refer to your security system's documentation.

UID
: The user identifier associated with this user ID.

  Rule: UIDs must be unique numbers, between 1 and 2,147,483,647, within the system.

Servant information

Jobname
: The jobname specified in the IWMSSNM parameter of the WLM Application Environment for the server.

Procedure name
: Name of member in your procedure library to start the Deployment Manager servant.

User ID
: The user ID associated with the Application Server servant.

  Note: If you are using a non-IBM security system, the user ID may have to match the procedure name. Please refer to your security system's documentation.

UID The user identifier associated with this user ID.

Rule: UIDs must be unique numbers, between 1 and 2,147,483,647, within the system.

Worksheet

Server Customization (3 of 4):

Table 43. Server Customization (3 of 4)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Deployment Manager definitions		
Node host name	(null)	
SOAP JMX Connector port	8879	
CELL DISCOVERY ADDRESS port	7277	
DRS Client Address port	7989	
ORB Listener host name	*	
ORB port	9809	
ORB SSL port	0	
HTTP transport host name	*	
HTTP port	9090	
HTTP SSL port	9043	

Deployment Manager definitions

Node host name
: IP name or address of the system on which the server is configured. This value is used by other WebSphere for z/OS functions to connect to this server.

SOAP JMX Connector port
: Port number for the JMX HTTP connection to this server based on the SOAP protocol. JMX is used for remote administrative functions, such as invoking scripts through wsadmin.sh.

CELL DISCOVERY ADDRESS port
: Port number used by node agents to connect to this Deployment Manager server.

DRS Client Address port
: Port address for access to the server's data replication service. This is important for configurations that define replication groups.

ORB Listener host name
: IP address on which the server's ORB listens for incoming IIOP requests. The default is "*", which instructs the ORB to listen on all available IP addresses.

ORB port
: Port for IIOP requests which acts as the bootstrap port for this server and also as the port through which the ORB accepts IIOP requests. Port value cannot be 0.

ORB SSL port
: Port for secure IIOP requests. The default is "0", which allows the system to choose this port.

HTTP transport host name
: IP address on which the server's Web container should listen for incoming HTTP requests. The default is "*", which instructs the Web container to listen on all available IP addresses.

HTTP port
: Port for HTTP requests. Port value cannot be 0.

HTTP SSL port
: Port for secure HTTP requests. Port value cannot be 0.

Worksheet

Server Customization (4 of 4):

Table 44. Server Customization (4 of 4)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
location service daemon definitions		
Daemon Home Directory	/WebSphere/V5R0M0/Daemon	(cannot change)
Daemon job name	BBODMNC	
Procedure name	BBO5DMN	
Userid	WSDMNCR1	
UID	2411	
IP name	(null)	
Port	5755	
SSL Port	5756	

location service daemon definitions

The location service daemon is the initial point of contact in WebSphere for z/OS for clients and the server contains the location service agent to place sessions in a cell.

Daemon Home Directory
: Directory in which the location service daemon resides.

Daemon Job Name
: Specifies the jobname of the location service daemon, specified in the JOBNAME parameter of the MVS start command used to start the location service daemon.

  Caution: When configuring a second cell, ensure you change the daemon job name from the default or value you used for the first cell.

  Note: A server automatically starts the location service daemon if it is not already running.

Procedure name
: Name of the member in your procedure library to start the location service daemon.

User ID
: The user ID associated with the location service daemon.

UID
: The user identifier associated with this user ID.

  Rule: UIDs must be unique numbers, between 1 and 2,147,483,647, within the system.

IP Name
: The fully-qualified IP name, registered with the Domain Name Service (DNS), that the location service daemon will use.

  Note: In a sysplex, you should consider using a virtual IP address (VIPA) for the location service daemon IP name. Review the WebSphere for z/OS Installation and Customization Guide for information about configuring this value.

Port
: The port number on which the location service daemon listens.

SSL Port
: The port number on which the location service daemon listens for SSL connections.

Note: Select the IP name and port number for the location service daemon carefully. You can choose any name you want, but, once chosen, it is difficult to change, even in the middle of customization.

Worksheet

4 Security Customization: This panel allows you to specify authentication and authorization options for your run-time resources. For more information about security and WebSphere for z/OS, see “Setting up security” on page 18.

Table 45. Security Customization (1 of 1)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Use EJBROLE class to control EJB method access	N	
Use OPERMCDS to control commands	N	
Use Kerberos over SSL	N	
Use SSL basic authorization	N	
Use SSL client certificates	N	
Use passtickets	N	
Use CLASS(APPL)	N	
Test certificate authority label	WAS TestCertAuth	
PassTicket Profile name	CBS390	(cannot change)
PassTicket KEYMASK value	(null)	

In the following, specifying “Y” (yes) tells the dialog to define the profile or enable an option in RACF. Specifying “N” (no) tells the dialog to not define the profile or enable the option.

Use EJBROLE class to control EJB method access
: Specify “Y” to create the RACF EJBROLE class. EJBROLE controls method accesses for enterprise beans.

Use OPERCMDS to control commands
: Specify “Y” to create the RACF OPERCMDS class. OPERCMDS controls the ability of an operator to start servers.

Use Kerberos over SSL
: Specify “Y” to create Kerberos security. With this option, SSL provides message security and authenticates the server to the client. Kerberos provides the ability for the server to authenticate the client.

Use SSL basic authorization
: Specify “Y” to create SSL basic authorization security. With this option, the server proves its identity by passing a digital certificate to the client. The client proves its identity by passing a user identity and password known by the target server.

Use SSL client certificates
: Specify “Y” to create SSL client certificate security. With this option, both the server and client pass digital certificates to prove their identities to each other.

Use passtickets
: Specify “Y” to use passtickets.

Use CLASS(APPL)
: If the APPL class is activated in your installation, specify “Y” to create a CBS390 profile in that APPL class to represent WebSphere for z/OS and permit the WebSphere for z/OS server identities to that profile. If your installation does not require use of the APPL class, specify “N”.

Test certificate authority label
: The dialog uses this label to create a test certificate authority certificate.

  Recommendation: Use this certificate for testing purposes only.

PassTicket Profile name
: Name of the PassTicket Profile

PassTicket KEYMASK value
: Specify any string of 16 hexadecimal characters as a mask for PassTickets.

Worksheet

3 Generate customization jobs for the Deployment Manager node

Table 46. Generate customization jobs

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Job card information	//jobname JOB (ACCTNO,ROOM),'*userid*',CLASS=A,REGION=0M //* //* //*	

Specify the job card according to your installation requirements.

Note: The dialog generates a job name and the ″JOB″ keyword for each job.

S Save customization variables for the Deployment Manager node

Table 47. Save customization variables

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Dsname	(null)	

Specify the name of the data set into which you want to save the customization variables.

Rules:

- The data set must be a sequential data set. Do not specify a member name.
- Place quotes around the data set name.

Worksheet

L Load customization variables for the Deployment Manager node

Table 48. Load customization variables

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Dsname	(null)	

Specify the data set from which you will prime the variables.

Following the customized Deployment Manager node instructions

The major stages of the Deployment Manager node customization process are:

Stage	Description
1	Make a variety of configuration changes to your z/OS system configuration (PARMLIB, TCP/IP, workload management, and so forth). The customized instructions provide details and pointers to relevant documentation.
2	Create a customized set of RACF commands for initial WebSphere for z/OS security setup through job BBODBRAJ. The RACF commands are saved in member BBODBRAK of the *hlq*.DATA data set. Job BBODBRAK executes these RACF commands. Later, you can use the RACF commands saved in BBODBRAK to help in defining security for additional servers or users.
3	Allocate and mount the WebSphere for z/OS run-time HFS through job BBOWCHFS. If your root HFS is mounted read-only, you may need to define one or more mount points manually. See the instructions for details. Job BBOMCFG creates subdirectories in the WebSphere for z/OS run-time HFS.
4	Configure the runtime HFS to include the directory for UDDIReg with the job BBOMCFGU, if desired.
5	Copy customized PROCLIB members into their proper locations using job BBODCPY1.
6	Copy customized HFS files into their proper locations using job BBODCPY2.
7	Create the was.env file for the location service daemon and the application servers through job BBODC2N.
8	Install the Administrative Console and the IVT application through job BBODIAPP.
9	Complete the HFS initialization with job BBODCFG2.
10	Activate Resource Recovery Services (RRS), if it is not already active, with the MVS command `START ATRRRS,SUB=MSTR`.
11	Start the Deployment Manager (and, hence, the location service daemon) with the MVS command `START BBO5DCR,JOBNAME=BBODMGR`. This process stores information in the WebSphere for z/OS databases and HFS. Once the process is complete, you have a working WebSphere for z/OS network deployment run-time that you will use to run your own applications.
12	Stop the Application Server with the MVS command `STOP BBOS001`.
13	Add the application server to the deployment manager's cell with job BBOWADDN.
14	Update the application server's WLM application environment according to the examples in the generated instructions.
15	Start the node agent server with the MVS command `START BBO5ACR,JOBNAME=BBON001` if desired.

16	Start the Application Server, if desired, with the following MVS command:
	`S <controller_procname>,JOBNAME=server_shortname,`
	`ENV=<cell_shortname.node_shortname.server_shortname>`
	
	Note: You must start the node agent server before you start the Application Server.

4 Configure Web Services — worksheets, definitions and instructions

Rule: You must configure your base Application Server node before you configure Web Services. See “1 Configure base Application Server node — worksheets, definitions and instructions” on page 57 for more information.

After following the steps in “Steps for starting the customization dialog” on page 49, choose option 4 and press Enter to configure Web Services. You will be presented with a screen with a long statement on it. Read the text then press Enter to continue.

Rule: You MUST turn off the PF keys display by issuing ″PFSHOW OFF″ in order to see the entire panel.

The next panel finishes the statment and asks if you wish to activate this Web Services Technology Preview (Y/N). Select ″Y″ if you wish to activate the Web Services Technology Preview. Otherwise, select ″N″ and you will go back to the main option panel. Selecting ″Y″ makes available to Web Services the WebSphere for z/OS runtime libraries that you designated when you configured your base server.

Result: If you select ″Y″, you see the following dialog panel:

```
-----------------     WebSphere for z/OS Customization      ------------------
Option  ===>                                                     Appl: BBO5

 Configure Web Services

   Use this dialog to define WebSphere for z/OS variables and generate
   customization jobs for your installation. Specify an option and press ENTER

   HLQ for WebSphere product data sets: BBO


   1  Allocate target data sets. The data sets will contain the WebSphere
      customization jobs and data generated by the dialog.

   2  Define variables. Define your installation-specific information for
      WebSphere customization.

   3  Generate customization jobs. Validate your customization variables
      and generate jobs and instructions.

   4  View instructions. View the generated customization instructions.


   Options for WebSphere for z/OS Customization Variables

   S  Save customization variables. Save your WebSphere customization
      variables in a data set for later use.

   L  Load customization variables. Load your WebSphere customization
      variables from a data set.
```

1 Allocate Target Data Sets for Web Services

Table 49. Allocate target data sets

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
High Level Qualifier	(null)	

Worksheet

This panel asks you to specify the high-level qualifiers (hlq) for the target data sets. Target data sets are those into which the customization dialog places the customized jobs and other data. The data sets are:

hlq.CNTL
: A partitioned data set of fixed block, 80-byte records, that contains WebSphere for z/OS customization jobs.

hlq.DATA
: A partitioned data set of variable length records that contains other data produced by the customization dialog.

Worksheet

2 Define variables for Web Services

Table 50. Define variables for Web Services (1 of 1)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
WebSphere SMP/E home directory	/usr/lpp/zWebSphere/V5R0M0	
Mount point	/WebSphere/V5R0M0	
Web Services Home directory	*	
Owner	ASSR1	(cannot change)
Group	WSCFG1	(cannot change)
SBBOEXEC	BBO.SBBOEXEC	

WebSphere SMP/E home directory
: The name of the directory where WebSphere for z/OS files reside after SMP/E installation.

Mount point
: Read/write HFS directory mount point where application data and environment files will be written. The customization process creates this mount point, if it didn't already exist.

Web Services Home directory
: The name of the directory under the mount point that corresponds to the Application Server on which you want to install the Web Services Technology Preview. The default is "*", which specifies all Application Servers under that mount point.

Owner and Group information

These are the values you initially set for your base system and are displayed for reference purposes only. If you wish to change them, you need to go back and reconfigure your base server.

Owner
: USS user ID assigned to the files that move into the Application Server's directories.

Group
: Group assigned to the files that move into the application server's directories.

SBBOEXEC
: SMP/E distribution PDS where WebSphere for z/OS is installed.

3 Generate customization jobs for Web Services

Table 51. Generate customization jobs

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Job card information	//jobname JOB (ACCTNO,ROOM),'*userid*',CLASS=A,REGION=0M //* //* //*	

Specify the job card according to your installation requirements.

Note: The dialog generates a job name and the ″JOB″ keyword for each job.

Worksheet

S Save customization variables for Web Services

Table 52. Save customization variables

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Dsname	(null)	

Specify the name of the data set into which you want to save the customization variables.

Rules:

- The data set must be a sequential data set. Do not specify a member name.
- Place quotes around the data set name.

L Load customization variables for Web Services

Table 53. Load customization variables

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Dsname	(null)	

Specify the data set from which you will prime the variables.

Following the customized Web Services instructions

The major stages of the Web Services customization process are:

Stage	Description
1	Make a variety of configuration changes to your z/OS system configuration. The customized instructions provide details and pointers to relevant documentation.
2	Configure the runtime HFS to include the directory for ″webservices″ with job BBOMCFGW.

Chapter supplement

This section provides a general reference for operations and jobs you might need during the installation.

Steps for cold-starting RRS

Perform the following steps to cold-start RRS:

1. Shut down WebSphere for z/OS (if running) and DB2 (if in use).

   _____

2. Shut down RRS using the `SETRRS CANCEL` command.

   _____

3. Delete and redefine the RRS resource manager data logstream (RM.DATA) using the same attributes you used to create it.

   Note: See member ATRCOLD in SYS1.SAMPLIB for a sample jobstream.

   _____

4. Start RRS using the `S ATRRRS,SUB=MSTR` command.

   _____

You know you are done when the job completes successfully.

Handling workload management and server failures

During operations, if your application fails repeatedly, causing the application servants to terminate, workload management may terminate the application environment for the application. WebSphere for z/OS issues the following message if it tries to use a failed application environment:

```
BBOU199E Unable to schedule work.  WLM application environment applenv has
         stopped.
```

You must fix the problem with your application, then restart the application environment with the RESUME option on the VARY WLM command.

Steps for checking and starting the workload management application environment

Perform these steps to check and start the workload management application environment:

1. To display the application environment, issue:

   ```
   d wlm,applenv=*
   ```

   for static application environments or

   ```
   d wlm,dynappl=*
   ```

   for dynamic application environments.

   _____

2. To start the application environment, issue:

   ```
   v wlm,applenv=environment_name,resume
   ```

   for static application environments or

   ```
   v wlm,dynappl=*
   ```

for dynamic application environments, where **environment_name** is the application environment name in either case.

Note: The dynamic application environment commands apply only if you are running z/OS V1.2 or above with the WLM-DAE support PTF (APAR OW54622). See the assembling applications information in the WebSphere Application Server InfoCenter, which you can access via the WebSphere Application Server for distributed platforms library Web page for more information.

You know you are done when a re-display of the application environment shows it is available.

Chapter 4. Performing post-installation tasks

This chapter covers topics and tasks that can occur after you have installed WebSphere for z/OS. Topics include:

- Guidelines for backing up your system
- Product service
- Setting up RACF protection for DB2
- Setting up automation and automatic restart management

Guidelines for backup of the WebSphere for z/OS system

Use the following guidelines to back up parts of your WebSphere for z/OS system:

1. Be sure to back up the RMDATA log for RRS. Otherwise, a failure could force you to do a cold start of RRS.
2. Set the ARCHIVE log retention period to one day.
3. Incorporate the following in your normal backup procedures:
   - WebSphere for z/OS proclibs
   - WebSphere for z/OS loadlibs
   - The directory where WebSphere for z/OS run-time information is written (the default is `/WebSphere/V5R0M0`).
4. Back up your own application executables, databases, and bindings.
5. If you wish to back up a single server, you can use the export/import function in the Administrative Console. For details on how to do this, see the assembling applications information in the WebSphere Application Server InfoCenter, which you can access via the WebSphere Application Server for distributed platforms library Web page.

Overview of product service

Contact the IBM Software Support Center for information about preventive service planning (PSP) upgrades for WebSphere for z/OS. For more information about PSP upgrades, see *WebSphere Application Server for z/OS V5.0: Program Directory*, GI11-2825. Although the *Program Directory* contains a list of required PTFs, the most current information is available from the IBM Software Support Center.

When applying service to WebSphere for z/OS, make copies of the product data sets and HFS, and apply maintenance to the copies. When you are ready to put the maintenance into production, the process is:

Stage	Description
1	Stop the application servers and the WebSphere for z/OS location service daemon.
2	Switch to the newly-serviced WebSphere for z/OS product data sets. You can do this by: • Renaming the new data sets to replace the old ones • Re-cataloging product data sets, if the names are identical, or • Changing WebSphere for z/OS cataloged procedures to refer explicitly to the new data sets. Make sure the MVS link list and APF list refer to the newly-serviced data sets.

Stage	Description
3	If the WebSphere for z/OS run-time is loaded into the link pack area, delete the old modules and load the new ones, or IPL the system to load the new modules into the LPA.
4	Verify that the newly-serviced HFS data sets are correctly mounted.
5	Perform any other migration actions (such as DB2 binds) as instructed in PTF or APAR cover letters.
6	Start the location service daemon and application servers.

Setting up RACF protection for DB2

You can use the RACF DSNR resource class to protect DB2 resources. This helps you centralize security management. This section gives you pointers to general information about setting up RACF protection for DB2 and specific information about the resources, groups, user IDs, and permissions used by WebSphere for z/OS.

There are three functional areas in RACF to consider regarding protection for DB2:

- The RACF DSNR class controls access to the DB2 subsystems. If the DSNR class is active, then WebSphere for z/OS controllers and servants need access to the *db2_ssn*.RRSAF profiles, where *db2_ssn* is your DB2 subsystem name. If a controller or servant does not have access, then that region will not initialize.
- DB2 identification and signon exits (DSN3@ATH and DSN3@SGN) assign authorization IDs. If you want to use secondary authorization IDs (RACF group names), then you must replace the default exits with these two sample routines. For details on how to install these sample routines, see *DB2 Administration Guide*, SC26-9931.
- WebSphere for z/OS does not support the protection of DB2 objects through the DSNX@XAC exit. To protect DB2 objects, you must use GRANT statements.

Steps for defining DB2 authorizations in RACF

Before you begin: You must complete general tasks for enabling RACF protection for your DB2 system. This includes adding entries to the RACF router table, installing identification and signon exits, and defining RACF user IDs for DB2 started tasks. You must also have your copy of the BBOCBRAJ sample provided with WebSphere for z/OS.

Perform the following steps to define DB2 resources and authorizations in RACF:

1. Remove the comment marks that surround the REXX and RACF commands. As shipped, the DSNR profile section is commented out.

   _____

2. Copy the BBOCBRAJ job to a new file.

   _____

3. Submit the job from a user ID with RACF SPECIAL authority.

   _____

You know you are done when the job completes successfully.

Setting up automation and automatic restart management

This section discusses recommendations for automation. See "Restarting WebSphere for z/OS" on page 135 for the steps for setting up automatic restart management and rules and restrictions for changing the automatic restart management policies.

Recommendation for automation for WebSphere for z/OS and its applications

You need to decide whether to start WebSphere for z/OS servers automatically at system IPL and implement this decision in your system automation. The automation policies should initialize WebSphere for z/OS and associated functions in the correct order, which is:

1. System Logger
2. RRS
3. DB2
4. TCP/IP
5. The location service daemon, which automatically starts the Deployment Manager and node agent
6. Your business application servers

For more information about automating WebSphere for z/OS servers, see *WebSphere Application Server for z/OS V5.0: Operations and Administration*, SA22-7912.

Chapter 5. Performing advanced tasks

This section covers advanced tasks, such as sysplex setup, advanced TCP/IP setup, and procedural application adapter setup.

Setting up WebSphere for z/OS on multiple systems in a sysplex

Once you have installed the WebSphere for z/OS run-time and associated business application servers on a monoplex, you can migrate the run-time and associated application servers to a sysplex configuration. The benefits of migrating to a sysplex include:

- You can balance the workload across multiple systems, thus providing better performance management for your applications.
- As your workload grows, you can add new systems to meet demand, thus providing a scalable solution to your processing needs.
- By replicating the run-time and associated business application servers, you provide the necessary system redundancy to assure availability for your users. Thus, in the event of a failure on one system, you have other systems available for work.
- You can upgrade WebSphere for z/OS from one release or service level to another without interrupting service to your users.

The following table shows the subtasks and associated procedures for enabling WebSphere for z/OS in a sysplex.

Subtask	Associated procedure (See . . .)
Setting up a sysplex	*z/OS MVS Setting Up a Sysplex*, SA22-7625
Making decisions about the WebSphere for z/OS configuration and sysplexes	"Steps for planning WebSphere for z/OS and cells" on page 128
Preparing your security system	"Steps for preparing your security system on a sysplex" on page 129
Setting up data sharing	*DB2 Data Sharing: Planning and Administration*, SC26-9935
Customizing base z/OS functions on the other systems in the sysplex	"Steps for customizing base z/OS functions on the other systems in the sysplex" on page 130
Making changes to TCP/IP	"Steps for making changes to TCP/IP" on page 132
Defining new WebSphere for z/OS clustered host instances in the sysplex	"Defining new WebSphere for z/OS systems in a sysplex" on page 133
Refreshing the WebSphere for z/OS systems	"Steps for restarting WebSphere for z/OS on another system in the cell" on page 133
Checking your configuration with the Installation Verification Test	"Running the Installation Verification Test (IVT) after initial customization" on page 134

WebSphere for z/OS and the sysplex

Before you perform the procedures in this chapter, it is important for you to understand the following topics:

- Setting up your sysplex for a rolling upgrade

Overview of setting up your sysplex for a rolling upgrade

Figure 8 shows the structure of a conventional sysplex HFS.

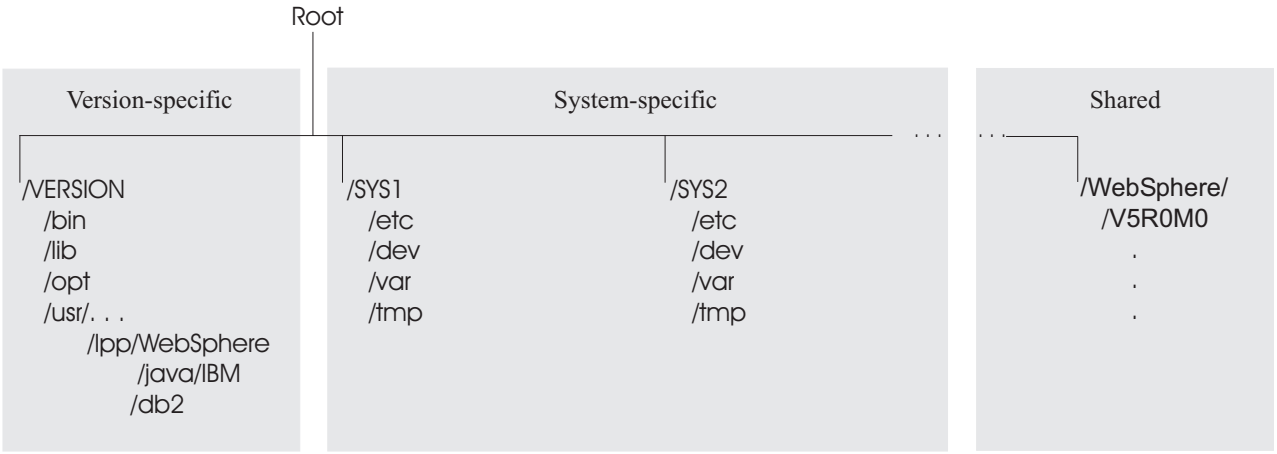


Figure 8. Conventional sysplex HFS structure

Utilizing cells in WebSphere for z/OS

Once you have installed WebSphere for z/OS on a monoplex or on a single system in a sysplex, you can enable it on a cell. This topic covers planning and building steps for your cell deployment.

Steps for planning WebSphere for z/OS and cells

Before you begin: You should have completed the WebSphere for z/OS installation and customization on a monoplex or on a single system in a sysplex. Also, you must have enabled a z/OS sysplex. For more information on sysplexes, see *z/OS MVS Setting Up a Sysplex*, SA22-7625.

Follow these steps to plan WebSphere for z/OS and cells:

1. Decide whether you want a single-system view of the error log. If you want a single-system view of the error log, and initially you set up the error log in the system logger and used DASD for logging, you must now configure the error log in the coupling facility.

   ---

2. Decide how you will share application executables in the cell. For tips and recommendations, see the assembling applications information in the WebSphere Application Server InfoCenter, which you can access via the WebSphere Application Server for distributed platforms library Web page.

   ---

3. Set up ARM. This release does not support cross-system restart, so you must set up your ARM policy accordingly. Make sure you specify TARGET_SYSTEM for the system on which each element runs (if you take the default TARGET_SYSTEM=*, you get cross-system restart).

   ---

4. Decide whether you will run all the WebSphere for z/OS run-time servers on every system in the cell.

   Recommendations: The following table provides recommendations and requirements for running servers in a cell.

Table 54. Running servers in a cell

Server	Recommendations and requirements for running servers in a cell
location service daemon and node agent	• You must run both these servers on each system in the cell in which you wish WebSphere for z/OS work to run. Thus, you may have some systems in your cell that do not run WebSphere for z/OS or WebSphere for z/OS applications at all. But, for those systems on which you want WebSphere for z/OS applications to run, you must have the location service daemon and node agent. • If a server indicates that PassTickets are desirable for interaction with a client, you must run the location service daemon and node agent on the system where the z/OS client resides.
Deployment Manager	Make sure you follow the correct steps to configure a deployment manager cell. See "Steps for building WebSphere for z/OS Deployment Manager cells" for more information.

Steps for building WebSphere for z/OS Deployment Manager cells

Before you begin: Follow the steps in "Steps for planning WebSphere for z/OS and cells" on page 128.

Follow these steps to build WebSphere for z/OS Deployment Manager cells:

1. Install the base server on each node in your sysplex.
2. Install a Deployment Manager cell on one node in your sysplex.
3. Add base server nodes to the Deployment Manager cell.

Steps for preparing your security system on a sysplex

Before you begin: Read the background information about security in "Setting up security" on page 18.

Follow these steps to prepare your security system:

1. When you place WebSphere for z/OS on several systems in the sysplex, you must implement a shared RACF database. WebSphere for z/OS assumes that a user ID represents the same user identity on all systems in the cell.
2. Define each like-configured controller and servant to have the same authorizations throughout the cell. You can accomplish this by using generic RACF profiles in the STARTED class and authorizing common user IDs to those profiles. For example, a BBOC001* profile would cover all start procedures for the BBOC001 servers.

3. Define authentitcation mechanisms for cell interactions. The choices you have are:
   - PassTickets
   - Asserted identities
   - Kerberos

   For an example of how to make your choice, see "Example of choosing system security" on page 29.

You are done with setting up security on the cell.

Steps for customizing base z/OS functions on the other systems in the sysplex

Repeat the same customizations to base z/OS functions that you did for your initial installation and customization of WebSphere for z/OS. The steps are repeated here for convenience.

Note: The following steps assume that your default WebSphere for z/OS data set high-level qualifier (hlq) is "BBO". If it is not, modify the examples to use your specified hlq.

Before you begin: You must have the WebSphere for z/OS product code installed through SMP/E and have created copies of the product sample files.

Perform the following steps to change the base system:

1. Change SCHEDxx to include the statements from the BBOSCHED sample file you ran in the customization dialog.

2. APF-authorize the BBO.SBBOLOAD, BBO.SBBOLD2, and BBO.SBBOLPA data sets.

   Example: Your PROGxx PARMLIB member could include:

```
APF FORMAT(DYNAMIC)
/*********************************************************************/
/* BOSS LOCAL DATASETS                                               */
/*********************************************************************/
APF ADD
    DSNAME(BBO.SBBOLOAD)
    VOLUME(vvvvvv)
APF ADD
    DSNAME(BBO.SBBOLD2)
    VOLUME(vvvvvv)
APF ADD
    DSNAME(BBO.SBBOLPA)
    VOLUME(vvvvvv)
```

   where vvvvvv is your volume identifier.

3. Ensure that the Language Environment data set, SCEERUN, and the DB2 data set, SDSNLOAD, are authorized.

4. Do **not** APF-authorize BBO.SBBOULIB or BBO.SBBOMIG, because they should run under the authority of the client user.

5. Use the following table to place WebSphere for z/OS modules:

Table 55. Placing modules in LPA or link list

Modules	Notes
BBO.SBBOLPA	Load all members into the LPA.
BBO.SBBOLOAD	We recommend you dynamically load all members into the LPA. If your virtual storage is constrained, place the members in the link list.
BBO.SBBOMIG	You can put members into the link list or LPA.
BBO.SBBOLD2	Do **not** put members from SBBOLD2 in the LPA. Place these members in the link list.
BBO.SBBOULIB	Do **not** place these members in **either** the LPA or link list.

Rule: These data sets are PDSEs and cannot be added to members in LPALSTxx or IEALPAxx.

Recommendation: For automation, if you want to ensure WebSphere for z/OS modules are loaded into dynamic LPA and available after an IPL, create a new PROGxx member with the SETPROG LPA commands and invoke the PROGxx member from PARMLIB COMMNDxx.

Example:

```
SETPROG LPA,ADD,MASK=*,DSNAME=BBO.SBBOLOAD
SETPROG LPA,ADD,MASK=*,DSNAME=BBO.SBBOLPA
```

Notes:

a. Change "BBO" if it is not the high-level qualifier for your WebSphere for z/OS data sets.

b. If using SETPROG on a running system, be sure to purge modules with the same name as those from BBO.SBBOLPA, BBO.SBBOLOAD, or BBO.SBBOMIG that are already in the LPA.

Attention: Be sure that the size of your LPA can hold the WebSphere for z/OS modules. See "Recommendations for using memory" on page 35.

6. If you used a PROGxx file for APF authorizations or the LPA, be sure to issue:

   ```
   SET PROG=xx
   ```

   where xx is the suffix on your PROGxx member.

7. Make sure all the BBO.* data sets are cataloged. While not required, this is highly recommended.

8. Update your SYS1.PARMLIB(BLSCUSER) member with the IPCS models supplied by member BBOIPCSP. For details in BLSCUSER, see *z/OS MVS IPCS User's Guide*, SA22-7596.

9. If you want to start SMF recording to collect system and job-related information on the WebSphere for z/OS system:

a. Edit the SMFPRMxx parmlib member.
   1) Insert an 'ACTIVE' statement to indicate SMF recording.
   2) Insert a SYS statement to indicate the types of SMF records you want the system to create.

      Example: Use SYS(TYPE(120:120)) to select type 120 records only. Keep the number of selected record types small, to minimize the performance impact.

b. To start writing records to DASD, issue the following command:

```
t smf=xx
```

   Where xx is the suffix of the SMF parmlib member (SMFPRMxx). For more information about the SMF parmlib member, see *z/OS MVS System Management Facilities (SMF)*, SA22-7630.

   When you activate writing to DASD, the data is recorded in a data set (specified in SMFPRMxx).

Note: Later, when you have installed the Administrative Console, you will enable the server to collect SMF records by defining properties on the server properties form. For more information about WebSphere for z/OS and its use of SMF recording, see *WebSphere Application Server for z/OS V5.0: Operations and Administration*, SA22-7912.

Steps for making changes to TCP/IP

Before you begin: You must have TCP/IP installed and configured.

Perform the following steps to make changes to TCP/IP

1. Change DNS entries. Assuming you use an implementation of the DNS that allows use of generic IP names that dynamically resolve to like-configured servers, you must adjust the IP names in your DNS. Keep the generic IP name of the location service daemon, but add a new IP name for the second and subsequent location service daemon servers. This is important not only for workload balancing, but in the event of a server failure: the DNS can direct work to other servers.

   For more information, see "Connection optimization" on page 138 and "IBM Network Dispatcher" on page 139.

---

2. In the TCP/IP profile for each additional system in the cell, add a port for the location service daemon and associate it with a new location service daemon server name. By default, WebSphere for z/OS uses port 5655 for the location service daemon. Also, WebSphere for z/OS names the first location service daemon server DAEMON01 and increments the suffix on that name for each new location service daemon server (DAEMON02, DAEMON03, and so forth). Thus, on your second system in the cell, add a port and associate it with DAEMON02.

   Example:

```
5655   TCP     DAEMON02
```

   Follow the same pattern for your third and subsequent systems in the cell.

You should now have completed your TCP/IP updates.

Defining new WebSphere for z/OS systems in a sysplex

Use the Administrative Console to define additional systems in the sysplex with their servers. We assume you have already created the first WebSphere for z/OS system with an application server called BBOASR2, BBOC001, or both (the application servers used for the Installation Verification Test).

We provide instructions for defining the second system. Follow the same pattern of steps for the third and subsequent systems.

Steps for defining the second WebSphere for z/OS system

This procedure explains how to use the Administrative Console to create a second WebSphere for z/OS run-time system.

Before you begin: You must have your initial WebSphere for z/OS system installed and running. If not, start RRS.

Follow these steps to define the second WebSphere for z/OS system:

1. Log onto the Administrative Console.

   _____

2. Define a second system in the sysplex. The run-time servers are defined automatically for you.

   _____

3. Check the WebSphere variables for each run-time sever instance on the second system. The WebSphere variables are defined hierarchically in the following order: sysplex, server, node, then cluster. A WebSphere variable lower in the hierarchy overrides a matching one higher in the hierarchy. Check the WebSphere variables for the following servers. Some WebSphere variables are common for all systems in the sysplex, while others are unique for each system.

   _____

4. Specify start procedures to be used by the location service daemon to start the node agent and Deployment Manager servers (controllers) on the second system. After you start the location service daemon, it starts these server controllers automatically.

   _____

You have defined the new WebSphere for z/OS run-time.

Steps for restarting WebSphere for z/OS on another system in the cell

Note: See “Restarting WebSphere for z/OS” on page 135 for more cell restart options.

Before you begin: You must complete all previous procedures in this section.

Follow these steps to cancel and restart WebSphere for z/OS on the second system:

1. Restart WebSphere for z/OS on the second system:

   ```
   S BBODMN.DAEMON02,SRVNAME='DAEMON02'
   ```

   _____

2. Start each Application Server on the second system.

You are done when all servers initialize on the second system.

Running the Installation Verification Test (IVT) after initial customization

If you want to run the Installation Verification Test at a time other than during initial customization, you can follow either of two methods:

- Use the BBOWIVT job
- Run `ivt.sh` from a command line

Steps for running the Installation Verification Test with a job

Before you begin:

- You must complete all procedures in "Setting up WebSphere for z/OS on multiple systems in a sysplex" on page 127.
- You must have your copy of the BBOWIVT client job.

Follow these steps to run the Installation Verification Test using the BBOWIVT job:

1. Run BBOWIVT on the new system you have defined.

---

2. Cancel the local server1 Java server and run the corresponding client job locally, forcing the work to move to a server on another system in the cell.

   Example: Cancel the server2 server on the second system. Leave server1 running on the first system. Use the Administrative Console or the CANCEL command:

   ```
   c server1.server2
   ```

   Submit BBOWIVT on the second system.

You are done when the Installation Verification Test runs successfully.

Steps for running the Installation Verification Test from a command line

Follow these steps to run the Installation Verification Test from a command line:

1. From a command line, navigate to the /WebSphere/V5R0M0/AppServer/bin/ directory.

---

2. Issue the following execution:

   ```
   ivt.sh [-p port_number]
   ```

   where *-p port_number* is an optional argument that specifies your port number. If you do not specify a port number, the program will use the default port number value of 9080.

   Example:

   ```
   /WebSphere/V5R0M0/AppServer/bin> ivt.sh -p 9090
   ```

You are done when the Installation Verification Test runs successfully.

Restarting WebSphere for z/OS

This section describes various methods for restarting your system when you are running in a cell environment. See "Setting up WebSphere for z/OS on multiple systems in a sysplex" on page 127 to set up WebSphere for z/OS to run in a cell environment.

Setting up automatic restart management

If you have an application that is critical for your business, you need facilities to manage failures. z/OS provides rich automation interfaces that you can use to detect and recover from failures, but there are some recovery situations that are too specialized to handle with automation. For such situations, z/OS provides automatic restart management, which handles the restarting of servers when failures occur. WebSphere for z/OS uses automatic restart management.

Each WebSphere for z/OS server (including servers you create for your business applications) automatically registers with the automatic restart management default group. Each registration uses a special element type called SYSCB, which automatic restart management treats as restart level 3, assuring that RRS and DB2 restart before any server.

Peer restart and recovery

Note: For a full overview of peer restart and recovery and more information, see *WebSphere Application Server for z/OS V5.0: Operations and Administration*, SA22-7912.

If a failure occurs, automatic restart management can restart WebSphere for z/OS and related servers on the same system or on an alternate system in the cell. The latter condition is achieved through peer restart and recovery, which restarts the controller on another system and goes through the transaction restart and recovery process so that we can assign outcomes to transactions that were in progress at the time of failure. During this transaction restart and recovery process, data might be temporarily inaccessible until the recovery process is complete. The restart and recovery process does not result in lost data.

Resource managers (such as DB2) that were being accessed at the time of failure may hold locks that are scoped to a transaction UR (unit of recovery). Once an outcome has been assigned to a UR, the resource managers will, generally, drop those locks.

Rule: Make sure **every** system (your original system as well as any systems intended for recovery) has the following installed:

- z/OS v1.2
- WebSphere Application Server for z/OS V5

Note: The following products individually support peer restart and recovery, providing the above prerequisites are all properly installed:

- IMS V8
- CICS Transaction Server V1.3

- MQSeries 5.3.1

The products mentioned above may not work in conjunction with other subsystems in the same transaction.

To allow WebSphere for z/OS to restart on an alternate system, the **prerequisites must be met** on every participating system in the cell **before** reconfiguring the ARM policies to enable peer restart and recovery. Installing the SPE on all your systems will not hinder your current running atmosphere if you want to continue to only restart in place. If this is not done, there is a possibility that the controller will not be able to move back—OTS will attempt to restart on the alternate system and fail. If there are any URs that are unresolved with RRS once this happens, the controller will not be allowed to restart on the home system until RRS is cancelled on the alternate system. For more information on OTS and RRS, see *z/OS MVS Programming: Resource Recovery*, SA22-7616.

Note: If you do not plan to use peer restart, you do not need to abide by these functional prerequisites. Your system will instead use the restart in place function that already exists.

Prior to peer restart, you must ensure that the location service daemon and node are already running on ALL the systems in the cell so that the recovering servers can collect the appropriate configuration information. If a system in the cell is not running the location service daemon and node, then this system must be ARM disabled. Otherwise, the recovering system might attempt to recover on the system not runnning the location service daemon and node. In this case, the recovery will fail and the workload manager will issue a cell-wide stop for the workload, therefore causing a cell-wide outage.

You must be running in a cell with a WebSphere for z/OS datasharing configuration to utilize peer restart and recovery. You cannot restart on a system that is not in datasharing with you, and you cannot restart out of place at all if you are not in datasharing with any other system. If you don't run in datasharing, your configuration is not known to the recovery system, and it cannot properly execute the restart and recovery. Your only option in that case, and in the case where you are not in a cell at all, would be to restart in place. See *z/OS MVS Setting Up a Sysplex*, SA22-7625 for instructions on setting up a cell.

Activating automatic restart management

Though servers automatically register with automatic restart management, you must activate the arm component itself, which means you must:

1. Allocate an ARM couple data set
2. Start the automatic restart management policy

If automatic restart management is not active, WebSphere for z/OS issues an error message to the hardcopy log.

You should also consider modifying the default automatic restart management policies for WebSphere for z/OS servers. It is not necessary to modify the policies to get started with WebSphere for z/OS, but you should consider doing so when you move your applications into production. For complete information about how to modify the policies, see *z/OS MVS Setting Up a Sysplex*, SA22-7625.

Steps for activating automatic restart management: The following procedure is intended to give you enough information to get automatic restart management running. Defining automatic restart management policies is beyond the scope of

this manual. For general information about defining automatic restart management policies, see *z/OS MVS Setting Up a Sysplex*, SA22-7625.

Before you begin: You must have access to the couple data set format utility, IXCL1DSU, in SYS1.MIGLIB. If you plan to modify the automatic restart management policy, you must have access to the administrative data utility, IXCMIAPU, also in SYS1.MIGLIB, and have UPDATE authorization to the RACF FACILITY class MVSADMIN.XCF.ARM. To start a policy, you must have READ authorization to the RACF FACILITY class MVSADMIN.XCF.ARM.

Follow these steps to activate automatic restart management for WebSphere for z/OS:

1. If you have not already formatted a couple data set for policies, do so now. For details, see *z/OS MVS Setting Up a Sysplex*, SA22-7625.

   _____

2. Submit the job to format the ARM couple data set.

   _____

3. If you do not want to modify the automatic restart management policy at this time, skip to the next step. To get started, you do not need to modify the policy.

   If you do want to modify the automatic restart management policy, go to *z/OS MVS Setting Up a Sysplex*, SA22-7625, and follow the instructions in that manual.

   _____

4. Issue the following operator commands to start the automatic restart management policy:

   ```
   SETXCF COUPLE,TYPE=ARM,PCOUPLE=(dsname,vvvvvv)
   SETXCF START,POLICY,TYPE=ARM
   ```

   where

   dsname
   Is the data set name for the couple data set.

   vvvvvv
   Is the volume serial of the volume on which the couple data set resides.

   _____

You are done when the SETXCF commands complete successfully.

Implementing an advanced TCP/IP network

This topic describes advanced TCP/IP configurations, including:

- TCP/IP sysplex functionality with Sysplex Distributor
- The use of multiple TCP/IP stacks on z/OS
- Connection optimization, a z/OS function by which workload management and the DNS cooperate to route requests
- The IBM Network Dispatcher, which is a network router
- Bind-specific support, which allows you to control the use of TCP/IP resources in WebSphere for z/OS

Sysplex Distributor

The IBM-recommended implementation if you are running in a sysplex is to set up your TCP/IP network with Sysplex Distributor. This makes use of dynamic virtual IP addresses (DVIPAs), which increase availability and aid in workload balancing.

The following are recommended environment considerations for Sysplex Distributor:

- You need only basic sysplex functionality to utilize DVIPAs and Sysplex Distributor because these functions do not rely on data stored permanently in the coupling facility.
- Set up your system such that each HTTP request connection results in no saved state or the HTTP and Application Servers are configured to share a persistent state.

When going this route, HTTP server plug-ins send no-affinity connections to Sysplex Distributor (a secondary connection load balancer) with more information to make a better distribution decision.

Note: As long as the HTTP catcher itself is not bound to any particular IP address, the application-specific DVIPA can be used when affinities dictate a particular server. This allows use of the Sysplex Distributor server address for requests that are not tied to a server, covering the same set of servers in the sysplex.

Since the client connection terminates at the plug-in/proxy and the secondary connection is established by the plug-in itself, there is no need for network address translation.

Requests to the node agent do not require any affinity, and each request is independent of other requests. Sysplex Distributor can be used to balance work requests among node agents, with the added benefit that Sysplex Distributor knows which nodes are available. Therefore, it will never route a work request to a node that is not listening for new connection requests.

Note: If you are running z/OS 1.2 or earlier, Sysplex Distributor is limited to distribution on only four ports for a particular distributed DVIPA. You may configure multiple DVIPAs when more than four ports exist, but this is a configuration burden.

Multiple TCP/IP stacks

You may want to run multiple TCP/IP stacks on the same system to reduce the chances of having a single point of failure. For instance, you may have multiple OSA Features connecting your System/390 to the network and want to assign a TCP/IP stack to each one; to do so, use the common INET physical file system (C_INET PFS). This physical file system allows multiple physical file systems (network sockets) to be configured and active concurrently.

Specify common INET through the NETWORK DOMAINNAME parameter of SYS1.PARMLIB(BPXPRMxx). See *z/OS UNIX System Services Planning*, GA22-7800, and *z/OS Communications Server: IP Configuration Reference*, SC31-8776, for details.

Connection optimization

Figure 9 on page 139 shows a configuration in which the Domain Name Server cooperates with workload management (WLM) to route client requests throughout a cell. Characteristics of this configuration are:

- The domain name server (DNS) is replicated by setting up a secondary DNS on more than one system in the cell.
- The client needs to know the location service daemon IP Name in order to connect to WebSphere for z/OS.
- Each system in the cell has the same location service daemon IP Name and Resolve IP Name. Workload management and the domain name server determine the actual system to which client requests go. The client sees the cell as a single system, though its requests may be balanced across systems in the cell.
- As part of workload balancing and maximizing performance goals, workload management also routes work requests to systems in the cell. This function is possible because WebSphere for z/OS cooperates with workload management (see "Workload management and WebSphere for z/OS" on page 201 for details). Because the system references that a client sees are indirect, even requests from that same client may be answered by differing systems in the cell.
- The implication for clients is that they should not cache IP addresses unless they can recover from failed connections. That is, if a connection fails, a client should be able to reissue a request, but, because the IP address is an indirect address, a reissue of the request can be answered by another system in the cell.

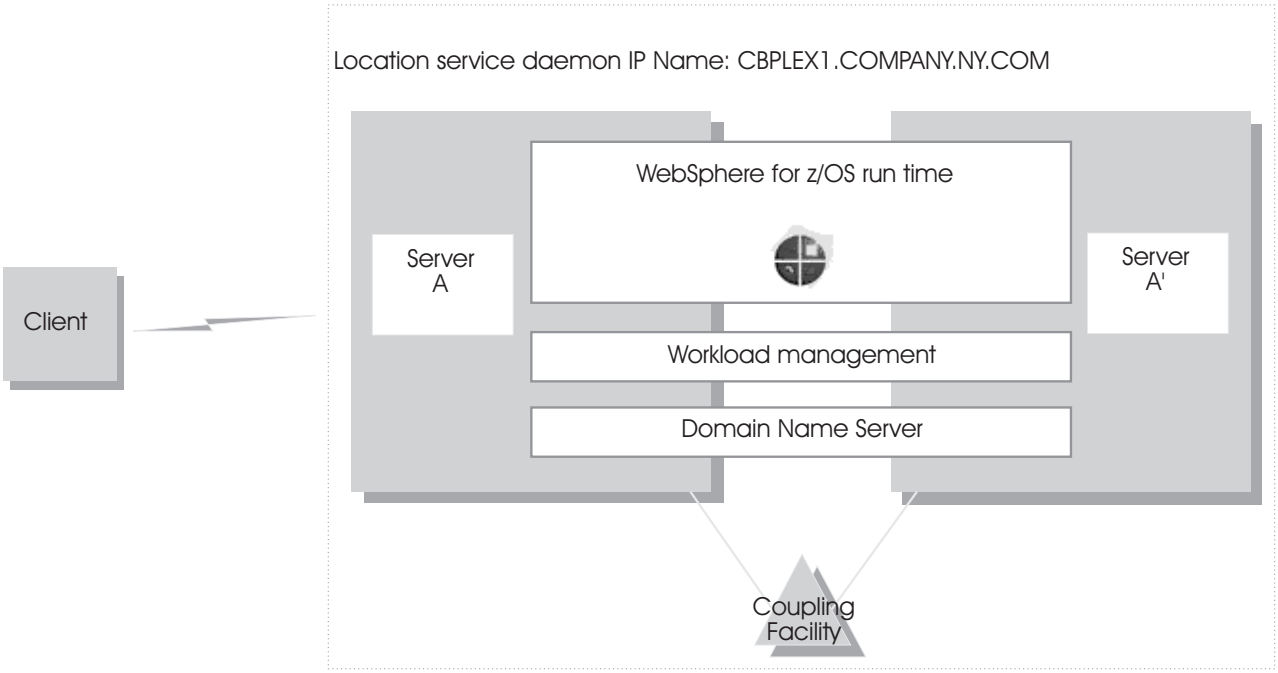


Figure 9. Connection optimization configuration

For details on setting up servers for connection optimization, see *z/OS Communications Server: IP Configuration Reference*, SC31-8776.

IBM Network Dispatcher

The IBM Network Dispatcher (see Figure 10 on page 140) is a router that handles network requests for the cell. Characteristics of such a configuration are:

- The location service daemon IP Name is associated with the IP address of the router.

- The IBM Network Dispatcher cooperates with workload management to route requests through the cell. The client never sees a change in IP addresses.
- The implication for clients is that they can cache the IP addresses, because this configuration does not change them dynamically.

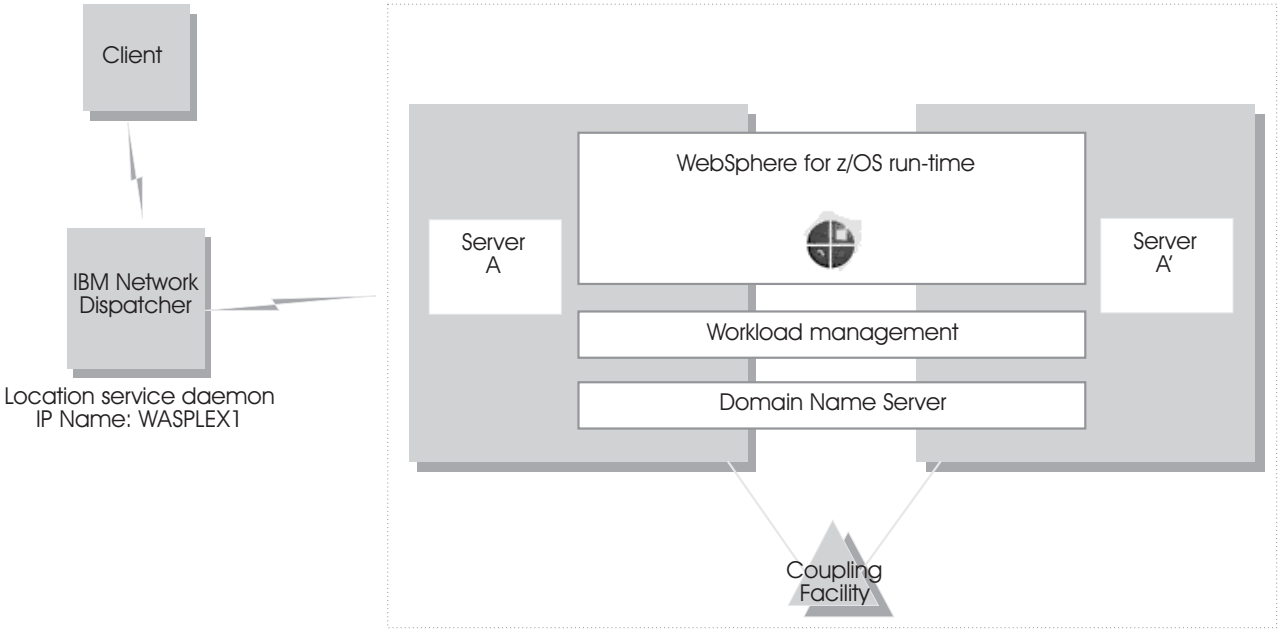


Figure 10. IBM Network Dispatcher configuration

Bind-specific support in WebSphere for z/OS

Bind-specific support in WebSphere for z/OS allows you to control the use of TCP/IP resources in WebSphere for z/OS. This support allows you to have the WebSphere for z/OS ORB and other products and applications on the same z/OS system without requiring the client code to configure unique ports. In other words, this support allows use of port 2809 by WebSphere for z/OS and other products and applications on the same system. This support allows the utilization of multiple TCP/IP stacks (Common INET) by the WebSphere for z/OS ORB and the use of multiple IP addresses on the same TCP/IP stack.

To use bind-specific support, use the SRVIPADDR WebSphere variable, which specifies the IP address in dotted decimal format. WebSphere for z/OS servers listen for client connection requests on this IP address.

Because a given IP address is associated with a given TCP/IP stack, you could specify the SRVIPADDR variable in the environment file so that a WebSphere for z/OS server uses a specific TCP/IP stack.

In addition, because you can define multiple IP addresses for a given TCP/IP stack, WebSphere for z/OS port 2809 servers could share the same TCP/IP stack with other products and applications requiring port 2809, because you made their IP addresses unique with SRVIPADDR.

Alternatively, you can, without the use of bind-specific support, define alternate ports for port 2809 and the location service daemon, which are the only values defined by the CORBA standard. However it is not clear that all client ORBs will

easily support configuring the WebSphere for z/OS port to something other than 2809. Configure the ports for the location service daemon and node by specifying port numbers on the DAEMON_PORT and RESOLVE_PORT WebSphere variables.

For details on WebSphere variables, see the Administrative Console or the InfoCenter.

For more information about multiple TCP/IP stacks (Common INET), see *z/OS UNIX System Services Planning*, GA22-7800. For more information about multiple IP addresses on the same TCP/IP stack, see *z/OS Communications Server: IP Configuration Reference*, SC31-8776.

Implementing advanced security

This topic covers advanced security issues:

- Selecting a user registry
- Selecting an authentication mechanism
- Enabling global security
- Configuring the authentication protocol
- How clients and clusters negotiate security protocols
- Setting up SSL security
- Setting up the asserted identity function
- Setting up the Web container security collaborator
- Setting up Kerberos security

Selecting a user registry

Information about users and groups reside in a user registry. In WebSphere Application Server, a user registry authenticates a user and retrieves information about users and groups to perform security-related functions, including authentication and authorization. Implementation is provided to support multiple operating system or operating environment-based user registries (z/OS SAF registry) and most of the major Lightweight Directory Access Protocol (LDAP)-based user registries. You can use the custom LDAP feature to support any LDAP server by setting up the correct configuration (user and group filters). However, support is not extended to these custom LDAP servers since there are many possibilities that cannot be tested.

In addition to Local OS and LDAP registries, WebSphere Application Server also provides a plug-in to support any registry by using the custom registry feature (also referred as custom user registry). The custom registry feature supports any user registry that is not implemented by WebSphere Application Server. The possibilities are endless in that you can make any registry to work in the product environment by implementing an interface called the UserRegistry interface. This interface is very helpful in situations where the current user and group information exists in some other formats (for example, a database) and cannot move to Local OS or LDAP. In such a case, implement the UserRegistry interface so that WebSphere Application Server can use the existing registry for all the security-related operations. Implementing a custom registry is a software implementation effort and it is expected that the implementation does not depend on other WebSphere Application Server resources, for example, data sources, for its operation.

Before configuring the user registry, decide which registry to use. The choices of user registry include:

- Local OS — SAF-based
- LDAP
- Custom user registry

Though different types of registries are supported, only a single active user registry can be configured at once. All the processes in WebSphere Application Server can use one active registry. Configuring the correct registry is a prerequiste to assigning users and groups to roles for applications. By default, when no registry is configured the Local OS SAF-based registry is used. So if your choice of registry is not Local OS you need to first configure the registry, which is normally done as part of enabling global security, restart the servers, and then assign users and groups to roles for all your applications.

Step for selecting Local OS user registry

Before you begin: Before configuring the Local OS registry you need to know the user name (ID) and password that will be used here. This user can be any valid user in the registry. This user will be referred to as either a product security server ID, a server ID or a server user ID in the documentation. Having a server ID means that a user has special privileges when calling protected internal methods.

You need to start the Administrative Console by specifying URL: http://<server_hostname>:9090/admin.

Perform the following steps to select the Local OS user registry.

1. Click **Security** > **User Registry** > **Local OS** in the Navigation tree on the left.

   _____

2. On the Local OS registry panel in the General Properties section of the Configuration tab, enter the Server user ID and password.

   This ID is the security server ID, which is only used for WebSphere Application Server security and is not associated with the system process that runs the server. The server calls the Local OS registry to authenticate and obtain privilege information about users by calling the native APIs in that particular registry.

   _____

3. Click OK.

   _____

You know you are done when...

Steps for selecting LDAP user registry

Before you begin: To use LDAP as the user registry, you need to know a valid user name (ID), the user password, the server host and port, the base distinguished name (DN) and if necessary the bind DN and the bind password. You can choose any valid user in the registry that is searchable. In some LDAP servers, the administrative users are not searchable and cannot be used (for example, cn=root in SecureWay). This user is referred to as WebSphere Application Server security server ID, server ID, or server user ID in the documentation. Being a server ID means a user has special privileges when calling some protected internal methods. Normally, this ID and password is used to log into the administrative console once security is turned on. You can use other users to log in if those users are part of the administrative roles.

You need to start the Administrative Console by specifying URL: http://<server_hostname>:9090/admin.

Perform the following steps to select LDAP as the user registry.

1. Click **Security** > **User Registry** > **LDAP** in the Navigation tree on the left.

---

2. On the LDAP user registry panel in the General Properties section of the Configuration tab, enter the Server user ID and password.

   This ID is the security server ID, which is only used for WebSphere Application Server security and is not associated with the system process that runs the server. The server calls the Local OS registry to authenticate and obtain privilege information about users by calling the native APIs in that particular registry.

---

3. In the type pull down, select the type of LDAP server to which you connect.

   The type is used to preload default LDAP properties. IBM Directory Server users can choose either IBM_Directory_Server or SecureWay as the directory type. Use the IBM_ Directory_server directory type for better performance. Users of the iPlanet Directory Server can choose either iPlanet Directory Server or NetScape as the directory type. Use the iPlanet Directory Server directory type for better performance after configuring the iPlanet to use role (nsRole) as the grouping method. For a list of supported LDAP servers, see the InfoCenter article, ″Supported directory services.″

---

4. In the Host box, enter the host ID (IP address or domain name system (DNS) name) of the LDAP server.

---

5. In the Port box, enter host port of the LDAP server. The default value is 389.

   If multiple WebSphere Application Servers are installed and configured to run in the same single signon domain, or if the WebSphere Application Server interoperates with a previous version of the WebSphere Application Server, then it is important that the port number match all configurations. For example, if the LDAP port is explicitly specified as 389 in a Version 4.0.x configuration, and a WebSphere Application Server at Version 5 is going to interoperate with the Version 4.0.x server, then verify that port 389 is specified explicitly for the Version 5 server.

---

6. In the Base Distinguished Name box, enter the base distinguished name of the directory service, indicating the starting point for LDAP searches of the directory service.

   For example, for a user with a distinguished name (DN) of cn=John Doe, ou=Rochester, o=IBM, c=US, you can specify the base DN as (assuming a suffix of c=us): ou=Rochester, o=IBM, c=us o=IBM, c=us c=us. For authorization purposes, this field is case sensitive. This implies that if a token is received (for example, from another cell or Domino) the base DN in the server must match exactly the base DN from the other cell or Domino. If case sensitivity is not a consideration for authorization, enable the Ignore Case field. This field is required for all LDAP directories except for the Domino Directory, where it is optional.

---

7. In the Bind Distinguished Name box, enter the distinguished name for the application server to use when binding to the directory service.

   If no name is specified, the application server binds anonymously. See the Base Distinguished Name field description for examples of distinguished names.

---

8. In the Bind Password box, enter the password for the application server to use when binding to the directory service.

---

9. In the Search Timeout box, enter the timeout value in seconds for an LDAP server to respond before aborting a request. The default value is 300.

---

10. Ensure that the Reuse Connection checkbox is checked.

    Enabled (or checked) is the default and specifies that the server should reuse the LDAP connection. Clear this option only in rare situations where a router is used to spray requests to multiple LDAP servers and when the router does not support affinity.

---

11. The Ignore Case checkbox allows you to enable or disable case insensitive authorization check.

    This field is required when IBM Directory Server is selected as the LDAP directory server. Otherwise, this field is optional and can be enabled when a case sensitive authorization check is required. For example, when you use certificates and the certificate contents do not match the case of the entry in the LDAP server. You can also enable the Ignore Case field when using single signon (SSO) between the product and Domino. Default: Disabled.

---

12. The SSL Enabled checkbox allows you to enable or disable secure socket communication to the LDAP server.

    When enabled, the LDAP Secure Sockets Layer (SSL) settings are used, if specified.

---

13. In the SSL Configuration pulldown, select the Secure Sockets Layer configuration to use for the LDAP connection.

    This configuration is used only when SSL is enabled for LDAP. Default: DefaultSSLSettings.

---

14. Click OK.

You know you are done when . . .

Steps for selecting custom user registry

Before you begin: Before you begin this task, implement and build the UserRegistry interface. For more information on developing custom user registries refer to the article, "Developing custom user registries" in the InfoCenter..

Perform the following steps to select a custom user registry.

1. Click **Security** > **User Registry** > **Custom** in the Navigation tree on the left.

---

2. On the Custom user registry panel in the General Properties section of the Configuration tab, enter the Server user ID and password.

   This ID is the security server ID, which is only used for WebSphere Application Server security and is not associated with the system process that runs the server. The server calls the Local OS registry to authenticate and obtain privilege information about users by calling the native APIs in that particular registry.

---

3. In the Custom User Registry box, enter the dot-separated class name that implements the com.ibm.websphere.security.UserRegistry interface.

   Put the custom registry class name in the class path. A suggested location is the %install_root%/classes directory. Although the custom registry implements the com.ibm.websphere.security.UserRegistry interface, for backward compatibility, a user registry can alternately implement the com.ibm.websphere.security.CustomRegistry interface. Default: com.ibm.websphere.security.FileRegistrySample

---

4. A check in the Ignore Case checkbox enables a case insensitive authorization check.

   Default: Enabled

---

5. Use the Custom Properties link to add any additional properties required to initialize the custom registry.

   The following property is pre-defined by the product; set this property only when required: WAS_UseDisplayName--When set to true, the methods getCallerPrincipal(), getUserPrincipal(), getRemoteUser() return the display name. By default, the securityName of the user is returned. This is primarily introduced to support backward compatibility with the Version 4.0 custom registry.

---

6. Click OK.

You know you are done when . . .

Selecting an authentication mechanism

The next step in setting up security is to select an authentication mechanism. An authentication mechanism defines rules about security information (for example, whether a credential is forwardable to another Java process), and the format of how security information is stored in both credentials and tokens. Authentication is the process of establishing whether a client is valid in a particular context. A client can be either an end user, a machine, or an application.

An authentication mechanism in WebSphere Application Server typically collaborates closely with a User Registry. The User Registry is the user and groups accounts repository that the authentication mechanism consults with when performing authentication. The authentication mechanism is responsible for creating a credential which is an internal product representation of successfully

authenticated client user. Not all credentials are created equal. The abilities of the credential are determined by the configured authentication mechanism.

Although this product provides several authentication mechanisms, only a single active authentication mechanism can be configured at once. The active authentication mechanism is selected when configuring WebSphere global security. WebSphere Application Server for z/OS V5 supports the following authentication mechanisms:

- Simple WebSphere Authentication Mechanism (SWAM)
- Light-Weight Third Party Authentication (LTPA)
- Integrated Crypotgraphic Service Facility (ICSF)

Steps for selecting the SWAM authentication mechanism

If you are using Simple WebSphere Authentication Mechanism (SWAM), there is no setup needed as this is the default mechanism.

Note: SWAM is only valid in a base installation. It is not supported in ND.

Continue with "Enabling global security" on page 147.

Steps for selecting LTPA as the authentication mechanism

Before you begin: You need to start the Administrative Console by specifying URL: http://<server_hostname>:9090/admin.

Perform the following steps to select LTPA as the authenication mechanism for this server.

1. Click **Security** > **Authentication Mechanisms** > **LTPA**in the Navigation tree on the left.

   ---

2. Enter the password and confirm it in the password fields. This password is used to encrypt and decrypt the LTPA keys during export and import of the keys. Remember this password because you enter it again when the keys from this cell are exported to another cell.

   ---

3. Enter a positive integer value in the Timeout field. This timeout value refers to how long an LTPA token is valid in minutes. The token contains this expiration time so that any server that receives the token can verify that the token is valid before proceeding further. When the token expires, the user is prompted to log in. An optimal value for this field depends on your configuration. The default value is 30 minutes.

   ---

4. Click Apply or OK. The LTPA configuration is now set.

   ---

5. Complete the information in the Global Security panel (see "Enabling global security" on page 147) and press OK. When OK or Apply is clicked in the Global Security panel the LTPA keys are generated automatically the first time, and therefore, you should not generate the keys manually.

You know you are done when . . .

Steps for selecting ICSF as the authentication mechanism

Before you begin: ICSF requires the Cryptographic Coprocessor features of the zSeries processor to be enabled and active. You need to have ICSF configured and running on your processor before selecting ICSF as your authentication mechanism. Refer to *z/OS ICSF Administrator's Guide*, SA22-7521.

You need to start the Administrative Console by specifying URL: http://<server_hostname>:9090/admin.

Perform the following steps to select ICSF as the authenication mechanism for this server.

1. Click **Security** > **Authentication mechanisms** > **ICSF** in the Navigation tree on the left.

   _____

2. In the Encryption Cryptographic Key box, specify the label of the cryptographic key to use for single sign-on tokens for Web applications and administrative security when using the Simple Object Access Protocol (SOAP) HTTP connector.

   _____

3. Enter a positive integer value in the Timeout field. Specifies the time period in which an ICSF token expires. Verify that this time period is longer than the cache time-out that is configured in the Global Security panel.

   _____

4. Click Apply or OK. The ICSF configuration is now set.

   _____

5. Continue with "Enabling global security" on page 147.

   _____

Enabling global security

The term global security refers to the security configuration that is effective for the entire security domain. A security domain consists of all servers configured with the same user registry realm name. Configuration of global security for a security domain consists of configuring the common user registry, the authentication mechanism, and other security information that defines the behavior of a security domain. Once Global Security is enabled, user identification must be provided to start and stop WebSphere.

It is helpful to understand security from an infrastructure standpoint so that you know the advantages of different authentication mechanisms, user registries, authentication protocols, and so on. Picking the right security components to meet your needs is a part of configuring global security. The following sections help you make these decisions. Read the following articles in the InfoCenter before continuing with the security configuration.

- Global Security and Server Security
- Getting started with Security

Once you understand the security components, you can proceed to configure global security in WebSphere Application Server.

Steps for enabling global security

Before you begin: Before you can enable global security you must select both an authentication mechanism and a user registry. If you have not performed these tasks, return to “Selecting an authentication mechanism” on page 145 and “Selecting a user registry” on page 141.

You need to start the Administrative Console by specifying URL: http://<server_hostname>:9090/admin.

Perform the following steps to enable global security

1. Click **Security** > **Global Security** in the Navigation tree on the left.

---

2. On the Global Security Configuration tab, the Enabled check box allows you to enable or not enable global security. Click the checkbox to enable.

   WebSphere Application Server security can be enabled or not enabled. You must enable security for all other security settings to function. Default: Not enabled.

---

3. The Enforce Java 2 Security checkbox allows you to enable or not enable Java 2 Security permission checking.

   By default, Java 2 security is disabled. However, if you enabled global security, this automatically enables Java 2 security. You can choose to disable Java 2 security, even when global security is enabled.

   When Java 2 Security is enabled and if an application requires more Java 2 security permissions then are granted in the default policy, then the application might fail to run properly until the required permissions are granted in either the app.policy file or the was.policy file of the application. AccessControl exceptions are generated by applications that do not have all the required permissions. Consult the InfoCenter and review the Java 2 Security and Dynamic Policy sections if you are unfamiliar with Java 2 security.

---

4. The Use Domain Qualified User IDs checkbox allows you to enable or not enable this option.

   If this option is enabled, user names will appear with their fully-qualified domain attribute when retrieved programmatically.

---

5. In the Cache Timeout box, enter the timeout value for security cache in seconds.

   When the timeout is reached, the Application Server clears the security cache and rebuilds the security data. Since this affects performance, this value should not be set too low. Default: 600 seconds.

---

6. The Issue Permission Warning checkbox allows you to enable or not enable this option.

   The filter.policy file contains a list of permissions that an application should not have. If an application is installed with a permission specified in this policy file and this option is enabled, a warning will be issued. Default: enabled.

7. The Active Protocol pulldown allows you to specify which security protocol is active when security is enabled.

   Specifies the active authentication protocol for RMI/IIOP requests when security is enabled. In previous releases the z/SAS protocol was the only available protocol. This release includes an OMG protocol called CSIv2 which supports increased vendor interoperability and additional features. If all servers in your entire security domain are Version 5.0 servers, it is best to specify CSI as your protocol. If some servers are 3.x or 4.x servers, specify CSI and zSAS. Default: Both CSI and zSAS.

8. The Active Authentication Mechanism pulldown specifies the authentication mechanism which is active when security is enabled.

   In WebSphere Application Server, Version 5, Simple WebSphere Authentication Mechanism (SWAM), Lightweight Third Party Authentication (LTPA), and Integrated Cryptographic Services Facility (ICSF) are the supported authentication mechanisms. Only ICSF and LTPA are configurable on WebSphere Application Server Network Deployment, Version 5. SWAM is not.

9. The Active User Registry pulldown specifies the user registry which is active when security is enabled.

   You can configure settings for one of the following user registries:

   - Local operating system. The implementation is a SAF compliant registry such as the Resource Access Control Facility (RACF), which is shared in an MVS sysplex.
   - LDAP user registry. The LDAP User Registry settings are used when users and groups reside in an external LDAP directory. When security is enabled and any of these properties are changed, go to the Global Security panel and click Apply or OK to validate the changes.
   - Custom user registry.

   Default: Local OS.

10. Click OK.

    This panel performs a final validation of the security configuration. When you click OK or Apply from this panel, the security validation routine is performed and any problems are reported at the top of the page. When you complete all of the fields, click OK or Apply to accept the selected settings. Click Save (at the top of the panel) to persist these settings out to a file. If you see any informational messages in red text color, then there is a problem with the security validation. Typically, the message indicates the problem. So, review your configuration to verify that the user registry settings are accurate and the correct registry is selected. In some cases, the LTPA configuration may not be fully specified. See the Global security settings article in the InfoCenter for detailed information.

You know you are done when no error messages appear at the top of the page.

How clients and clusters negotiate security protocols

Because there are several security protocols supported by clients and clusters, there are many possible ways a client and cluster can secure their communications. A cluster may support many security mechanisms simultaneously. At run time, a client and cluster dynamically negotiate the kind of security used for their interaction. For instance, one client may support user ID/password security, another client may support SSL security, while the cluster they interact with may support SSL, and user ID/password security. Each client and cluster negotiates the type of security to use based on an ordered list of choices. The negotiation starts at the top of the list. If the client and cluster cannot agree to the type of security at the top of the list, negotiation continues to the second type of security on the list, then the third, and so on. This negotiation continues until the client and cluster agree on the type of security they will use. Once the type of security to use is negotiated, the authentication phase begins. If authentication fails, communication ends and the client request fails.

Notes:

1. It is possible that the negotiation between client and cluster ends in no security being used.

The ordered list of choices a client uses varies depending on the kind of interaction between the client and cluster. Figure 11 shows the types of interactions between clients and clusters. The number labels on the diagram are explained in Table 56.

Figure 11. Interactions between clients and clusters

Table 56. Ordered list of choices based on interaction

Item	Type of interaction	Ordered list used for this interaction
1	cluster to cluster within the cell	1. CSIv2 asserted identity 2. Kerberos over SSL 3. IBM asserted identity 4. User ID/PassTicket 5. SSL client certificates 6. User ID/password 7. No security
2	cluster to a remote z/OS cluster	1. CSIv2 asserted identity 2. Kerberos over SSL 3. IBM asserted identity 4. SSL client certificates 5. User ID/password 6. No security
3	Client to cluster within a sysplex	1. CSIv2 client authentication 2. SSL client certificates 3. Kerberos over SSL 4. SSL basic authentication 5. User ID/PassTicket 6. User ID/password 7. No security
4	Client to cluster within a z/OS system	User ID (RACO) always used

Table 56. Ordered list of choices based on interaction (continued)

Item	Type of interaction	Ordered list used for this interaction
5	Client to a remote z/OS cluster	1. CSIv2 client authentication 2. SSL client certificates 3. Kerberos over SSL 4. SSL basic authentication 5. User ID/password 6. No security
6[1]	cluster to workstation	1. CSIv2 asserted identity 2. SSL client certificates 3. No security
7[1] and 9[1]	Workstation to z/OS cluster	1. CSIv2 client authentication 2. Determined by the workstation client configuration
8[1]	Client to workstation	1. CSIv2 client authentication 2. SSL with DCE principal/password authentication 3. No security

1. Subject to the workstation configuration. See the specific workstation product documentation. SSL Client Certificates are standard in the industry. Depending on the type and configuration, WebSphere on a distributed platform may support proprietary authentication mechanisms such as SSL Basic Authentication.

Setting up SSL security for WebSphere for z/OS

This topic assumes you understand the SSL protocol and how Cryptographic Services System SSL works on z/OS or OS/390. For information about the SSL protocol, go to the following web site:

`http://home.netscape.com/eng/ssl3/ssl-toc.html`

For more information about Cryptographic Services System SSL, see *z/OS System Secure Sockets Layer Programming*, SC24-5901.

Secure Sockets Layer (SSL) is used by multiple components within WebSphere Application Server to provide trust and privacy. These components are the built-in HTTP Transport, the ORB (client and server), and the secure LDAP client. Configuring SSL is different between client and server with WebSphere Application Server. If you want the added security of protected communications and user authentication in a network, you can use Secure Sockets Layer (SSL) security. The SSL support in WebSphere for z/OS has several objectives:

- To provide ways accepted by the industry to protect the security of messages as they flow across the network. This is often called *transport layer security*. Transport layer security is a function that provides privacy and data integrity between two communicating applications. The protection occurs in a layer of software on top of the base transport protocol (for example, on top of TCP/IP).

  SSL provides security over the communications link through encryption technology, ensuring the integrity of messages in a network. Because communications are encrypted between two parties, a third party cannot tamper

with messages. SSL also provides confidentiality (ensuring the message content cannot be read), replay detection, and out-of-sequence detection.

- To provide a secure communications medium through which various authentication protocols may operate. A single SSL session can carry multiple authentication protocols, that is, methods to prove the identities of the parties communicating.

  SSL support always provides a mechanism by which the server proves its identity. The SSL support on WebSphere for z/OS allows these ways for the client to prove its identity:

  – Basic authentication (also known as SSL Type 1 authentication), in which a client proves its identity to the server by passing a user identity and password known by the target server.

    With SSL basic authentication:

    - A z/OS or OS/390 client can communicate securely with a WebSphere for z/OS server by using a user ID and password as defined by the CSIv2 Username and Password Mechanism (GSSUP).
    - A distributed platform client can communicate securely with a WebSphere for z/OS server by using a MVS user ID and password.
    - Because a password is always required on a request, only simple client-to-server connections can be made. That is, the server cannot send a client's user ID to another server for a response to a request.

  – Client certificate support, in which both the server and client supply digital certificates to prove their identities to each other.

    Web applications may have thousands of clients, which makes managing client authentication an administrative burden. Through RACF *certificate name filtering*, SSL support on WebSphere for z/OS allows you to map client certificates, without storing them, to MVS user IDs. Through certificate name filtering, you can authorize sets of users to access servers without the administrative overhead of creating MVS user IDs and managing client certificates for every user.

  – CSIv2 identity assertion support, which includes z/OS and OS/390 principals, X501 distinguished names, Kerberos principals, and X509 identity certificates.

  – Identity assertion, or trusted association, in which an intermediate server can send the identities of its clients to a target server in a secure yet efficient manner. This support uses client certificates to establish the intermediate server as the owner of an SSL session. Through RACF, the system can check that the intermediate server can be trusted (to confer this level of trust, CBIND authorization is granted by administrators to RACF IDs that run secure system code exclusively). Once trust in this intermediate server is established, client identities (MVS user IDs) need not be separately verified by the target server; those client identities are simply asserted without requiring authentication.

- To interoperate in a secure way with other products such as:
  – CICS Transaction server for z/OS
  – WebSphere on distributed platforms
  – CORBA-compliant Object Request Brokers

SSL is disabled by default and SSL support is optional. Running WebSphere for z/OS without using SSL affects only the SSL functions that protect communication and authenticate clients and servers.

If you choose to use SSL, there are two types of SSL repertoires from which you must choose:

- System SSL (SSSL) is the SSL repertoire type used for Web container and ORB transport.
- Java Secure Socket Extension (JSSE) is the SSLrepertoire type used for the JMX SOAP Connector

The following describes how an SSL connection works:

Stage	Description
Negotiation	After the client locates the server, the client and server negotiate the type of security for communications. If SSL is to be used, the client is told to connect to a special SSL port.
Handshake	The client connects to the SSL port and the SSL handshake occurs. If successful, encrypted communication starts. The client authenticates the server by inspecting the server's digital certificate. If client certificates are used during the handshake, the server authenticates the client by inspecting the client's digital certificate.
Ongoing communication	During the SSL handshake, the client and server negotiate a cipher spec to be used to encrypt communications.
First client request	The determination of client identity depends upon the client authentication mechanism chosen, which is one of the following: • CSIv2 user id and password (GSSUP) • CSIv2 asserted identiy • zSAS Kerberos • z/SAS Basic Authentication Asserted Identities • z/SAS Asserted Identities

Rules:

- Only server controllers and z/OS or OS/390 clients require access to Cryptographic Services System SSL. Your controllers and z/OS or OS/390 clients require access to the *hlq*.SGSKLOAD data set. Place SGSKLOAD into LPA. For more information, see *z/OS System Secure Sockets Layer Programming*, SC24-5901.
- Either a Java or C++ client on z/OS or OS/390 can interoperate with a WebSphere for z/OS or workstation server and use SSL. CSIv2 security only supports Java clients on z/OS or OS/390.
- Part of the handshake is to negotiate the cryptographic specs used by SSL for message protection. There are two factors that determine the cipher specs and key sizes used:
  - The security level of the Cryptographic Services installed on the system, which determines the cipher specs and key sizes available to WebSphere for z/OS.
  - The configuration of the server through the Administrative Console allows you to specify SSL cipher suites.

  (For more information, see *z/OS System Secure Sockets Layer Programming*, SC24-5901.)
- For z/OS System SSL sockets you must use RACF or equivalent for storing digital certificates and keys. Placing digital certificates and keys into a key database in the HFS is not an option.

Overview of SSL basic authentication security for your Application Server and clients

To define SSL basic authentication security, you must first request a signed certificate for your server and a certificate authority (CA) certificate from the certificate authority that signed your server certificate. The process of requesting certificates is beyond the scope of this manual. For more information about requesting a certificate, see *z/OS System Secure Sockets Layer Programming*, SC24-5901.

After you have received a signed certificate for your server and a CA certificate from the certificate authority, you must use RACF to authorize the use of digital certificates, store server certificates and server key rings in RACF, create an SSL repetoire alias, and define SSL security properties for your server through the Administrative Console.

For clients, you must create a key ring and attach to it the CA certificate from the certificate authority that issued the server's certificate. For a z/OS or OS/390 client, you must use RACF to create a client key ring and to attach the CA certificate to that key ring.

Figure 12 on page 155 shows the certificate arrangement involved in SSL basic authentication.

- **For the client to authenticate the server**, the server (actually, the controller user ID) must possess a signed certificate created by a certificate authority (CA). The server passes the signed certificate to prove its identity to the client. The client must possess the CA certificate from the same certificate authority that issued the server's certificate. The client uses the CA certificate to verify that the server's certificate is authentic. Once verified, the client can be sure that messages are truly coming from that server, not someone else.
- **For the server to authenticate the client**, note that there is no client certificate that the client passes to prove its identity to the server. In the SSL basic authentication scheme, the server authenticates the client by challenging the client for a user ID and password.

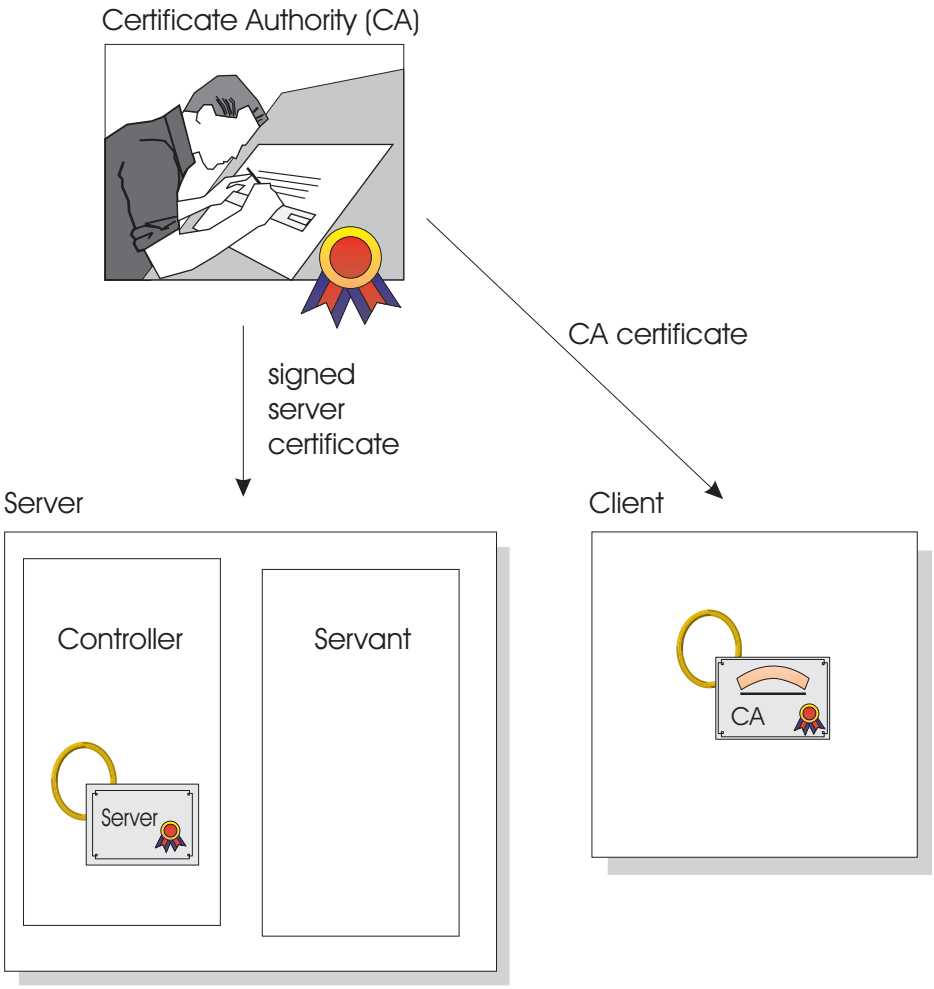


Figure 12. Certificate arrangement for SSL basic authorization

Rules:

- For Java clients on platforms other than z/OS or OS/390, you must have WebSphere Application Server Enterprise Edition 3.5 or WebSphere Advanced Edition 4.0 on those platforms to interoperate with a WebSphere for z/OS server and use SSL basic authentication. C++ clients on other platforms cannot use SSL basic authentication when interoperating with WebSphere for z/OS.
- For SSL basic authentication, clients are authenticated in the following ways:
  - A z/OS or OS/390 client communicating with a remote z/OS or OS/390 server uses the remote user ID and password (REM_USERID and REM_PASSWORD) WebSphere variables in the client environment file to authenticate the client identity.
  - If a z/OS or OS/390 client uses SSL with a Component Broker server on other platforms, the client must pass a DCE principal and password defined to the server by using the REM_DCEPRINCIPAL and REM_DCEPASSWORD WebSphere variables.
  - A z/OS or OS/390 client must also identify its key ring through the SSL_KEYRING WebSphere variable.
  - A client on a WebSphere Application Server distributed platform communicating with a z/OS or OS/390 server uses a user dialog supplied by the ORB, in which the user supplies a user ID and password.

The following table shows the subtasks and associated procedures for defining SSL basic authentication security:

Subtask	Associated procedure (See . . .)
Requesting a server certificate and a certificate authority (CA) certificate	*z/OS System Secure Sockets Layer Programming*, SC24-5901
Create an SSL repertoire alias	
Setting up SSL basic authentication security for servers	"Steps for using RACF to authorize the server to use digital certificates" on page 158
Setting up SSL basic authentication security for clients	"Steps for setting up SSL security for clients" on page 163

Overview of SSL client certificate security for your Application Server and clients

To define SSL client certificate security, you must first request signed certificates for your server and clients and certificate authority (CA) certificates from the certificate authority that signed those certificates. The process of requesting certificates is beyond the scope of this manual. For more information about requesting a certificate, see *z/OS System Secure Sockets Layer Programming*, SC24-5901.

After you have received signed certificates and CA certificates from the certificate authority, you must use RACF to authorize the use of digital certificates, store certificates and key rings in RACF, and define SSL security properties for your server through the Administrative Console.

Each client identified by a digital certificate must eventually be converted into a MVS user ID by the target WebSphere for z/OS server. If the client and server share the same RACF database, then you do not have to do any additional configuration for this mapping. If the client and server do not share the same RACF database, you can configure the mapping by:

- Adding client certificates to the RACF database of the target server. This may be impractical in most cases.
- Mapping groups of clients into RACF identities using RACF certificate name filtering.
- Using a combination of the two.

Figure 13 on page 157 shows the certificate arrangement involved in SSL client certificate authentication.

- **For the client to authenticate the server**, the server (actually, the controller user ID) must possess a signed certificate created by a certificate authority (CA). The server passes the signed certificate to prove its identity to the client. The client must possess the CA certificate from the same certificate authority that issued the server's certificate. The client uses the CA certificate to verify that the server's certificate is authentic. Once verified, the client can be sure that messages are truly coming from that server, not someone else.
- **For the server to authenticate the client**, the client must possess a signed certificate created by a certificate authority (CA2). (In Figure 13 on page 157 we show two different certificate authorities for clarification; it is possible that the same certificate authority supplies signed certificates to both the server and client.) The server must possess the CA2 certificate from the same certificate authority that issued the client's certificate. The server uses the CA2 certificate to

verify that the client's certificate is authentic. Once verified, the server can be sure that messages are truly coming from that client, not someone else.

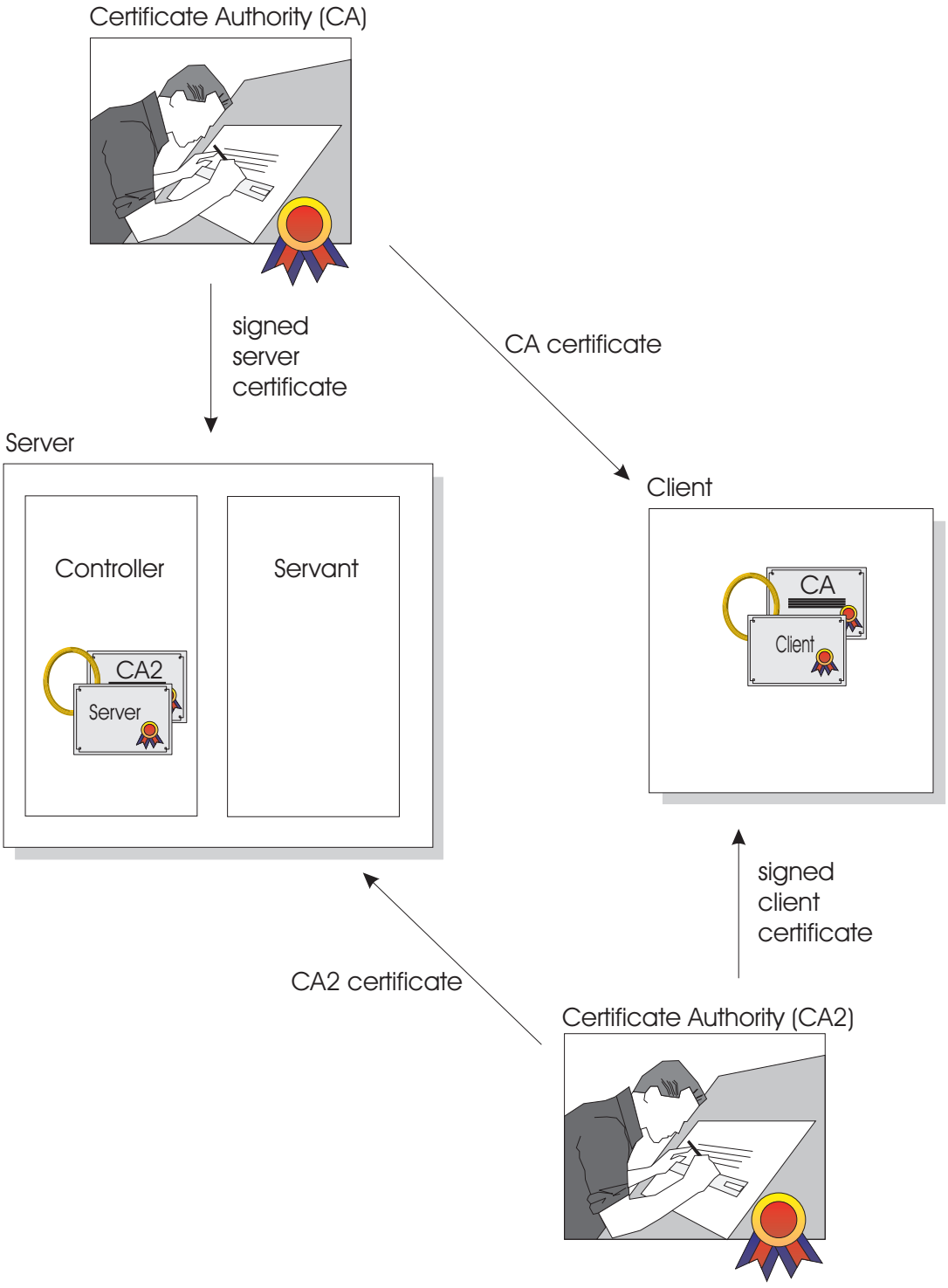


Figure 13. Certificate arrangement for SSL client certificate security

The following table shows the subtasks and associated procedures for defining SSL client certificate security:

Subtask	Associated procedure (See . . .)
Requesting a server certificate and a certificate authority (CA) certificate	*z/OS System Secure Sockets Layer Programming*, SC24-5901
Setting up SSL client certificate security for servers	"Steps for using RACF to authorize the server to use digital certificates"
Create an SSL repertoire alias	"Overview of creating a new SSL repertoire alias" on page 159
Setting up SSL client certificate security for clients	"Steps for setting up SSL security for clients" on page 163
Mapping client digital certificates to MVS user IDs on your server's system	"Steps for mapping client digital certificates to MVS user IDs on your server's system" on page 165

Defining SSL security for clients and servers

This section includes the procedures you must follow to implement all SSL–based authentication mechanisms.

Steps for using RACF to authorize the server to use digital certificates: SSL uses digital certificates and public/private keys. If your Application Server uses SSL, you must use RACF to store digital certificates and public/private keys for the user identities under which the server controllers run.

Before you begin: You need to request a certificate authority (CA) certificate and a signed certificate for your server.

If you plan to implement SSL client certificate support, you must also have certificate authority (CA) certificates from each certificate authority that verifies your client certificates. See *z/OS System Secure Sockets Layer Programming*, SC24-5901.

You must have a user ID with the authority to use the RACDCERT command in RACF (for example, SPECIAL authority). For details about RACDCERT, see *z/OS Security Server RACF Command Language Reference*, SA22-7687, and *z/OS Security Server RACF Security Administrator's Guide*, SA22-7683.

Perform the following steps authorizing the use of digital certificates:

1. For each server that uses SSL, create a key ring for that server's controller user ID.

   Example: Your controller is associated with the user ID called ASCR1. Issue:

   ```
   RACDCERT ADDRING(ACRRING) ID(ASCR1)
   ```

   ---

2. Receive the certificate for your Application Server from the certificate authority.

   Example: You requested a certificate and the certificate authority returned the signed certificate to you, which you stored in a file called ASCR1.CA. Issue:

   ```
   RACDCERT ID (ASCR1) ADD('ASCR1.CA') WITHLABEL('ACRCERT') PASSWORD('password')
   ```

   ---

3. Connect the signed certificate to the controller user ID's key ring and make the certificate the default certificate.

Example: Connect the certificate labelled ACRCERT to the key ring ACRRING owned by ASCR1. Issue:

```
RACDCERT ID(ASCR1) CONNECT (ID(ASCR1) LABEL('ACRCERT') RING(ACRRING) DEFAULT)
```

---

4. If you plan to have the server authenticate clients (SSL client certificate support):
   - Receive each certificate authority (CA) certificate that verifies your client certificates. Give each CA certificate the CERTAUTH attribute.

     Example: Receive the CA certificate that will verify a client with user ID CLIENT1. That certificate is in a file called USER.CLIENT1.CA. Issue:

     ```
     RACDCERT ADD('USER.CLIENT1.CA') WITHLABEL('CLIENT1 CA') CERTAUTH
     ```

   - Connect each client's certificate authority (CA) certificate to the controller user ID's key ring.

     Example: Connect the CLIENT1 CA certificate to the ring ACRRING owned by ASCR1.

     ```
     RACDCERT ID(ASCR1) CONNECT(CERTAUTH LABEL('CLIENT1 CA') RING(ACRRING))
     ```

---

5. Give read access for IRR.DIGTCERT.LIST and IRR.DIGTCERT.LISTRING in the RACF FACILITY class to the controller user ID.

   Example: Your controller user ID is ASCR1. Issue:

   ```
   PERMIT IRR.DIGTCERT.LIST     CLASS(FACILITY) ID(ASCR1) ACC(READ)
   PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(ASCR1) ACC(READ)
   ```

You are done with the RACF phase when the RACF commands succeed. Continue on to .

Overview of creating a new SSL repertoire alias: WebSphere Application Server for z/OS V5 supports two SSL repertoire types:

- System SSL (SSSL) is the SSL repertoire type used for the Web container and ORB transport.
- Java Secure Socket Extension (JSSE) is the SSL repertoire type used for the JMX SOAP Connector

You must first create an SSL configuration repertoire alias or entry. You can then select the alias later when a component is configured for SSL support. The SSL configuration repertoire allows administrators to define any number of SSL settings which can be used to make HTTPS, IIOPS or LDAPS connections. You can pick one of the SSL settings defined here from any location within the Administrative Console which allows SSL connections. This simplifies the SSL configuration process since you can reuse many of these SSL configurations by simply specifying the alias in multiple places. The appropriate repertoire is referenced during the configuration of a service that sends and receives requests encrypted using SSL, such as the Web and enterprise beans containers. Before deleting SSL configurations from the repertoire, remember that if an SSL configuration alias is referenced somewhere, and it is deleted here, an SSL connection will fail if the deleted alias is accessed.

This next sections describe the steps you need to follow to create a new SSSL or JSSE repertoire alias.

Steps for creating a System SSL repertoire alias: **Before you begin:** You need to start the Administrative Console by specifying URL: http://<server_hostname>:9090/admin.

Perform the following steps to create a new System SSL repertoire alias:

1. Click **Security** > **SSL** on the left-hand navigation tree to open the SSL Configuration Repertoires panel.

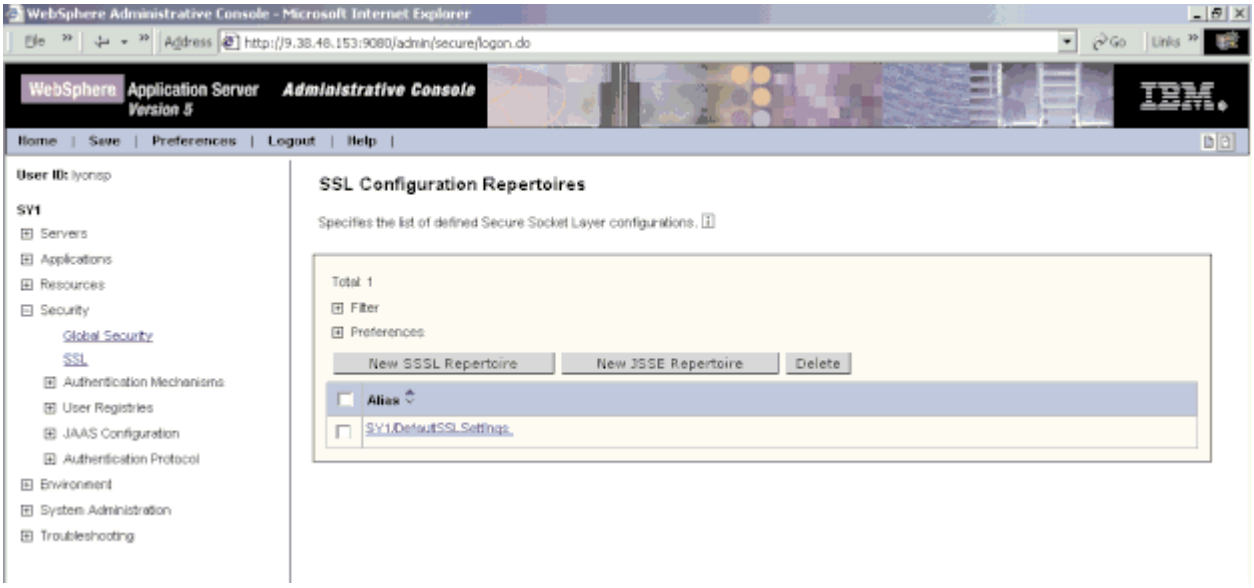


Figure 14. The SSL Configurations Repertoires panel

.

---

2. To create a new System SSL alias, click in the checkbox next to the word **Alias** and click on the **New SSSL Repertoire** button near the top of the panel. The System SSL Repertoire panel appears.

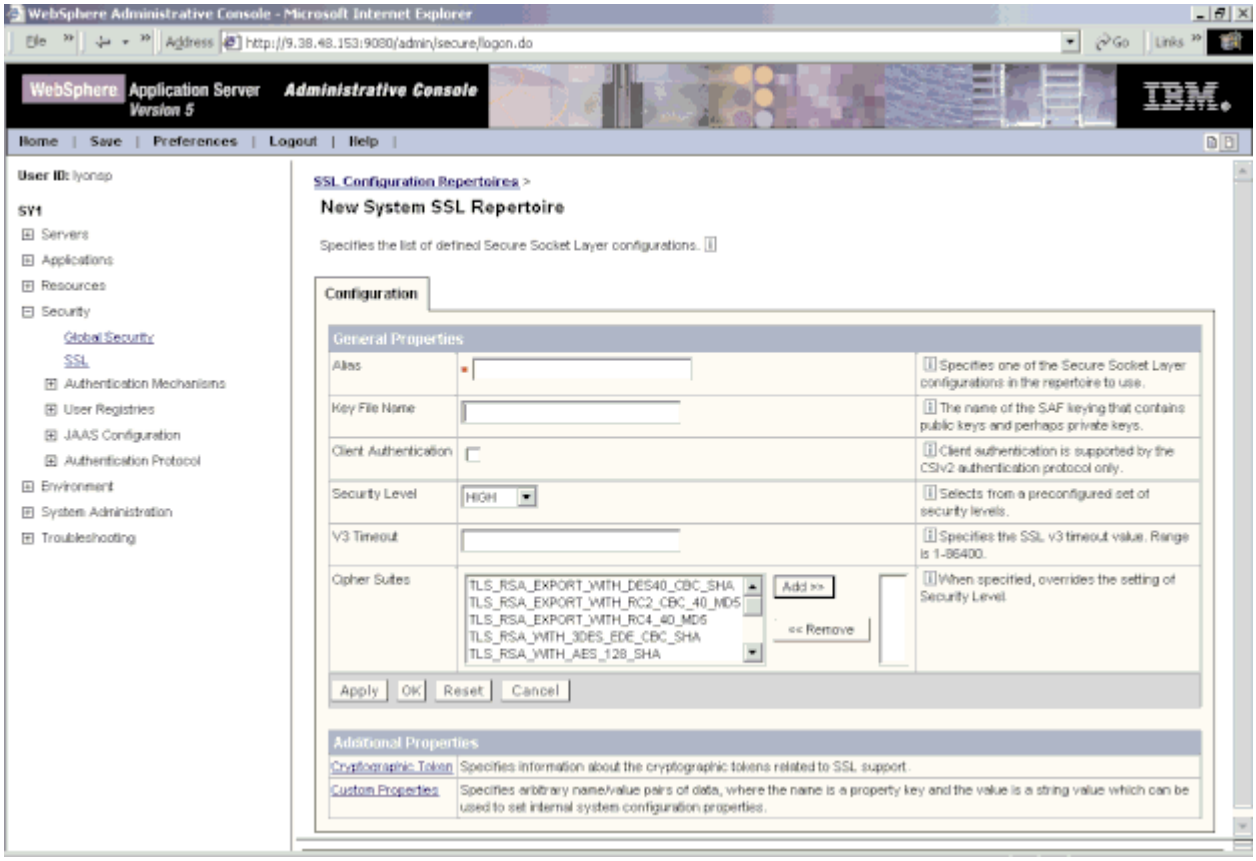


Figure 15. System SSL repertoire panel

3. Enter the alias name in the **Alias** box

4. Specify the SSL RACF key ring. This is the key ring you defined in step 1 in "Steps for using RACF to authorize the server to use digital certificates" on page 158.

   Note: If you specify the wrong RACF key ring, the server gets an error message at run time.

5. If using the CSIv2 authentication protocol, you may select Client Authentication by checking the box.

   Note: For authentication with the IIOP protocol (for EJB requests), you must click **Security** > **Authentication Protocol** > **CSIv2 Inbound or Outbound Authentication** from the left navigation panel of the administrative console. Select SSL Client Certificate Authentication to enable it for these requests.

6. Select High, Medium, or Low from the Security Level pull down.

7. Specify the SSL V3 timeout value, which is the length of time, in seconds, that the system holds session keys. The range is 0-86400 (1 day). The default is 600 seconds.

8. Click on the cipher suites you want to add. By default, this is not set and the set of cipher suites available is determined by the value of the Security Level (High, Medium, or Low). A cipher suite is a combination of cryptographic algorithms used for an SSL connection.The available cipher suites are presented in the following table:

Table 57. System SSL cipher suites

System SSL Cipher Suites	Description
High Security Level	
TLS_RSA_WITH_RC4_128_MD5	128-bit RC4 encryption with MD5 message authentication and RSA key exchange.
TLS_RSA_WITH_RC4_128_SHA	128-bit RC4 encryption with SHA-1 message authentication and RSA key exchange.
TLS_RSA_WITH_3DES_EDE_CBC_SHA	168-bit Triple DES encryption with SHA-1 message authentication and RSA key exchange.
TLS_RSA_WITH_AES_128_SHA	128-bit AES encryption with SHA-1 message authentication and RSA key exchange.
TLS_RSA_WITH_AES_256_SHA	256-bit AES encryption with SHA-1 message authentication and RSA key exchange.
Medium Security Level	
TLS_RSA_EXPORT_WITH_RC4_40_MD5	40-bit RC4 encryption with MD5 message authentication and RSA key exchange.
TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5	40-bit RC2 encryption with MD5 message authentication and RSA key exchange.
TLS_RSA_WITH_DES_CBC_SHA	56-bit DES encryption with SHA-1 message authentication and RSA key exchange.
Low Security Level	
TLS_RSA_WITH_NULL_MD5	No encryption with MD5 message authentication.
TLS_RSA_WITH_NULL_SHA	No encryption with SHA-1 message authentication.

9. Click OK when you have made all your selections.

Rules:

- All System SSL repertoires that become effective on a server must have the same v3 timeout values.
- All System SSL repertoires that become effective on a server must specify the same key file name.
- Web Container, CSIv2, and zSAS SSL selections may only choose SSSL type repertoires.
- SOAP Connector may choose only JSSE type repertoires.

You know you are done when . . .

Steps for creating a JSSE SSL repertoire alias: **Before you begin:** You need to start the Administrative Console by specifying URL: http://<server_hostname>:9090/admin.

Perform the following steps to create a new JSSE SSL repertoire alias:

1. Click **Security** > **SSL** on the left-hand navigation tree to open the SSL Configuration Repertoires panel.

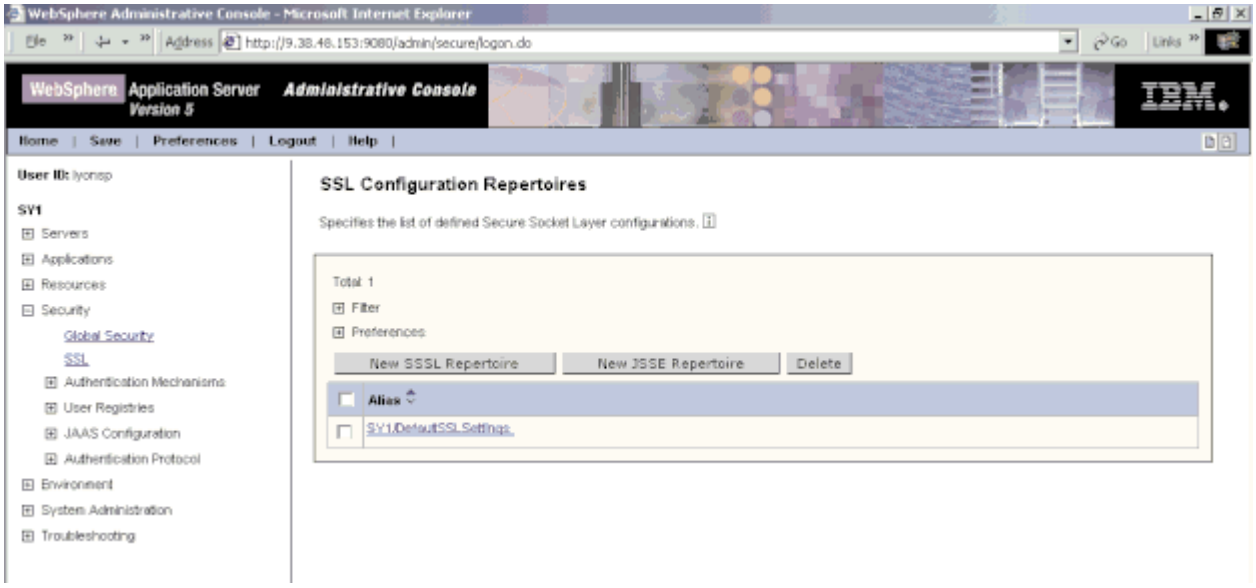


Figure 16. The SSL Configurations Repertoires panel

.

---

2. To create a new JSSE SSL alias, click in the checkbox next to the word **Alias** and click on the **New JSSE Repertoire** button near the top of the panel.

   Note: The only time you would select the JSSE repertoire types is when you are setting up SSL for JMX SOAP connectors.

   The System SSL Repertoire panel appears. For the complete steps to create a JSSE SSL repertoire alias, refer to the the "Security" section in the WebSphere Application Server InfoCenter, access to which can be obtained through the WebSphere for z/OS library Web site http://www.ibm.com/software/webservers/appserv/zos_os390/library.html

---

Steps for setting up SSL security for clients: All clients must have access to the server's certificate authority (CA) certificate so they can authenticate the server during the SSL handshake. If you plan to implement SSL client certificate support, clients additionally must have their own certificates as the default certificate on their key rings.

- If your clients are connecting to WebSphere for z/OS from WebSphere on workstations, you must import SSL certificates into the workstation system. For more information and instructions, see IBM WebSphere InfoCenter.
- On z/OS or OS/390, clients must have certificates attached to their keyrings in RACF.

This procedure explains how to attach certificates to z/OS or OS/390 clients.

Before you begin: For SSL basic authentication and Kerberos, you must request a CA certificate from the same certificate authority that issued signed certificates for your Application Servers. If you plan to implement SSL client certificate support, you must additionally request a signed certificate for the client from a certificate authority.

You must have a user ID with the authority to use the RACDCERT command in RACF (for example, SPECIAL authority). For details about RACDCERT, see *z/OS Security Server RACF Command Language Reference*, SA22-7687, and *z/OS Security Server RACF Security Administrator's Guide*, SA22-7683.

Perform the following steps to authorize use of digital certificates by z/OS or OS/390 clients:

1. Create a key ring for the z/OS or OS/390 client.

   Example: Your client user ID is CLIENT1. Issue:

   ```
   RACDCERT ADDRING(C1RING) ID(CLIENT1)
   ```

   _____

2. Receive the server's certificate authority (CA) certificate and give it the CERTAUTH attribute.

   Example: You requested a CA certificate and the certificate authority returned its certificate to you, which you stored in a file called USER.WSSERVER.CA. Issue this command:

   ```
   RACDCERT ADD('USER.WSSERVER.CA') WITHLABEL('VERI CA') CERTAUTH
   ```

   _____

3. Connect the server's CA certificate to the client key ring.

   Example: Connect the VERI CA certificate to the C1RING key ring owned by CLIENT1.

   ```
   RACDCERT ID(CLIENT1) CONNECT(CERTAUTH LABEL('VERI CA') RING(C1RING))
   ```

   _____

4. In the client's environment file, code the security.sslkeyring= WebSphere variable to correspond to the client's key ring.

   For more information, see the WebSphere variables in the Administrative Console or the InfoCenter.

   _____

5. If you are implementing SSL client certificate support:
   - Receive the certificate for your client from the certificate authority.

     Example: You requested a certificate and the certificate authority returned a signed certificate which you stored in CLIENT1.SIGNED.CERT. Issue:

     ```
     RACDCERT ID (CLIENT1) ADD('CLIENT1.SIGNED.CERT') WITHLABEL('CLIENT1 CERT') PASSWORD('password')
     ```

   - Connect the client's signed certificate to the client user ID's key ring and make the certificate the default certificate.

Example: Connect the certificate labelled CLIENT1 to the key ring C1RING owned by CLIENT1. Issue:

```
RACDCERT ID(CLIENT1) CONNECT (ID(CLIENT1) LABEL('CLIENT1 CERT') RING(C1RING) DEFAULT)
```

You are done when the RACF commands succeed and you save your environment file.

Steps for mapping client digital certificates to MVS user IDs on your server's system: Each client that presents a digital certificate to authenticate its identity, but does not have an individual certificate registered with RACF on the target server's system or cell, must have a mapping to a valid MVS user ID. You can create this mapping by using RACF certificate name filters.

You can create RACF certificate name filters based on either the client's or certificate issuer's distinguished name, as contained in the X.509 digital certificates.

Before you begin: You should know how you want to organize sets of clients that will be presenting digital certificates, and what sort of access those clients need.

You need to have the authority to issue the RACDCERT MAP command.

Perform the following steps to set up certificate name filtering:

1. Define a MVS user ID for each user ID you associate with a certificate name filter. Consider assigning the PROTECTED and RESTRICTED attributes to each one. The PROTECTED attribute protects the user ID from being used to log on directly to the system and from being revoked through incorrect password attempts. The RESTRICTED attribute ensures that the user ID will not be used to access protected resources it is not explicitly authorized to access. **Example:**

   ```
   ALTUSER WEBUSER NOPASSWORD RESTRICTED
   ```

   ---

2. Activate certificate name filtering. **Example:**

   ```
   SETROPTS CLASSACT(DIGTNMAP) RACLIST(DIGTNMAP)
   ```

   ---

3. Create a certificate name filter. **Example:** The following filter associates the user ID WEBUSER to any user presenting a certificate issued by VeriSign Class 1, who does not have an individual certificate registered with RACF on your system:

   ```
   RACDCERT ID(WEBUSER) MAP WITHLABEL('INTERNET OTHERS') +
           IDNFILTER('OU=VeriSign Class 1 Individual Subscriber.O=VeriSign, Inc.L=Internet')
   ```

   This filter is based on the issuer's name. You can create other filters based on the subject's name, or on combinations of the issuer's and subject's names. For more information about certificate name filtering, see *z/OS Security Server RACF Security Administrator's Guide*, SA22-7683.

   ---

4. Refresh the DIGTNMAP class. **Example:**

   ```
   SETROPTS RACLIST(DIGTNMAP) REFRESH
   ```

You are done when the SETROPTS command completes.

Using certificates to set up secure HTTPS internal transport connections: An HTTPS internal transport can use server and client certificates to set up secure server-client connections for HTTPS application requests. The HTTPS internal transport enables you to set up client authentication using:

- Server certificates you have created and are administering, and for which you are your own certificate authority (CA).
- Client certificates signed by an internal CA. (Using an internal CA to sign your client certificates is independent of whether you used an internal or external CA to sign your server certificate.)
- Cluster certificates signed by an external CA.
- Client certificates that are signed by an external CA. (Using an external CA to sign your client certificates is independent of whether you used an internal or external CA to sign your server certificate.)

Before you can use a server certificate to set up secure HTTPS internal transport connections, you must:

- Create or obtain a server certificate, if you don't already have one.
- Create or obtain a CA certificate if you don't already have one.
- Create a controller key ring that is connected to your server certificate, and has this certificate as the default for this key ring.
- Configure the HTTP transport:
  1. Open the Administrative Console.
  2. Click **Servers** > **Application Servers** on the left-hand navigation tree.
  3. Click on the name of the server.
  4. On the **Additional Properties** menu of the Server panel, click **Web Container** .
  5. On the **Additional Properties** menu of the Web Container panel, click **HTTP Transport**.
  6. Click on the **Host** you want to configure.
  7. Enter the **Port** number you want to bind (For HTTPS it is usually 443).
  8. Click the checkbox to **Enable SSL**.
  9. Select the SSL alias in which you specified the controller key ring.
  10. Click **OK**.

If you also want to use a client certificate to set up secure HTTPS internal transport connections, you must perform the following additional tasks:

- Use the Administrative Console to specify that client certificates are allowed.
- Create or obtain a client certificate, if you don't already have one.

See "Steps for setting up secure HTTPS internal transport connections using a server certificate signed by an internal CA", "Steps for setting up secure HTTPS internal transport connections using client certificates signed by an internal CA" on page 171,"Steps for setting up secure HTTPS internal transport connections using server certificates signed by an external CA" on page 175, and "Steps for setting up secure HTTPS internal transport connections using client certificates signed by an external CA" on page 181 for more information on how to perform these steps.

Steps for setting up secure HTTPS internal transport connections using a server certificate signed by an internal CA: Using SSL, WebSphere for z/OS allows you to set up your own certificate authority, and administer your own certificates.

Notes:

1. Acting as your own certificate authority (CA) is recommended only for test environments and private intranets. With this method, you set up your own CA and sign certificates. You can optionally use client authentication to verify the identity of those accessing your controller.
2. You must use System SSL to establish secure connections. To use System SSL with the HTTPS internal transport, the System SSL load library must exist in linklist and must be under program control. If you have not already done so:
   - Add the load library to the linklist.
   - Turn on program control for the library by issuing the following RACF commands from a user ID that has the proper authority:

     ```
     RALTER PROGRAM * ADDMEM('hlq.SGSKLOAD'//NOPADCHK) UACC(READ)
     SETROPTS WHEN(PROGRAM) REFRESH
     ```

     If turning on program control for the first time, use the RDEFINE command instead of the RALTER command.
3. You will issue the RACF command, RACDCERT, to create certificates and key rings for your Java server. On most of the RACDCERT commands you must specify a user ID. This ID must be the same user ID as the controller ID for your server. If it is not, SSL will not initialize. The following example uses ASCR1 as the controller ID. Therefore, ASCR1 is specified as the ID on the RACCERT commands in this example.

Before you begin: Before issuing any of the RACF commands in the following steps, make sure you are using an MVS ID that:

1. Has the authority to use the RACDCERT command in RACF (for example, SPECIAL authority).

   For details about RACDCERT, see *z/OS Security Server RACF Security Administrator's Guide*, SA22-7683. To access this book on the Web, go to the z/OS Book server Web site at URL:

   ```
   http://www.ibm.com/servers/eserver/zseries/zos/bkserv/
   ```
2. Has been defined as a WebSphere for z/OS administrator for the Java server to which the certificates will apply. (Use the Administrators dialog in the Administrative Console to give an MVS ID administrative authority over a Java server.) See *WebSphere Application Server for z/OS V5.0: Operations and Administration*, SA22-7912 for more information. To access this book on the Web, go to the product library page at URL:

   ```
   http://www-3.ibm.com/software/webservers/appserv/zos_os390/library.html
   ```

Perform these steps to set up secure connections using self-signed CA certificates:

1. Create a self-signed CA certificate. In this step you will:
   a. Create a self-signed CA certificate with label **CA certificate for ASCR1**.
   b. Create an ASCII z/OS or OS/390 data set, **CERT.ARM**, which contains your CA public-private key pair and self-signed CA certificate.

   Example: To create your self-signed CA certificate, issue the RACF command:

   ```
   RADCERT CERTAUTH GENCERT SUBJECTSDN(CN('IBM Raleigh Webapps CA')
   o('IBM Webapps Raleigh') ou('IBM Webapps') L('Raleigh') SP('North Carolina')
   C('US')) SIZE(512) WITHLABEL('CA certificate for ASCR1') NOTBEFORE(DATE
   (2002-07-01)) NOTAFTER(DATE(2004-10-12))
   ```

   where
   - The Distinguished Name consists of the:

  - Common name (Domain Name), IBM Raleigh Webapps CA
  - Organization name, IBM Webapps Raleigh
  - Optional organizational unit, IBM Webapps
  - Optional city or locality, Raleigh
  - Optional state or province, North Carolina
  - Country code, US
- **CERTAUTH** indicates that a CA certificate is being generated.
- 512 is the key size
- **CA certificate for ASCR1** is the label of the CA certificate.
- NOTBEFORE(DATE(2002–07–01)) NOTAFTER(DATE(2004–10–12)) indicates that the certificate is valid from July 1, 2002 through October 12, 2004.

Example: To export the CA certificate to an MVS data set so that in Step 8 the CA can be added to the list of trusted CAs on the browser, issue the following command:

```
RACDCERT CERTAUTH EXPORT(LABEL('CA certificate for ASCR1')) DSN(CERT.ARM)
FORMAT(CERTB64)
```

where:

- **CERTAUTH** indicates that a CA certificate is being exported.
- **CA certificate for ASCR1** is the label of the CA certificate
- **CERT.ARM** is the data set that will contain the CA certificate
- **CERTB64** indicates that the CA certificate is saved to the data set in BASE 64 encoded ASCII.

---

2. Create an SSL RACF controller key ring and connect your CA certificate to that key ring.

   Example: To create an SSL RACF controller key ring, and connect it to the CA certificate, issue the following commands:

   ```
   RACDCERT ID(ASCR1) ADDRING(CRRING)
   RACDCERT ID(ASCR1) CONNECT(CERTAUTH LABEL('CA certificate for ASCR1')
   RING(CRRING))
   ```

   where:

   - **CERTAUTH** indicates that a CA certificate is being connected.
   - **CA certificate for ASCR1** is the label of the CA certificate
   - **CRRING** is the controller key ring
   - **ASCR1** is the controller ID under which the CRRING key ring resides.

---

3. Create and sign your server certificate.

   In Step 1, you set up your CA environment which enables you to act as your own CA and sign certificates. A signed server certificate is required before clients can establish an SSL connection to your Java server controller. Because you are acting as your own CA, you will sign the server certificate that you create in this step. If you were using an external commercial CA, such as VeriSign, you would send the server certificate request to the CA for signature.

   In this step you will create a server certificate with label **Certificate for ASCR1** signed by the internal CA using label **Certificate for ASCR1**.

Example: To create the server certificate signed by the internal CA, issue the following command:

```
RACDCERT ID(ASCR1) GENCERT SUBJECTSDN(CN('IBM Raleigh Webapps')
O('IBM Webapps Raleigh')  OU('IBM Webapps')  L('Raleigh')
SP('North Carolina')  C('US')) SIZE(512) WITHLABEL('Certificate
for ASCR1') SIGNWITH(CERTAUTH LABEL('(Certificate for ASCR1'))
```

where

- The Distinguished Name consists of the:
  - Common name (Domain Name), **IBM Raleigh Webapps**
  - Organization name, **IBM Webapps Raleigh**
  - Optional organizational unit, **IBM Webapps**
  - Optional city or locality, **Raleigh**
  - Optional state or province, **North Carolina**
  - Country code, **US**
- **ASCR1** is the controller ID under which the server certificate is created.
- 512 is the key size
- **Certificate for ASCR1** is the label of the server certificate request.
- **CA certificate for ASCR1** is the label of the CA certificate that is used to sign the server certificate.

---

4. Connect your signed server certificate to the controller key ring. In this step you will:
   - Connect the server certificate with label **Certificate for ASCR1** to the controller key ring.
   - Ensure the certificate will be the default in this key ring.

   Example: To connect the server certificate to the controller key ring **CRRING**, and make this server certificate the default certificate in this key ring, issue the following command:

   ```
   RACDCERT ID (ASCR1) CONNECT(ID(ASCR1) LABEL('Certificate for ASCR1')
   RING(CRRING) DEFAULT)
   ```

   where:
   - **ASCR1** is the controller's ID under which this key ring and certificate reside.
   - **CRRING** is the controller key ring
   - **Certificate for ASCR1** is the label that identifies the key and server certificate in the key ring.
   - **DEFAULT** makes the server certificate the default in this key ring.

     Note: DEFAULT **must** be included on this command.

---

5. Using the TSO/E OPUT command in MVS, copy the MVS data set containing your server certificate to your document root directory in the HFS.In step 8, you will add the certificate to the list of trusted CAs in your browser.

   Example: To copy the MVS data set CERT.ARM from MVS to your document root directory, issue the following TSO/E OPUT command:

   ```
   oput 'USER1.CERT.ARM' '/usr/lpp/WebSphere/mydoc/cert.arm'
   ```

Note: You can execute this TSO/E command from TSO/E, ISPF option 6, and the shell. To find out more about this command, including where to execute it, please see the *z/OS UNIX System Services Command Reference*, SA22-7802. To access this book on the Web, go to the z/OS Book server Web site at URL:

`http://www.ibm.com/servers/eserver/zseries/zos/bkserv/`

---

6. Permit the controller ID to access the key ring through DIGTCERT.

   In this step you will permit the controller user ID **ASCR1** to access the key ring through the **DIGTCERT** general resource class.

   This ID must have access to the key ring that was created using RACDCERT. If the ID does not have access, SSL initialization fails. To permit ASCR1 to access the controller key ring, issue RACF commands to perform the following tasks:

   - Define the IRR.DIGTCERT.LIST and IRR.DIGTCERT.LISTRING resources with universal access of None.
   - Permit the ASCR1 ID read access to the IRR.DIGTCERT.LIST and IRR.DIGTCERT.LISTRING resources in the FACILITY class.
   - Activate the FACILITY general resource class.
   - Refresh the FACILITY general resource class.

   Example: To perform the preceding tasks, issue the following commands:

   ```
   RDEFINE FACILITY IRR.DIGTCERT.LIST UACC(NONE)
   PE IRR.DIGTCERT.LIST CLASS(FACILITY) ID (ASCR1) ACCESS(READ)

   RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(NONE)
   PE IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID (ASCR1) ACCESS(READ)

   SETR CLASSACT(FACILITY)
   SETR RACLIST(FACILITY) REFRESH
   ```

   To find out more about controlling access to the RACDCERT function through the FACILITY general resource class, see the *z/OS Security Server RACF Security Administrator's Guide*, SA22-7683. To access this book on the Web, go to the z/OS Book server Web site at URL:

   `http://www.ibm.com/servers/eserver/zseries/zos/bkserv/`

---

7. Register the controller key ring with the Java server.
   a. Open the Administrative Console.
   b. Click **Servers** > **Application Servers** on the left-hand navigation tree.
   c. Click on the name of the server.
   d. On the **Additional Properties** menu of the Server panel, click **Web Container** .
   e. On the **Additional Properties** menu of the Web Container panel, click **HTTP Transport**.
   f. Click on the **Host** you want to configure.
   g. Enter the **Port** number you want to bind..
   h. Click the checkbox to **Enable SSL**.
   i. Select the SSL alias in which you specified the controller key ring.
   j. Click **OK**.

8. Verify that you can establish a secure connection with the controller.

   To verify that you can establish a secure connection with the controller, make sure the Java server is running, and then point your browser at the following URL:

   `https://domain:port_number/directory/webapp_name`

   where:

   domain
   is the domain where the Web application being requested resides.

   port_number
   is the port number specified in step 7.

   directory
   is the directory that contains the application.

   webapp_name
   is the name of the certificate protected Web application being requested.

   Example: :

   ```
   https://www.raleigh.ibm.com:443/webap1/my.jsp
   ```

   The first time you enter this URL, you should receive a warning that the CA certificate is not trusted. You will then be prompted to accept the certificate for the current request and all future requests. If you accept the certificate, it will be added the the browser's list of trusted CA certificates. The next time you enter this URL, you should not receive the warning.

9. Optionally, set up client authentication.

   For instructions, see "Steps for setting up secure HTTPS internal transport connections using client certificates signed by an internal CA" or "Steps for setting up secure HTTPS internal transport connections using client certificates signed by an external CA" on page 181.

Steps for setting up secure HTTPS internal transport connections using client certificates signed by an internal CA: Using SSL, WebSphere for z/OS allows you to set up client authentication using client certificates signed by an internal CA. Using an internal CA to sign your client certificates is independent of whether you used an internal or external CA to sign your server certificate.

Before you begin:

- Before issuing any of the RACF commands in the following steps, make sure you are using an MVS ID that:
  1. Has the authority to use the RACDCERT command in RACF (for example, SPECIAL authority).

     For details about RACDCERT, see *z/OS Security Server RACF Security Administrator's Guide*, SA22-7683. To access this book on the Web, go to the z/OS Book server Web site at URL:

     `http://www.ibm.com/servers/eserver/zseries/zos/bkserv/`

  2. Has been defined as a WebSphere for z/OS administrator for the Java server to which the certificates will apply. (Use the Administrators dialog in the Administrative Console to give an MVS ID administrative authority over a Java server.) See *WebSphere Application Server for z/OS V5.0: Operations and*

Administration, SA22-7912 for more information. To access this book on the Web, go to the product library page at URL:

`http://www-3.ibm.com/software/webservers/appserv/zos_os390/library.html`

- You must set up secure connections using one of the following processes:
  - "Steps for setting up secure HTTPS internal transport connections using a server certificate signed by an internal CA" on page 166
  - "Steps for setting up secure HTTPS internal transport connections using server certificates signed by an external CA" on page 175
- You must ensure that the internal CA that signs your client certificates is marked with a status of TRUST and that it is connected to your controller key ring. During the SSL handshake, the controller tells the client which CAs it trusts based on the trusted CAs in the controller key ring. The browser then searches its client certificates for ones issued by these CAs and allows the user to choose which client certificate to send to the controller.

  If you created and signed your server certificate using the process described in "Steps for setting up secure HTTPS internal transport connections using a server certificate signed by an internal CA" on page 166, you can use the internal CA defined in that example for the internal CA in this process. If you created and signed your server certificate using the process described in "Steps for setting up secure HTTPS internal transport connections using server certificates signed by an external CA" on page 175, you must set up an internal CA as described in Step 1 of "Steps for setting up secure HTTPS internal transport connections using a server certificate signed by an internal CA" on page 166 and connect the CA certificate to the controller key ring as described in Step 2 of that process before proceeding.

Perform these steps to set up client authentication using client certificates signed by an internal CA:

1. Using the WebSphere for z/OS Administrative Console, open a conversation for the appropriate Java server, and verify that SSL client certificates are allowed:
   a. Select Conversations → the name of the conversation → cells → cell name → Java Servers → the name of the appropriate Java server → modify.
   b. In the properties form, check the SSL Client Certificates box, if it is not already checked to indicate that SSL client certificates are allowed.

---

2. Ensure the CA certificate is in the client's browser, if this is a requirement for the client's browser. (Some browsers do not require the CA certificate to reside in the browser.) The following example assumes it is not already there.

   Example: Assuming that:
   - Your CA certificate must be added to the client's browser.
   - You are using the same CA certificate you created in section "Steps for setting up secure HTTPS internal transport connections using a server certificate signed by an internal CA" on page 166 to sign client certificates, as well as your server certificate. (This certificate resides in the data set CERT.ARM.)

   Download the CA certificate:
   a. Make sure the Java server is running with SSL initialized.
   b. Open a browser.
   c. Enter the following URL:

      `https://www.raleigh.ibm.com:443/cert.arm`

- **www.raleigh.ibm.com** is the fully qualified host name of the server.
- **443** is the port on which the HTTPS internal transport is listening.
- **cert.arm** is the CA certificate data set.

Follow the browser prompts to install the CA certificate. Newer versions of the Netscape and Microsoft Internet Explorer browsers automatically start a wizard to help you install the certificate. If you use Microsoft Internet Explorer, you may need to open the file rather than saving it to disk to start the wizard. See the online help in your browser or browser documentation for additional information. Generally for Netscape, if you receive a window asking if you would like to accept the CA to certify network sites, electronic mail (e-mail) users, and software developers, do so.

---

3. Create the client certificate and associate it with a RACF user ID.

   Assuming your CA certificate was added to the list of trusted CAs in your browser, generate a client certificate under a RACF user ID. This enables the client certificate to be used to authenticate the user ID.

   Example: In this example, you will create the client certificate with label **Certificate for Jane Smith** signed by the internal CA using label **CA certificate for ASCR1**. The client certificate will be created under user ID **JSMITH**.

   To create the client certificate signed by the internal CA, issue the RACF command:

   ```
   RADCERT ID(JSMITH) GENCERT SUBJECTSDN(CN('Jane Smith')
   o('IBM ID Raleigh') ou('IBM ID z/OS') L('Raleigh') SP('North Carolina')
   C('US')) SIZE(512) WITHLABEL('Certificate for Jane Smith')
   SIGNWITH(CERTAUTH LABEL('CA certificate for ASCR1'))
   ```

   where

   - The Distinguished Name consists of the:
     - Common name (Domain Name), **Jane Smith**
     - Organization name, **IBM ID Raleigh**
     - Optional organizational unit, **IBM ID z/OS**
     - Optional city or locality, **Raleigh**
     - Optional state or province, **North Carolina**
     - Country code, **US**
   - **JSMITH** is the z/OS user ID under which the client certificate is to be added.
   - 512 is the key size
   - **Certificate for Jane Smith** is the label of the client certificate.
   - **CA certificate for ASCR1** is the label of the CA certificate that will sign the client certificate.

   The client certificate will be created with status **TRUST**. Trust indicates that the client certificate can be used to authenticate the user ID **JSMITH**.

4. Add the signed client certificate to the client's browser. To perform this step, you must:
   - Export the client certificate to a z/OS data set.

     Example: To export the client certificate to a data set so that the client certificate can be added to the client's browser, issue the following command:

```
RACDCERT ID(JSMITH) EXPORT(LABEL('Certificate for Jane Smith'))
DSN('JSMITH.CLIENT.P12') FORMAT(PKCS12DER) PASSWORD('Test')
```

where:

- **JSMITH** is the user ID associated with the client certificate being exported.
- **Certificate for Jane Smith** is the label of the client certificate.
- **'JSITH.CLIENT1.P12'** is the data set that will contain the client certificate.
- **PKCS12DER** indicates that the client certificate and private key are DER encoded when saved to the data set.
- **Test** is the password associated with the encrypted client certificate. You will be required to provide this password when you import the client certificate into the browser. The password is case sensitive.

- FTP the client certificate to the client's workstation.

  Example: This example shows how to use the FTP command to transfer the PKCS12 data set containing the signed client certificate to the client's workstation. The following steps are performed on the workstation:

  a. Enter the FTP command and the host name or IP address of the controller. For example:

     ```
     ftp www.raleigh.ibm.com
     ```

  b. When prompted, enter your user ID and password.

  c. Enter **bin** to transfer the file in binary format.

  d. Transfer the file to the workstation by entering:

     ```
     get 'JSMITH.CLIENT1.P12' client1.p12
     ```

  e. Enter **quit** or **bye** to exit.

- Load the client certificate into the client's browser.

  Example: This example shows how to load the PKCS12 file into the Netscape Communicator browser:

  a. Start the browser.

  b. To access the security information, click **Communicator, Tools, Security, Info**.

  c. Under **Certificates**, click **Yours**.

  d. Click **Import a Certificate**. You may need to scroll down to see this option.

  e. Highlight the PKCS12 file.

  f. Click **Open**, and enter the case sensitive password protecting the file.

  g. Click **OK**. The following messages will be displayed:

     ```
     Your certificates have been successfully imported.
     ```

  h. Click OK. You should be able to see this certificate label in the window called **These are your certificates**. You may need to scroll down to find the label.

     Note: On browser versions prior to Netscape 4.6.1, there may be a problem displaying the label. For example, the label name may appear as ????@????.

---

5. Verify that the client can use the client certificate to access a protected page.

To verify that the client can establish a secure connection with a protected page, make sure the Java server is running, and point your browser at the following URL:

```
https://domain:port_number/directory/webapp_name
```

where:

domain
is the domain where the Web application being requested resides.

port_number
is the port number specified for the BBOC_HTTP_SSL_PORT WebSphere variable in step 7.

directory
is the directory that contains the application.

webapp_name
is the name of the certificate protected Web application being requested.

Example: :

```
https://www.raleigh.ibm.com:443/webap1/my.jsp
```

When prompted by the browser, select the label for the client certificate. If the setup is correct, you will be able to view the protected page without prompts for a user ID and password.

Steps for setting up secure HTTPS internal transport connections using server certificates signed by an external CA: Using SSL, WebSphere for z/OS allows you to use an external Commercial CA to sign your server certificate.

Notes:

1. You must use System SSL to establish secure connections. To use System SSL with the HTTPS internal transport, the System SSL load library must exist in linklist and must be under program control. If you have not already done so:
   - Add the load library to the linklist.
   - Turn on program control for the library by issuing the following RACF commands from a user ID that has the proper authority:

     ```
     RALTER PROGRAM * ADDMEM('hlq.SGSKLOAD'//NOPADCHK) UACC(READ)
     SETROPTS WHEN(PROGRAM) REFRESH
     ```

     If turning on program control for the first time, use the RDEFINE command instead of the RALTER command.
2. You will issue the RACF command, RACDCERT, to create certificates and key rings for your Java server. On most of the RACDCERT commands you must specify a user ID. This ID must be the same user ID as the controller ID for your server. If it is not, SSL will not initialize. The following example uses ASCR1 as the controller ID. Therefore, ASCR1 is specified as the ID on the RACCERT commands in this example.

Before you begin: Before issuing any of the RACF commands in the following steps, make sure you are using an MVS ID that:

1. Has the authority to use the RACDCERT command in RACF (for example, SPECIAL authority).

   For details about RACDCERT, see *z/OS Security Server RACF Security Administrator's Guide*, SA22-7683. To access this book on the Web, go to the z/OS Book server Web site at URL:

http://www.ibm.com/servers/eserver/zseries/zos/bkserv/

2. Has been defined as a WebSphere for z/OS administrator for the Java server to which the certificates will apply. (Use the Administrators dialog in the Administrative Console to give an MVS ID administrative authority over a Java server.) See *WebSphere Application Server V4.0.1 for z/OS and OS/390: Operations and Administration* for more information. To access this book on the Web, go to the product library page at URL:

   http://www-3.ibm.com/software/webservers/appserv/zos_os390/library.html

Perform these steps to set up secure connections using server certificates signed by an external CA:

1. Create the controller key ring.

   A key ring is required for each controller that clients connect to using a secure SSL connection.

   Example: To create the controller key ring, CRRING, issue the following RACF command:

   ```
   RACDCERT ID(ASCR1) ADDRING(CRRING)
   ```

2. Create a server certificate request. In this step you will:

   a. Create a self-signed certificate in order to establish your common name (host name) and public-private key pair.

      Example: To create the self-signed CA certificate for the controller with user ID ASCR1, issue the following RACF command:

      ```
      RADCERT ID(ASCR1) GENCERT SUBJECTSDN(CN('ASCR1.Raleigh.ibm.com')
      o('IBM ID Raleigh') ou('IBM ID z/OS') L('Raleigh') SP('North Carolina')
      C('US')) SIZE(512) WITHLABEL('certificate for ASCR1')
      ```

      where

      - The Distinguished Name consists of the:
        - Common name (Domain Name), **ASCR1.Raleigh.ibm.com**
        - Organization name, **IBM ID Raleigh**
        - Optional organizational unit, **IBM ID z/OS**
        - Optional city or locality, **Raleigh**
        - Optional state or province, **North Carolina**
        - Country code, **US**
      - **ASCR1** is the controller ID under which the server certificate resides.
      - 512 is the key size.
      - **Certificate for ASCR1** is the label of the server certificate request.

   b. Generate a server certificate request from the self-signed certificate and save it to a data set.

      Example: To generate a server certificate request and save it to a data set, issue the following RACF command:

      ```
      RACDCERT ID(ASCR1) GENREQ(LABEL('Certificate for ASCR1')) DSN(CERTREQ.ARM)
      ```

      where

      - **ASCR1** is the controller ID under which the server certificate resides.
      - **Certificate for ASCR1** is the label of the server certificate request.
      - **CERTREQ.ARM** is the data set which contains your public-private key pair and unsigned server certificate.

The self-signed certificate that you create will contain the controller's common name and public-private key pair. This information is required in order to generate the certificate request and obtain a server certificate signed by your external CA.

---

3. Transfer the server certificate request to your workstation.

   Example: To use the File Transfer Protocol (FTP) command to transfer the CERTREQ.ARM data set containing the server certificate request to your workstation, perform the following steps from your workstation:

   a. From a DOS prompt line, enter an FTP command specifying either the host name or IP address of the controller. For example:

      ```
      ftp www.raleigh.ibm.com
      ```

   b. When prompted, enter your user ID and password.

   c. Change to the directory where you put the data set containing the server certificate request, CERTREQ.ARM. For example, if the data set resides in the directory USER1, enter:

      ```
      cd 'USER1'
      ```

   d. Transfer the file in ASCII format to the workstation by entering:

      ```
      get certreq.arm
      ```

   e. Enter **quit** or **bye** to exit the FTP command process.

---

4. Send the request to a CA to be signed, using the CA's instructions for sending certificate requests and receiving signed server certificates.

---

5. Ensure that your CA certificate is in the RACF list of CA certificates and marked with a status of **TRUST**. These conditions must be met before you can receive the CA-signed certificated into your controller key ring.

   By default, the following CAs are designated in RACF, with a status of **NO TRUST**. Before you can use one of these CAs, you must first mark the CA with a status of **TRUST**:

   - Integration Certification Authority Root
   - IBM World Registry Certification Authority
   - Thawte Personal Premium CA
   - Thawte Personal Freemail CA
   - Thawte Personal Basic CA
   - Thawte Premium server CA
   - Thawte server CA
   - RSA Secure server Certification Authority
   - Verisign Class 1 Public Primary Certification Authority
   - Verisign Class 2 Public Primary Certification Authority
   - Verisign Class 3 Public Primary Certification Authority

   In this step you must:

   - Check whether your external CA is in the list of CAs in RACF and whether it is marked with a status of **TRUST**.
     a. If the CA certificate is not already in the list of CAs, add it to the list and mark it with a status of **TRUST**.

b. If the CA certificate is already in the list of CAs, but has a status of **NO TRUST**, change the status to **TRUST**.

- Connect the CA certificate to your **CRRING** key ring.

To check the list of CAs, issue the following RACF command:

```
RACDCERT CERTAUTH LIST
```

After receiving the CA certificate, add it to RACF with **TRUST** status.

For example: Issue the following RACF command:

```
RACDCERT CERTAUTH ADD(CERT1.ARM) TRUST WITHLABEL('CA cert for ASCR1')
```

where:

- **CERTAUTH** indicates the type of certificate you are adding to RACF, in this case, a certificate authority certificate.
- **CERT1.ARM** is the data set containing the CA certificate.
- **CA cert for ASCR1** is the label for the CA certificate.

If your CA is in the list of CAs in RACF, but has a current status of **NO TRUST**, then change the status to **TRUST**. For example, issue the following command:

```
RACDCERT CERTAUTH ALTER(LABEL('CA cert for ASCR1')) TRUST
```

where:

- **CERTAUTH** indicates the type of certificate you are adding to RACF, in this case, a certificate authority certificate.
- **CA cert for ASCR1** is the label for the CA certificate.

Now that your CA certificate is in the RACF list of CA certificates and has a status of trust, connect the CA certificate to your CRRING key ring. To connect the CA certificate to the key ring, issue the following RACF command:

```
RACDCERT ID(ASCR1) CONNECT(CERTAUTH LABEL('CA cert for ASCR1') RING(CRRING)
```

where:

- **ASCR1** is the controller ID under which the key ring resides.
- **CA cert for ASCR1** is the label for the CA certificate.
- **CERTAUTH** indicates the type of certificate you are adding to RACF, in this case, a certificate authority certificate.

---

6. Add your server certificate to the controller key ring.

   In this step, you will:

   - Alter the server certificate if necessary to make it look like the example.
   - Put the server certificate in an MVS data set, **CRCERT.ARM**.
   - Add the server certificate to RACF and associate it with the **ASCR1** ID.

   Before you can add the signed server certificate to your controller key ring, you must put this certificate in an MVS data set. The certificate data set you create in this example is **CRCERT.ARM**. Alter the certificate data set if necessary. Include the BEGIN CERTIFICATE and END CERTIFICATE lines and all data in between as shown in the following example. If your certificate

contains additional information before BEGIN CERTIFICATE or after END CERTIFICATE , remove all of the extraneous information in the file.

```
-----BEGIN CERTIFICATE-----
MIIB0DCCATkCBDV8PgswDQYJKoZIhvcNAQECBQAwcjELMAkGA1UEBhMCVVMxDTAL
BgNVBAgTBE4uQy4xDDAKBgNVBAcTA1JUUDEMMAoGA1UEChMDSUJNMRcwFQYDVQQL
Ew5XZWJzZXJ2ZXIgVGVzdDEfMB0GA1UEAxMWbXZzMTY3LnJhbGVpZ2guaWJtLmNv
bTAaFws5ODA2MDgxOTM5WhcLOTkwNjA4MTkzOVowNDELMAkGA1UEBhMCVVMxDDAK
BgNVBAoTA0lCTTEXMBUGA1UEAxMOcGtjczEwLmlibS5jb20wXDANBgkqhkiG9w0B
AQEFAANLADBIAkEA1IYGldVmnKAI8hJQGT074oXTD0Tb+jFN8wkPqc+DVhYix1fj
h/sbiuDZF66BMh5hnHfJr75633CgjW10EpID0wIDAQABMA0GCSqGSIb3DQEBAgUA
A4GBAF1KVppAM7Gh2F9BBIY/jPMFlRp8+HAAVkK29Q4DxeF2FrTzQutKmO8duCWv
xnJo4pgl5Uj29DSAsrX8mULfczyuZwVVXiCGnhN03pYj8bbQjo0edqQ7hYsRl3P4
C72I+yRwtWUukfVgwdo0mWXyEclx7eT5jsW4weVEqWvuht8j
-----END CERTIFICATE-----
```

This example assumes that you received the server certificate on your workstation, and need to FTP this certificate, in ASCII format, to a z/OS or OS/390 data set. The following steps are performed on the workstation:

a. Enter the FTP command and the host name or IP address of the controller. For example:

   ```
   ftp www.raleigh.ibm.com
   ```

b. When prompted, enter your user ID and password.

c. Transfer the file to the z/OS or OS/390 data set that will be created on execution of the PUT command by entering:

   ```
   put crcert.arm 'CRCERT.ARM'
   ```

d. Type **quit** or **bye** to exit.

Issue the following RACF command to add the server certificate signed by your external CA to RACF and associate it with the ASCR1 ID. In doing so, you will replace the self-signed certificate created in Step 2:

```
RACDCERT ID(ASCR1) ADD(CRCERT.ARM) WITHLABEL('Certificate for ASCR1')
```

where:

- **CRCERT.ARM** is the server certificate data set.
- **Certificate for ASCR1** is the label of the server certificate.

---

7. Connect your signed server certificate that is now in RACF to your CRRING key ring and make this certificate the default certificate in the key ring. For example, issue the following RACF command:

   ```
   RACDCERT ID(ASCR1) CONNECT(ID(ASCR1) LABEL('Certificate for ASCR1')
   RING(CRRING) DEFAULT)
   ```

   where:

   - **ASCR1** is the controller ID under which the controller key ring and the server certificate reside.
   - **CRRING** is the control key ring.
   - **Certificate for ASCR1** is the label that identifies the key and server certificate in the key ring.
   - **DEFAULT** makes the server certificate the default in the key ring.

---

8. Permit the controller ID to access the key ring through DIGTCERT general resource class.

The controller ID must have access to the key ring created using RACDCERT. If the ID does not have access, SSL initialization fails. In this example the controller ID is ASCR1. To permit ASCR1 to access the controller key ring, you issue RACF commands to perform the following tasks:

- Define the IRR.DIGTCERT.LIST and IRR.DIGTCERT.LISTRING resources with universal access of None.
- Permit the ASCR1 ID read access to the IRR.DIGTCERT.LIST and IRR.DIGTCERT.LISTRING resources in the FACILITY class.
- Activate the FACILITY general resource class.
- Refresh the FACILITY general resource class.

To perform these tasks, issue the following commands:

```
RDEFINE FACILITY IRR.DIGTCERT.LIST UACC(NONE)
PE IRR.DIGTCERT.LIST CLASS(FACILITY) ID(ASCR1) ACCESS(READ)

RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(NONE)
PE IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(ASCR1) ACCESS(READ)

SETR CLASSACT(FACILITY)
SETR RACLIST(FACILITY) REFRESH
```

To find out more about controlling access to the RACDCERT function through the FACILITY general resource class, see the description of the RACDCERT command in the *z/OS Security Server RACF Security Administrator's Guide*, SA22-7683. To access this book on the Web, go to the z/OS Book server Web site at URL:

```
http://www.ibm.com/servers/eserver/zseries/zos/bkserv/
```

---

9. Register your controller key ring with the Java server:
   a. Open the Administrative Console.
   b. Click **Servers** > **Application Servers** on the left-hand navigation tree.
   c. Click on the name of the server.
   d. On the **Additional Properties** menu of the Server panel, click **Web Container** .
   e. On the **Additional Properties** menu of the Web Container panel, click **HTTP Transport**.
   f. Click on the **Host** you want to configure.
   g. Enter the **Port** number you want to bind..
   h. Click the checkbox to **Enable SSL**.
   i. Select the SSL alias in which you specified the controller key ring.
   j. Click **OK**.

---

10. Verify that you can establish a secure connection with the controller.

    To verify that you can establish a secure connection with the controller, make sure the Java server is running, and point your browser at the following URL:

    ```
    https://domain:port_number/directory/webapp_name
    ```

    where:

    domain
    is the domain where the Web application being requested resides.

port_number
is the port number specified for the BBOC_HTTP_SSL_PORT WebSphere variable in step 7.

directory
is the directory that contains the application.

webapp_name
is the name of the certificate protected Web application being requested.

Example: :

```
https://www.raleigh.ibm.com:443/webap1/my.jsp
```

---

11. Optionally, set up client authentication.

    For instructions, see "Steps for setting up secure HTTPS internal transport connections using client certificates signed by an internal CA" on page 171 or "Steps for setting up secure HTTPS internal transport connections using client certificates signed by an external CA".

Steps for setting up secure HTTPS internal transport connections using client certificates signed by an external CA: WebSphere for z/OS allows you to set up client authentication using client certificates that are signed by an external CA. Using an external CA to sign your client certificates is independent of whether you used an internal or external CA to sign your server certificate. However, before using this example, there are a few things you must do first.

- You must set up secure connections by following the instructions in one of the following sections:
  - "Steps for setting up secure HTTPS internal transport connections using a server certificate signed by an internal CA" on page 166
  - "Steps for setting up secure HTTPS internal transport connections using server certificates signed by an external CA" on page 175
- You must ensure that the external CA that signs your client certificates is marked with a status of TRUST and that it is connected to your controller key ring. During the SSL handshake, the controller tells the client which CAs it trusts based on the trusted CAs in the controller key ring. The browser then searches its client certificates for ones issued by these CAs and allows the user to choose which client certificate to send to the controller.
- If you created and signed your server certificate using the steps in the section "Steps for setting up secure HTTPS internal transport connections using server certificates signed by an external CA" on page 175, you can use the external CA defined in that example for the external CA in this example.
- If you created and signed your server certificate using the steps in the section "Steps for setting up secure HTTPS internal transport connections using a server certificate signed by an internal CA" on page 166, you must ensure that your CA certificate is in RACF and marked with a status of TRUST, and that the external CA is connected to the controller key ring. (See "Steps for setting up secure HTTPS internal transport connections using server certificates signed by an external CA" on page 175 for a description of how to do this.)

Notes:

1. Choose the environment in which you execute the RACF commands (TSO READY, ISPF option 6, and so forth). Implementing these commands varies

from one environment to another. See your local RACF administrator for assistance, or review the appropriate books for your environment.

Before you begin: Before issuing any of the RACF commands in the following steps, make sure you are using an MVS ID that:

1. Has the authority to use the RACDCERT command in RACF (for example, SPECIAL authority).

   For details about RACDCERT, see *z/OS Security Server RACF Security Administrator's Guide*, SA22-7683. To access this book on the Web, go to the z/OS Book server Web site at URL:

   `http://www.ibm.com/servers/eserver/zseries/zos/bkserv/`
2. Has been defined as a WebSphere for z/OS administrator for the Java server to which the certificates will apply. (Use the Administrators dialog in the Administrative Console to give an MVS ID administrative authority over a Java server.) See *WebSphere Application Server for z/OS V5.0: Operations and Administration*, SA22-7912 for more information. To access this book on the Web, go to the product library page at URL:

   `http://www-3.ibm.com/software/webservers/appserv/zos_os390/library.html`

Perform these steps to set up client authentication using client certificates signed by an external CA:

1. Verify that SSL client certificates are allowed.
   a. Open the Administrative Console.
   b. Click on **Security** > **Authentication Protocols** > **CSIv2 Inbound Authentication** in the left-hand navigation tree.
   c. In the General Properties section of the Configuration Tab for the CSIv2 Inbound Authentication panel, check to see if either **Supported** or **Required** is checked for **Client Certificate Authentication**.

   ---

2. Ensure the CA certificate is in the client's browser. Browsers vary as to whether they require the CA certificate to be in the browser. This example assumes you will ensure that your CA certificate is added to your browser if it is not already there.

   Example: If you are using the same external CA certificate to sign client certificates that you used to sign your server certificate, then clients may have already loaded the CA certificate into their browsers in step 5 of section "Steps for setting up secure HTTPS internal transport connections using server certificates signed by an external CA" on page 175. If they have not, they should contact the external CA to obtain the CA certificate.

   ---

3. Obtain client certificate. Follow the external CA's instructions for obtaining the signed client certificate and loading it into your browser.

   ---

4. Map a client certificate to a RACF user ID. RACF maps client certificates that are in various formats, including PKCS12, binary, and base 64 encoded ASCII. Some browsers may be able to output the client certificate in these formats or other formats. If, for instance, either the binary or base 64 encoded ASCII format is outputted, the resulting file would contain the client certificate without the private key. If the PKCS12 format is outputted, the resulting file would contain the private key (which RACF doesn't use) and the client certificate. If a browser does not output the client certificate in the format that

you want, contact the signing authority to obtain the client certificate in the desired format. The format of the client certificate in RACF can be different from the format of the client certificate in the browser.

Since some browsers require client certificates in PKCS12 format, we will map a PKCS12 formatted certificate to a RACF user ID. You can export the client certificate from the browser so that it can be input to RACF.

You can use the following Resource Access Control Facility (RACF) options to map a client certificate to RACF when the client certificate is created with some method other than RACF commands or a RACF application such as PKISERV:

- **Certificate Name Filtering function:** This function is available for OS/390 Release 10 and later.
- **Automatic registration of digital certificates on the Web**: The Autoregistration Web application enables a client to automatically register a certificate with the controller.
- Using ISPF panels or the RACDCERT command: These two options take the same inputs, but you use panels with ISPF and a command line with RACDCERT.

  For information on these options, see the *z/OS Security Server RACF Security Administrator's Guide*, SA22-7683. To access this book on the Web, go to the z/OS Book server Web site at URL:

  `http://www.ibm.com/servers/eserver/zseries/zos/bkserv/`

You can use one or more of these options. This example shows how to use the RACDCERT command.

In this step you will:

- FTP the PKCS12 formatted client certificate from the workstation to the HFS on your z/OS system.
- Copy the PKCS12 formatted client certificate from the HFS to an MVS data set.
- Issue RACF commands to associate the client certificate with a RACF user ID.

FTP the client certificate from the client's workstation to the HFS: We will FTP the client certificate from the workstation into the HFS in this section and then do an OGET of the file into an MVS data set in the next section to insure that the data set containing the client certificate has the correct format. This example shows how to use the FTP command to transfer the signed PKCS12 formatted client certificate on a client's workstation to the HFS. The following steps are performed on the workstation:

a. Enter the FTP command and the host name or IP address of the controller. For example:

   ```
   ftp www.raleigh.ibm.com
   ```

b. When prompted, enter your user ID and password.

c. Change to the directory where you will place the client certificate. For example:

   ```
   cd /ibm/security/user1
   ```

d. Enter **bin** to transfer the file in binary format.

e. Transfer the file to the HFS by entering:

   ```
   put client1.p12
   ```

f. Type **quit** or **bye** to exit.

Copy the client certificate from the HFS to an MVS data set: You must store client certificates on MVS in variable block (VB) format. The client certificates in this example are in an HFS directory. Use the TSO/E OGET command to move the certificate file from the HFS directory into an MVS sequential data set. If you move the client certificate into a new data set, the OGET command creates a VB sequential data set by default. You must use the OGET command to move the client certificate into the MVS sequential data set; exporting it from the browser to FTP it directly into the MVS data set does not work because the certificate file is not in the correct format.

Example:

```
oget '/ibm/security/user1/client1.p12' 'jsmith.client1.p12' binary
```

Note: You can execute this TSO/E command from TSO/E, ISPF option 6, and the shell. To find out more about this command, including where to execute it, please see the z/OS UNIX System Services Command Reference. To access this book on the Web, go to the z/OS Book server Web site at URL:

```
http://www.ibm.com/servers/eserver/zseries/zos/bkserv/
```

Associate the client certificate with a RACF user ID: To perform the following steps, ensure that you are using an MVS user ID that has the authority to map a client certificate to an MVS user ID:

a. Issue the RACDCERT command to add the client certificate to the RACF data base for each client. For example:

```
RACDCERT ID(RACFID1) ADD('JSMITH.CLIENT1.P12')
WITHLABEL('Certificate for Jane Smith') TRUST PASSWORD('X2RL')
```

- **RACFID1** is the RACF user ID under which the client certificate is added.
- **'JSMITH.CLIENT1.P12'** is the name of the data set where the certificate file is located.
- **TRUST** indicates that you can use the client certificate to authenticate the user ID RACFID1.
- **Certificate for Jane Smith** is the label of the client certificate.
- **X2RL** is the required password for PKCS12 certificates.

b. Issue the following SETROPTS command to refresh the DIGTCERT class for all the clients:

```
SETROPTS RACLIST(DIGTCERT) REFRESH
```

c. Verify that the client certificate is associated with a user ID that has been defined to RACF, by issuing the following RACDCERT command and specifying the user ID in the ID field:

```
RACDCERT ID(RACFID1) LIST
```

If you are logged onto the user ID you are trying to verify, you can issue the RACDCERT command without operands to display the client certificate for that user ID.

---

5. Verify that a client can establish a secure connection with the controller.

   To verify that a client can establish a secure connection with the controller, make sure the Java server is running, and point your browser at the following URL:

   ```
   https://domain:port_number/directory/webapp_name
   ```

where:

domain
is the domain where the Web application being requested resides.

port_number
is the port number specified in step 7.

directory
is the directory that contains the application.

webapp_name
is the name of the certificate protected Web application being requested.

Example: :

```
https://www.raleigh.ibm.com:443/webap1/my.jsp
```

When prompted by the browser, select the label for the client certificate. If the setup is correct, you will be able to view the protected page without prompts for a user ID and password.

Setting up the asserted identity function

Identity assertion is the invocation credential that is asserted to the downstream server. The steps involved in setting up identity assertion depends on the authentication protocol you are using. WebSphere Application Server for z/OS V5 supports the following authentication protocols:

- zSAS
- CSIv2

The information here relates to the zSAS authentication protocol. For information on CSIv2 asserted identity see the article on ″Identity Assertion″ in the ″Security″ section in the WebSphere Application Server InfoCenter, access to which can be obtained through the WebSphere forz/OS library Web site http://www.ibm.com/software/webservers/appserv/zos_os390/library.html.

SSL client certificate support provides a function called asserted identity, in which an intermediate cluster can send the identities of its clients to a target cluster in a secure yet efficient manner. This function requires client certificate support to establish the intermediate cluster as the owner of the SSL session. Through RACF, the system can check that the intermediate cluster can be trusted (special RACF permission is given to the address spaces, such as controllers, that run secure system code). Once trust in this intermediate cluster is established, client identities (MVS user IDs) need not be separately verified by the target cluster; those client identities are simply asserted without requiring authentication.

For the steps for setting up asserted identity for zSAS, refer to “Steps for configuring the zSAS transport protocol” on page 198. For the steps for setting up asserted identity for CSIv2, refer to “Steps for configuring the CSIv2 inbound authentication protocol” on page 192 and “Steps for configuring the CSIv2 outbound authentication protocol” on page 194.

Selecting a Web container security collaborator level

The security functions the Web container can provide is determined by the version of the Web container security collaborator that is specified in the webcontainer.conf file:

- Version 1 of the Web container security collaborator uses a SAF user registry and only provides the following security functions for requests received by the IBM

HTTP cluster for z/OS and forwarded to the Web container via the WebSphere for z/OS Local Redirector plug-in. None of these functions were available for requests received by the HTTP or HTTPS Transport Handlers:
  - Basic authentication
  - Form Based authentication
  - Client Certificates
  - Single Sign-On across WebSphere/390 clusters
- Version 2 of the Web container security collaborator enables the Web container to provide most of these security functions for requests that are received by the HTTP or HTTPS Transport Handler as well as for requests received by the IBM HTTP cluster for z/OS. This version of the collaborator also enables you to use a trust association interceptor with WebSphere for z/OS.

The following table summarizes the capability and configuration requirements for the version 1 and Version 2 web security collaborators.

Table 58. Summary of the two Versions of the Web container security collaborator

	Version 1	**Version 2**
Security functions supported	• Basic authentication • Form Based authentication[1] • Client certificate authentication • Single sign-on authentication across IBM HTTP clusters for z/OS[1]	• Basic authentication • Form Based authentication[2] • Single sign-on across IBM HTTP clusters for z/OS[2] • Trust asssociation interceptor[3]
Security is applied to requests received via	IBM HTTP cluster for z/OS and forwarded to the Web container via the WebSphere for z/OS Local Redirector plug-in.	• HTTPTransport Handler • HTTPS Transport Handler • IBM HTTP cluster for z/OS and forwarded to the Web container via the WebSphere for z/OS Local Redirector plug-in.
Enabled by	Specifying WEB_SECURITY_VERSION=1 in the JVM properties file or by not including a WEB_SECURITY_VERSION property in the JVM properties file (1 is the default value).	Specifying WEB_SECURITY_VERSION=2 in the JVM properties file.

Table 58. Summary of the two Versions of the Web container security collaborator (continued)

Notes:

1. To enable Form Based authentication or single sign-on capability for Web applications being received by the IBM HTTP cluster for z/OS, you must:
   - Set the following properties in the webcontainer.conf file:
     - The **WebAuth.EncryptionKeyLabel** property must specify the label of the cryptographic key that is to be used for Web application security.
     - The **WebAuth.LoginToken.Encrypt** property must be set to true.
   - Grant bpx.surrogat authority to the IBM HTTP cluster for z/OS's address space.
   - Create ICSF keys and grant the IBM HTTP cluster for z/OS's address space access to them.
   - Set the JAVA_PROPAGATE variable In the IBM HTTP cluster for z/OS's httpd.envvars file to NO.
2. To enable Form Based authentication or single sign-on capability for Web applications being received by the HTTP/HTTPS, you must:
   - Set the following properties in the webcontainer.conf file:
     - The **WebAuth.EncryptionKeyLabel** property must specify the label of the cryptographic key that is to be used for Web application security.
     - The **WebAuth.LoginToken.Encrypt** property must be set to true.
   - Add the WEB_SECURITY_VERSION property to the jvm.properties file and set it to 2.
   - Create ICSF keys and make them available to the servant.
   - Permit the servant user ID to the CSFSERV general resource class.
   - Add the ENABLE_TRUSTED_APPLICATIONS environment variable to your Java cluster's current.env. file, and set it to 1.
3. To enable trust association interceptor support, you must:
   - Make the following changes to your Java cluster's current.env file:
     - Add the ENABLE_TRUSTED_APPLICATIONS=1 environment variable.
     - Add tthe TrustAssociationInterceptor class to the CLASSPATH environment variable.
   - Add the following properties to the WebSphere for z/OS webcontainer.conf configuration file:
     - WebAuth.TrustAssociationInterceptor.<value>.ImplClass=<classname>
     - WebAuth.TrustAssociationInterceptor.<value>.Properties=<filename>
   - Add the WEB_SECURITY_VERSION property to the jvm.properties file and set it to 2.

Setting up Kerberos security for WebSphere for z/OS

On WebSphere for z/OS, Kerberos works with SSL to provide a complete authentication mechanism:

- SSL secures the transportation layer to protect messages. SSL also provides the mechanism whereby the client authenticates the server.
- Kerberos provides the mechanism whereby the server authenticates the client. That is, the client sends the server a Kerberos Generic Security Service Application Program Interface (GSS_API) token, which is used by the server to authenticate the identity of the client.
- Through the GSS_API token, a server is able to pass the client's identity to another server in order to satisfy a client's request. This is called delegation.

The following describes how a Kerberos over SSL connection works:

Stage	Description
Negotiation	After the client locates the server, the client and server negotiate the type of security for communications. If Kerberos is to be used, the client is told to connect to a special SSL port.
Handshake	The client connects to the SSL port and the SSL handshake occurs. If successful, SSL message protection begins. The client authenticates the server by inspecting the server's digital certificate.
Client authentication	After the SSL handshake occurs, the client establishes its Kerberos identity and obtains a Kerberos GSS_API token based on this identity and the server's Kerberos principal. The client sends this token to the server along with a unique SSL connection identifier. The server uses the GSS_API token to authenticate the Kerberos principal that represents the client. Once the client has been authenticated, the system uses RACF to obtain the z/OS user ID that has been mapped to the client's Kerberos principal. This z/OS user identity is used in future authorization checks. By default the client constructs the GSS_API token so that delegation is enabled. This will allow the server to impersonate the client on requests made on its behalf. The z/OS user ID, the Kerberos delegated credentials, and the unique SSL connection identifier are stored for use on future requests made over this SSL Kerberos connection. If the Kerberos client authentication, or the mapping of the authenticated principal fails, communication stops.
Ongoing communication	Communication between the client and server use SSL services for message protection. Each message includes the unique SSL connection identifier, which allows the server to match a request to its stored z/OS user ID and Kerberos delegated credentials.

This support requires SSL security to be set up. In addition to SSL requirements, Kerberos requires the following to be installed and configured on your z/OS or OS/390 system:

- OS/390 SecureWay Security Server Network Authentication and Privacy Service for OS/390. For OS/390 V2R8 and V2R9, this support is available through the following Web site:

  `http://www.software.ibm.com`

  For OS/390 V2R10 and z/OS, this support is part of SecureWay Security Server.
- The PTFs for your z/OS or OS/390 system. Consult the PSP bucket for more information.
- The Kerberos security server must be active on the client and server systems where this support is used.
- All z/OS or OS/390 user IDs (for clients and servers) that participate in Kerberos authentication must have a Kerberos RACF segment that defines their Kerberos principal.
- The Kerberos server is not required to have a file that contains its Kerberos secret key. Kerberos on z/OS or OS/390 has eliminated this requirement and can

use the Kerberos principal associated with the current system identity to decrypt the service ticket. WebSphere for z/OS servers must use this feature.

- The WebSphere for z/OS server must have READ access to the IRR.RUSERMAP resource in the RACF FACILITY class.
- Kerberos security relies on time coordination among its participants. The Kerberos security administrator should select a time provider and ensure that participants in Kerberos security use that time source to maintain their system time.

The following table shows the subtasks and associated procedures for defining Kerberos security:

Subtask	Associated procedure (See . . .)
Enabling the Kerberos server	*z/OS Security Server Network Authentication Service Administration*, SC24-5926
Setting up the server for SSL authorization	"Steps for using RACF to authorize the server to use digital certificates" on page 158
Associating the server identity with a Kerberos principal	"Step for associating a server identity with a Kerberos principal"
Defining server attributes for Kerberos	
Setting up a client to use Kerberos **Note:** Kerberos is not supported using the CSiv2 protocol. It is only supported on zSAS.	"Steps for setting up a client to use Kerberos"

Step for associating a server identity with a Kerberos principal

Before you begin: You need to have a RACF user ID established for the server's controller.

You need to install and configure OS/390 SecureWay Security Server Network Authentication and Privacy Service for OS/390 (Kerberos). Enable a SecureWay Security Server (KDC) on each z/OS or OS/390 image where servers will use Kerberos. For more information, see *z/OS Security Server Network Authentication Service Administration*, SC24-5926.

Perform the following step to associated the server identity with a Kerberos principal:

⇔ Issue the ALTUSER command to make the association. **Example:**

```
ALTUSER ctl_ID PASSWORD(new_password) NOEXPIRED
        KERB(KERBNAME(kerberos_principal))
```

where

ctl_ID
Is the user ID assigned to the server's controller through the STARTED class.

new_password
Is the shared z/OS or OS/390 and Kerberos password.

kerberos_principal
Is the Kerberos principal name associated with this z/OS or OS/390 user ID.

You know you are done when the RACF command succeeds.

Steps for setting up a client to use Kerberos

Before you begin: You must have SSL communication set up on your system.

You need to install and configure OS/390 SecureWay Security Server Network Authentication and Privacy Service for OS/390 (Kerberos). Enable a SecureWay Security Server (KDC) on each z/OS or OS/390 image where clients will use Kerberos. For more information, see *z/OS Security Server Network Authentication Service Administration*, SC24-5926.

Perform the following steps to set up a client to use Kerberos.

1. Use RACF to map each z/OS or OS/390 user that will participate as a Kerberos client to a Kerberos principal on the local realm.

   Example:

   ```
   ALTUSER client_ID PASSWORD(WSIVT) NOEXPIRED KERB(KERBNAME(kerberos_principal))
   ```

   where

   client_ID
   Is the client's user ID.

   kerberos_principal
   Is the Kerberos principal name that will be associated with this z/OS or OS/390 user ID.

   Tip: You can use a utility to help a security adminstrator migrate a z/OS or OS/390 RACF registry to Kerberos. The utility is located at the following Web site:

   ```
   http://sandbox.s390.ibm.com/products/racf/kmigrate.html
   ```

   ---

2. Use RACF to set up cross-realm trust relationships between the realms where the target servers reside and the clients reside.

   Example: A client is in Kerberos realm CLIENTREALM and the server is in SERVERREALM:

   ```
   RDEFINE REALM /.../CLIENTREALM/krbtgt/SERVERREALM KERB(PASSWORD(password1))
   RDEFINE REALM /.../SERVERREALM/krbtgt/CLIENTREALM KERB(PASSWORD(password2))
   ```

   where *password1* and *password2* are passwords. These two commands must be issued to each RACF database.

   ---

3. Use RACF to set up foreign user mapping in server realms.

   Examples:

   a. To map all principals from a foreign-realm to a single user ID, issue:

      ```
      RDEFINE KERBLINK /.../foreign_realm APPLDATA('user_ID')
      ```

   b. To map an individual principal from a foreign-realm to a user ID, issue:

      ```
      RDEFINE KERBLINK /.../foreign_realm/principal APPLDATA('user_ID')
      ```

   where

   foreign_realm
   Is the foreign realm.

   user_ID
   Is the MVS user ID.

   principal
   Is the principal.

You know you are done when the RACF commands succeed.

Configuring the authentication protocol

In WebSphere Application Server for z/OS V5, there are two authentication protocols to choose from:

- Common Secure Interoperability Version 2 (CSIv2). The Object Management Group (OMG) has defined a new authentication protocol, called CSIv2, so that vendors can interoperate securely. CSIv2 has been implemented in WebSphere Application Server for z/OS V5 and is considered the strategic protocol.
- IBM z/OS Secure Association Service (zSAS) is the authentication protocol used by all previous releases of WebSphere Application Server for z/OS and thus is maintained for backwards compatibility.

Invoking EJB methods in a secure WebSphere Application Server environment requires an authentication protocol to determine the level of security and type of authentication, which occurs between any given client and server for each request. It is the job of the authentication protocol during a method invocation to coalesce the server authentication requirements (determined by the object IOR) with the client authentication requirements (determined by the client configuration) and come up with an authentication policy specific to that client and server pair.

The authentication policy makes the following decisions, among others, which are all based on the client and server configurations.

- What kind of connection can you make to this server--SSL or TCP/IP?
- If secure sockets layer (SSL) is chosen, how strong is the encryption of the data?
- If SSL is chosen, should the client be authenticated using client certificates?
- Should the client be authenticated using a user ID and password?
- Is there an existing credential to use?
- Should the client identity be asserted to downstream servers?
- Given the configuration of the client and server, should a secure request proceed?

Authentication features include three layers of authentication, which you can use simultaneously:

- Transport layer. The transport layer, the lowest layer, might contain a Secure Socket Layer (SSL) client certificate as the identity.
- Message layer. The message layer might contain a user ID and password.
- Attribute layer. The attribute layer might contain an identity token which is an identity from an upstream server and is already authenticated. The identity layer has the highest priority followed by the message layer and then the transport layer. If a client sends all three, only the identity layer is used. The only way to use the SSL client certificate as the identity is if it is the only information presented during the request. The client picks up the IOR from the name space and reads the values from the tagged component to determine what the server needs for security.

You can configure both protocols (IBM zSAS and CSIv2) to work simultaneously. If a server supports both protocols, it exports an IOR containing tagged components describing the configuration for IBM zSAS and CSIv2. If a client supports both protocols, it reads tagged components for both CSIv2 and IBM zSAS. If the client

supports both and the server supports both, CSIv2 is used. However, if the server supports IBM zSAS (for example, it is a previous WebSphere Application Server for z/OS release) and the client supports both, the client chooses IBM zSAS for this request, since the IBM zSAS protocol is what both have in common. Choose a protocol by specifying the com.ibm.CSI.protocol property on the client side and configure it through the administrative console on the server side. More details are included in the IBM zSAS and CSIv2 properties articles in the InfoCenter.

In the procedures that follow you will configure these authentication protocols for servers:

- CSIv2 Inbound Authentication
- CSIv2 Outbound Authentication
- CSIv2 Inbound Transport
- CSIv2 Outbound Transport
- zSAS Transport

Steps for configuring the CSIv2 inbound authentication protocol

Before you begin: Inbound authentication refers to the configuration that determines the type of accepted authentication for inbound requests. This authentication is advertised in the Interoperable Object Reference (IOR) that the client retrieves from the name server. For more information read the CSIv2 properties articles in the InfoCenter.

You need to start the Administrative Console by specifying URL: http://<server_hostname>:9090/admin.

Perform the following steps to configure the CSIv2 inbound authentication protocol.

1. Click **Security** > **Authentication Protocol** > **CSIv2 Inbound Authentication** in the Navigation tree on the left.

   The CSIv2 Authentication Inbound panel appears.

   Note: This panel will configure the default CSIv2 Inbound Authentication policy for all defined servers. These same values can be defined at the server level by clicking **Servers** > **Application servers** in the left-hand navigation tree and then slecting the server and clicking on **Server Security** > **CSI Authentication** > **Inbound**.

   ---

2. On the General Properties section of the Configuration tab, select the **Basic Authentication** setting if you want clients to authenticate using userids and passwords.

   In the message layer, Basic Authentication (user ID and password) takes place. This type of authentication typically involves sending a user ID and password from the client to the server for authentication.

   Options:

   - **Never** indicates that the server is not configured to accept message layer authentication from any client.
   - **Supported** (default) indicates that this server accepts Basic Authentication. However, other methods of authentication can occur if configured and anonymous requests may be accepted.

- **Required** indicates that only clients which are configured to authenticate to this server through the message layer are allowed to invoke requests on the server.

---

3. Select the **Client Certificate Authentication** setting.

   Specifies if the transport layer, Secure Socket Layer (SSL), client certificate authentication should be performed. Client certificate authentication requires some additional setup steps. These additional steps involve ensuring the server has the signer certificate of each client to which it is connected. If the client uses a certificate authority (CA) to create its personal certificate, then you need only the CA root certificate in the server signer RACF keyring. When the certificate is authenticated to a LocalOS user registry, it must be mapped using the RACF digitial certificate mapping faciity. The identity from client certificates is used for authorization only if no other layer of authentication is presented to the server.

   Options:

   - **Never** indicates that the server is not configured to accept accept certificate authentication from any client.
   - **Supported** (default) indicates that this server accepts SSL client certificate authentication, however, other methods of authentication can occur (if configured) and anonymous requests may be accepted.
   - **Required** indicates that only clients which are configured to authenticate to this server through SSL client certificates are allowed to invoke requests on the server.

---

4. Choose whether or not to enable **Identity Assertion**.

   When enabled, specifies that the server supports identity assertion from downstream servers. Identity assertion is performed in the attribute layer and is only applicable on servers. The principal determined at the server is based on precedence rules. If identity assertion is performed, the identity is always derived from this layer. If basic authentication is performed without identity assertion, the identity is always derived from this layer. Finally, if SSL client certificate authentication is performed without either basic authentication, or identity assertion, then the identity is derived from this layer.

---

5. The data in **Trusted Servers** box is not used by z/OS servers. The box remains displayed for visual compatiblity with distributed versions of WebSphere.

---

6. Choose whether or not to enable **Stateful** sessions. Performance is optimum when choosing stateful sessions.

   The first contact between a client and server must fully authenticate. However, all subsequent contacts, while the sessions are still valid, reuse the security information. The client passes a context ID to the server, and the ID is used to look up the session. The context ID is scoped to the connection, which guarantees uniqueness. Whenever the security session is no longer valid, if the authentication retry is enabled (it is by default), the client-side security interceptor invalidates the client-side session and resubmits the request without the user awareness. This might occur if the session does not exist on

the server (the server failed and resumed operation). When this value is disabled, every method invocation must reauthenticate.

Default: enabled.

---

7. Under **Addition Properties** click on **Additional Settings** to specify additional CSIv2 inbound authentication settings.

   This panel allow you to define the supported types of supplemental client authentication. At present only RACF userid and password (SAFUSERIDPASSWORD) is allowed. It also allows you to define the forms of asserted identity supported.

   - Client Authentication Type specifies type of client authentication supported for inbound requests. At present only RACF userid and password (SAFUSERIDPASSWORD) is allowed.
   - SAF Identity Assertion. When enabled, this server permits a trusted server to assert client identities in the form of SAF user names.
   - DN Identity Assertion. When enabled, this server permits a trusted server to assert client identities in the form of distinguished names.
   - Certificate Identity Assertion When enabled, this server permits a trusted server to assert client identities in the form of X509 certificates.

---

8. Click OK. On the next screen, click save to apply the changes to the master configuration. On the next screen check the bos for synchronizing the changes with the nodes, and click save.

You know you are done when the above sequence of saves does not produce any error messages and after the last save the initial entry panel is displayed.

Steps for configuring the CSIv2 outbound authentication protocol

Before you begin: Outbound authentication refers to the configuration which determines what type of authentication is performed for outbound requests to downstream servers. There are several layers or methods of authentication that can occur. The downstream server inbound authentication configuration must support at least one choice made in this server outbound authentication configuration. If nothing is supported, the request might go outbound as unauthenticated. This does not create a security problem, as the authorization run time is responsible for preventing access to resources that are protected. However, if you choose to prevent an unauthenticated credential to go outbound, you might want to choose one of the authentication layers to be required rather than supported.

You need to start the Administrative Console by specifying URL: http://<server_hostname>:9090/admin.

Perform the following steps to configure the CSIv2 outbound authentication protocol.

1. Click **Security** > **Authentication Protocol** > **CSIv2 Outbound Authentication** in the Navigation tree on the left.

   The CSIv2 Authentication Outbound panel appears.

Note: This panel will configure the default CSIv2 Outbound Authentication policy for all defined servers. These same values can be defined at the server level by clicking **Servers** > **Application servers** in the left-hand navigation tree and then slecting the server and clicking on **Server Security** > **CSI Authentication** > **Outbound**.

---

2. On the General Properties section of the Configuration tab, select the **Basic Authentication** setting.

   The only valid choise for this setting is never. Other choises are ignored. This field is displayed to maintain visual consistency with the distributed version of WebSphere.

   Options:

   - **Never** indicates that this server will not send user ID/password authentication to downstream servers.
   - **Supported** Not supported on zSAS.
   - **Required** Not supported on zSAS.

---

3. Select the **Client Certificate Authentication** setting.

   The main reason to enable outbound SSL client authentication from one server to a downstream server is to create a trusted environment between those servers. For delegating client credentials, use one of the two layers mentioned previously. However, you might want to create SSL personal certificates for all servers in your domain, and only trust those servers in your SSL truststore file. No other servers or clients can connect to the servers in your domain, except for at the tiers where you want them. This process can protect your enterprise beans servers from being accessed by anything other than your servlet servers. Refer to the SSL Client Certificate Authentication article in the InfoCenter for more information.

   A server can send multiple layers simultaneously, therefore, an order of precedence rule decides which identity to use. The identity assertion layer has the highest priority, the message layer follows, and the transport layer has the lowest priority. SSL client certificates are only used as the identity for invoking method requests, when that is the only layer provided. Although, SSL client certificates are useful for trust purposes, even if the identity is not used for the request. If only the message layer and transport layer are provided, the message layer is used to establish the identity for authorization. If the identity assertion layer is provided (regardless of whatever else is provided), then the identity from the identity token is always used by the authorization engine as the identity for that request.

   Options:

   - **Never** indicates that this server will not attempt SSL client certificate authentication with downstream servers.
   - **Supported** (default) indicates that this server may use SSL client certificates to authenticate to downstream servers, however, a method may be invoked without this type of authentication (for example, using anonymous instead).
   - **Required** indicates that this server must use SSL client certificates to authenticate to downstream servers.

---

4. Choose whether or not to enable **Identity Assertion**.

When enabled, this server submits an identity token to a downstream server, if the downstream server supports identity assertion. When an originating client authenticates to this server, the authentication information supplied is preserved in the outbound identity token. That is, if the client authenticating to this server uses client certificate authentication, then the identity token format is a certificate chain containing the exact client certificate chain on the socket. The same scenario is true for other mechanisms of authentication. Read the Identity Assertion article in the InfoCenter for more information.

---

5. Choose whether or not to enable **Stateful** sessions. Performance is optimum when choosing stateful sessions.

   The first method request between this server and the downstream server is authenticated. All subsequent requests (or until the credential token expires) reuse the session information, including the credential. A unique session entry is defined as a unique client authentication token and identity token combined, scoped to the connection.

   Default: enabled.

---

6. Click OK. On the next screen, click save to apply the changes to the master configuration. On the next screen check the box for synchronizing the changes with the nodes (ND only), and click save.

You know you are done when the above sequence of saves does not produce any error messages and after the last save the initial entry panel is displayed.

Steps for configuring the CSIv2 inbound transport protocol

Before you begin: Inbound transports refer to the types of listener ports and their attributes that are opened to receive requests for this server.

You need to start the Administrative Console by specifying URL: http://<server_hostname>:9090/admin.

Perform the following steps to configure the CSIv2 inbound transport protocol.

1. Click **Security** > **Authentication Protocol** > **CSIv2 Inbound Transport** in the Navigation tree on the left.

   The CSIv2 Transport Inbound panel appears.

   Note: This panel will configure the default CSIv2 Inbound Transport policy for all defined servers. These same values can be defined at the server level by clicking **Servers** > **Application servers** in the left-hand navigation tree and then slecting the server and clicking on **Server Security** > **CSI Transport** > **Inbound**.

---

2. On the General Properties section of the Configuration tab, select the desire transport setting.

   Options:

   - **TCPIP** indicates that the server only supports TCP/IP and cannot accept SSL connections.
   - **SSL-Required** indicates that an SSL listener port is opened, and all CSIv2 requests come through SSL connections.

- **SSL-Supported** (default) indicates that this server can support either TCP/IP or SSL connections..

---

3. Select the SSL settings that correspond to an SSL transport.

   These SSL settings are defined in the **Security** > **SSL** panel where you define SSL configuration repertoires. A System SSL repertoire must be used here.

---

4. Click OK On the next screen, click save to apply the changes to the master configuration. On the next screen, check the box for synchronizing the changes with the nodes, this for ND only, and click save.

You know you are done when the above sequence of saves does not produce any error messages and after the last save the initial entry panel is displayed.

Steps for configuring the CSIv2 outbound transport protocol

Before you begin: Outbound transport refers to the transport used to connect to a downstream server. When you configure the outbound transport, you should consider the transports the downstream servers support. If Secure Sockets Layer (SSL), consider including the signers of the downstream servers in this server RACF keyring for the handshake to succeed. When you select an SSL configuration, that configuration points to a RACF keyring that should contain the necessary signers. If you have configured client certificate authentication for this server in the **Security** > **Authentication Protocols** > **CSIv2 Outbound Authentication** panel, then the downstream servers should contain the signer certificate belonging to the server personal certificate.

You need to start the Administrative Console by specifying URL: http://<server_hostname>:9090/admin.

Perform the following steps to configure the CSIv2 outbound transport protocol.

1. Click **Security** > **Authentication Protocol** > **CSIv2 Outbound Transport** in the Navigation tree on the left.

   The CSIv2 Transport Outbound panel appears.

   Note: This panel will configure the default CSIv2 Outbound Transport policy for all defined servers. These same values can be defined at the server level by clicking **Servers** > **Application servers** in the left-hand navigation tree and then slecting the server and clicking on **Server Security** > **CSI Transport** > **Outbound**.

---

2. On the General Properties section of the Configuration tab, select the desired transport type.

   Options:

   - **TCPIP** indicates that the server only supports TCP/IP and cannot initiate SSL connections.
   - **SSL-Required** indicates that this server must use SSL to initiate connections to downstream servers.
   - **SSL-Supported** (default) indicates that this server can initiate either TCP/IP or SSL connections.

3. Select the **SSL Settings** that correspond to an SSL transport.

   These SSL settings are defined in the **Security** > **SSL** panel. The selected SSL configuration must be a system SSL repertoire. Ensure that the RACF keystore in the selected SSL configuration contains the signers for any downstream servers. Also, ensure that the downstream servers contain the server signer certificates when outbound client certificate authentication is used.

4. Click OK. On the next screen, click save to apply the changes to the master configuration. On the next screen, check the box for synchronizing the changes with the nodes (this is for ND only) and click save.

You know you are done when the above sequence of saves does not produce any error messages and after the last save the initial entry panel is displayed.

Steps for configuring the zSAS transport protocol

Before you begin: zSAS protocols are implemented differently on z/OS than are the corresponding SAS protocols on the WebSphere Appliucation Server ND product. For one, there are not separate inbound and outbound configurations. Use the zSAS protocol for requests that are received and sent by a server that uses the z/OS authentication protocol.

You need to start the Administrative Console by specifying URL: http://<server_hostname>:9090/admin.

Perform the following steps to configure the zSAS transport protocol.

1. Click **Security** > **Authentication Protocol** > **zSAS Transport** in the Navigation tree on the left.

   The zSAS Transport panel appears.

   Note: This panel will configure the default zSAS Transport policy for all defined servers. These same values can be defined at the server level by clicking **Servers** > **Application servers** in the left-hand navigation tree and then slecting the server and clicking on **Server Security** > **zSAS Transport**.

2. On the General Properties section of the Configuration tab, choose whether or not to enable **Basic Authentication**.

   Enabling this option specifies that clients to this server can provide a System Authorization Facility (SAF) user ID and password over a Secure Sockets Layer (SSL) connection. This option requires a valid System SSL Repertoire selection on the SSL Settings option.

   Default: Not enabled.

3. Choose whether or not to enable **Client Certificate**.

   Enabling this option specifies that clients to this server can authenticate using SSL client certificates. The client certificates must be capable of mapping to a SAF user ID. You must connect the public certificate of the

client Certificate Authority to the server key ring. The client certificate option requires a valid System SSL Repertoire selection on the SSL Settings option.

Default: Not enabled.

---

4. Choose whether or not to enable **Kerberos**.

   Enabling this option specifies that this security mechanism uses SSL to establish the trust of the client in the server. The client authenticates to the server by using Kerberos. The Kerberos identity must be capable of converting to a SAF identity. This option requires a valid System SSL Repertoire selection on the SSL Settings option.

   Default: Not enabled.

---

5. Choose whether or not to enable **Userid Password**.

   Enabling this option specifies that clients can connect to this server with a SAF user ID and password without requiring a connection sent over an SSL session.

   Default: Not enabled.

---

6. Choose whether or not to enable **Userid Passticket**.

   Enabling this option specifies that clients or other servers on the same sysplex can connect to this server with a one-time user credential that represents the SAF user.

   Default: Not enabled.

---

7. Choose whether or not to enable **Identity Assertion Inbound**.

   Enabling this option specifies that inbound requests using SAF user IDs forwarded by a z/OS Application Server can be accepted. The immediate downstream server establishes its identity by sending a digital certificate. Identity assertion is available only if client certificates are supported. When you enable this setting, you must select an SSL setting.

   Default: Not enabled.

---

8. Choose whether or not to enable **Identity Assertion Outbound**.

   Enabling this option specifies that outbound requests originating from this server can forward authenticated client user IDs over an SSL connection to another z/OS Application Server in which it has established trust. This option requires a valid System SSL Repertoire selection on the SSL Settings option.

   Default: Not enabled.

---

9. Choose whether or not to enable **Allow Unauthenticated Clients**.

   Enabling this option specifies that the server accepts Internet Inter-ORB Protocol (IIOP) requests without any authentication information. If you enable this property, specify the Remote Identity setting to associate a user ID with requests from a remote server.

   Default: Not enabled.

---

10. In the **Remote Identity** box, specifiy the SAF user ID assumed for the IIOP unauthenticated clients that make requests of this server from another system. Specify this setting even if security is not enabled.

---

11. In the **Local Identity** box, specifiy the SAF user ID assumed for the Internet Inter-ORB Protocol (IIOP) unauthenticated clients that makes requests of this server from the same system. Specify this setting even if security is not enabled.

---

12. Choose whether or not to enable **Sync to OS Thread Allowed**.

    Enabling this option specifies that the synchToOSThread method is supported for applications that specify it.

    When you enable this setting, you allow the method to process a request that modifies the operating system identity to reflect the Java™ 2 Platform, Enterprise Edition (J2EE) identity. This function is required if you wish to take advantage of thread identity support. J2EE Connector Architecture (JCA) connectors that access local resources on a z/OS system can use the thread identity support. A set of JCA connectors that accesses local z/OS resources defaults to the Java identity of the application if all the following are true:

    - Resource authorization is set to container-managed (res-auth=container)
    - No alias entry has been coded when deploying the application
    - Sync to OS Thread Allowed is set to enabled.

    Any JCA connector that uses the thread identity support must itself support thread identity. Customer Information Control System (CICS), Information Management System (IMS) and DATABASE 2 (DB2) support thread identity. CICS and IMS allow thread identity support only if the target CICS or IMS is configured on the same system as the z/OS WebSphere Application Server. DB2 always supports thread identity. If a connector does not allow thread identity, the user identity associated with the connection is based on the default user identity supported by the particular connector.

    Default: Not enabled.

---

13. Select an **SSL** setting from a predefined list of SSL settings for connections.

    The selection must be a System SSL repertoire. Configure these settings on the SSL Repertoire panel.

---

14. Click save to apply the changes to the master configuration. On the next screen, check the box for synchronizing the changes with the nodes (this isfor ND only), and click save.

You know you are done when if the above sequence of saves does not produce any error messages and after the last save the initial entry panel is displayed.

Implementing advanced performance controls

This section discusses performance issues for:

- Resource serialization
- WLM classification rules and work qualifiers

Recommendation for resource serialization

For performance reasons, we recommend you use a global resource serialization star complex. For more information, see *z/OS MVS Planning: Global Resource Serialization*, SA22-7600.

Workload management and WebSphere for z/OS

This topic discusses how WebSphere for z/OS uses the z/OS workload management subsystem and tells you how to set up workload management controls.

Background on workload management and WebSphere for z/OS

WebSphere for z/OS exploits workload management for the following general functions:

- Sysplex routing of work requests
- Address space management for work requests

Sysplex routing of work requests: WebSphere for z/OS routes work requests throughout the cell by using the domain name server (DNS). Figure 17 on page 202 shows how work gets routed in the cell. The DNS accepts a generic host name from the client and maps the name to a specific system. In order to select the best available system, the DNS asks workload management (WLM) for a recommendation. Workload management analyzes the current state of the cell and considers a number of factors, such as CPU, memory, and I/O utilization, to determine the best placement of new work. The DNS then routes the client request to the optimal system for execution. This use of workload management and the DNS is optional but highly recommended because it eliminates a single point of failure.

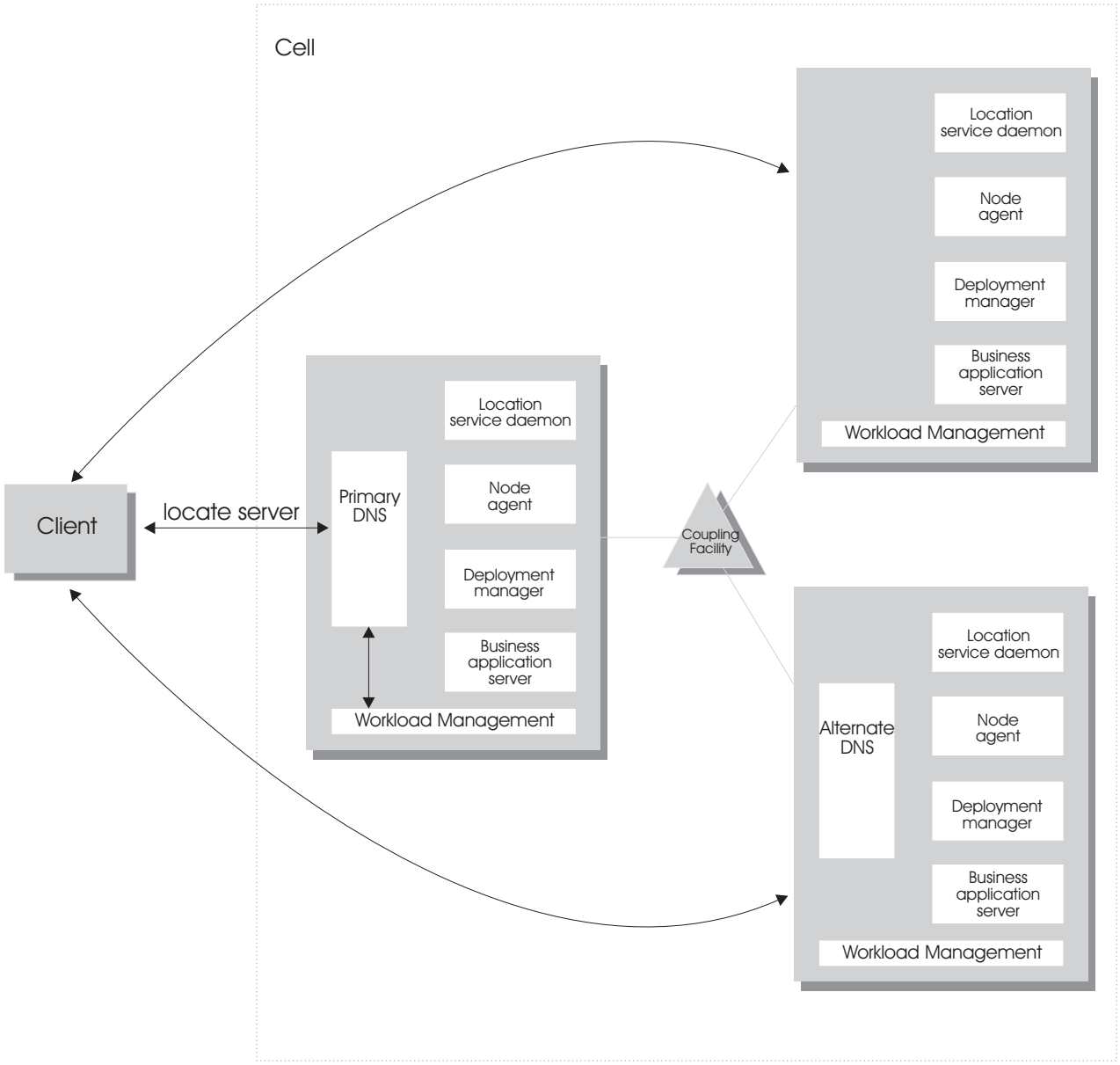


Figure 17. WebSphere for z/OS, the domain name server (DNS), and workload management

In Figure 17, each system in the cell has the WebSphere for z/OS run-time (the location service daemon, node agent and Deployment Manager), plus business application servers. The client uses the CORBA General Inter-ORB Protocol (GIOP) to make requests of WebSphere for z/OS. The location service daemon acts as a location service agent. It accepts locate requests with object keys in the requests. The location service daemon uses the object key to locate a server that supports the object represented by the object key, then hands the server name to workload management. Workload management chooses the optimal server in the cell to handle the request. The location service daemon merges specific IOR information related to the chosen server with object key information stored in the original IOR. The result of this merging is a direct IOR that gets returned to the client. The client ORB uses this returned reference to establish the IOR connection to the server holding the object of interest.

The transport mechanism that WebSphere for z/OS uses depends on whether the client is local or remote. If the client is remote (that is, not running on the same z/OS system), the transport is TCP/IP. If the client is local, the transport is through a program call. Local transport is fast because it avoids the physical trip over the network, eliminates data transforms, simplifies the marshalling of requests, and uses optimized RACF facilities for security rather than having to invoke Kerberos or SSL.

Address space management for work requests: WebSphere for z/OS propagates the performance context of work requests through the use of workload management (WLM) enclaves. Each transaction has its own enclave and is managed according to its service class. As depicted in Figure 18, the controller of a server, which workload management views as a queue manager, uses the enclave associated with a client request to manage the priority of the work. If the work has a high priority, workload management can direct the work to a high-priority servant in the server. If the work has a low priority, workload management can direct the work to a low-priority servant. The effect is to partition the work according to priority within the same server.

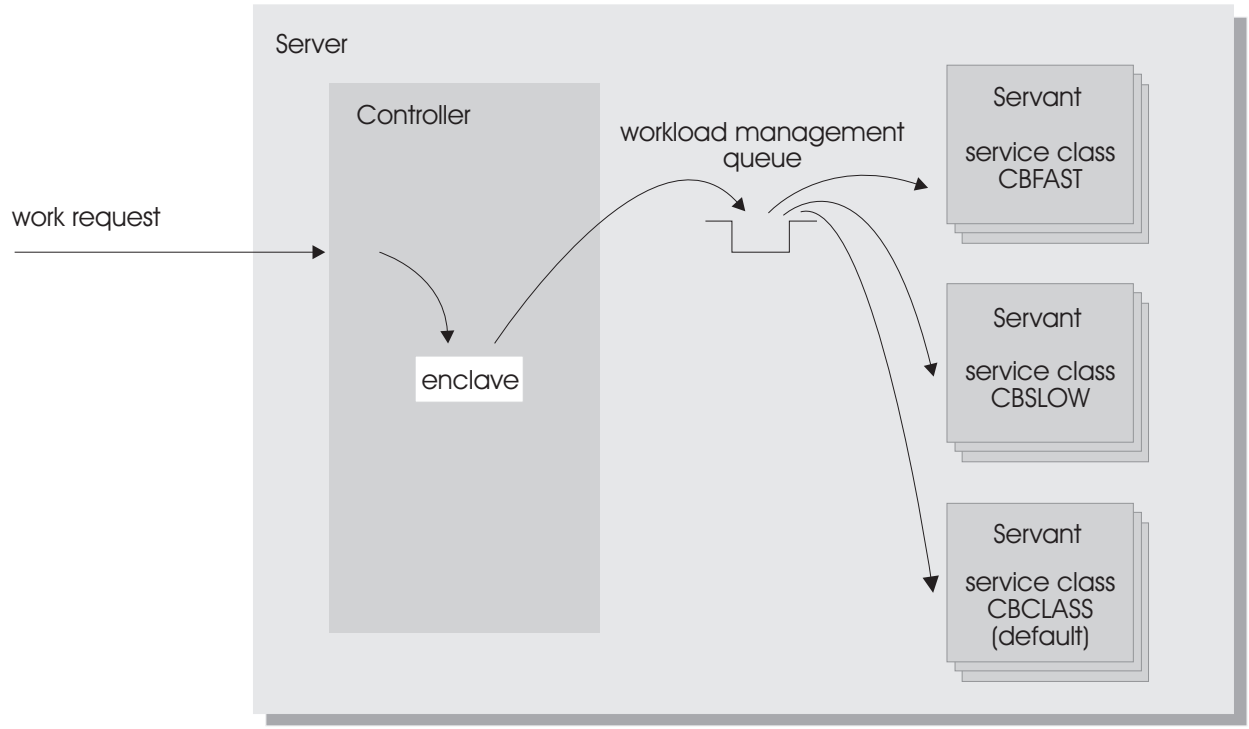


Figure 18. Use of enclaves for managing the priority of work

Enclaves can originate in several ways:

- WebSphere for z/OS uses its own set of rules to create an enclave for a client request from the network.
- Some subsystems (such as Web Server) create enclaves and pass them to WebSphere for z/OS, which, in turn, passes the enclaves on.
- WebSphere for z/OS treats batch jobs as if they were remote clients.

To communicate the performance context to workload management, you must classify the workloads in your system according to the following work qualifiers.

Table 59. WLM work qualifiers and corresponding WebSphere for z/OS entities

Work qualifier abbreviation	Work qualifier	Corresponding WebSphere for z/OS entity
CN	Collection name	Cluster name
UI	User ID	User ID under which work is running

For more information about classification rules and workload qualifiers, see *z/OS MVS Planning: Workload Management*, SA22-7602.

In addition to client workloads, you must consider the performance of the WebSphere for z/OS run-time servers and your business application servers. In general, server controllers act as work routers, so they must have high priority. Because workload management starts and stops servants dynamically, servants also need high priority in order to be initialized quickly. Once initialized, however, servants run work according to the priority of the client enclave, so the servant priority you assign has no significance after initialization.

In summary, use the following table to set the performance goals for each class:

Table 60. Workload management rules

If you are classifying...	... assign it to:	Reason
The location service daemon	SYSSTC	The system treats it as a started task, and it must route work requests quickly.
An WebSphere for z/OS run-time server controller	SYSSTC	A controller must route work quickly.
An WebSphere for z/OS run-time server servant	SYSSTC	A servant must initialize quickly, but, once initialized, it runs work according to the priority of the client enclave.
Your business application controller	A class having at least as much importance as that of the work that flows through it.	A controller must route work quickly, but you must balance the priority of your business application server with other work in the system.
Your business application servant	SYSSTC	A servant must initialize quickly, but, once initialized, it runs work according to the priority of the client enclave.
A client workload	A class having importance relative to other work in your system	WebSphere for z/OS and workload management run the work according to the goals you set.

Example of classification rules

Example: Let us assume you have three workload management service classes defined for WebSphere for z/OS (subsystem type CB):

1. CBFAST—designed for transactions requiring fast response times.
2. CBSLOW—designed for long-running applications that do not require fast response times.
3. CBCLASS–designed for remaining work requests.

You design a client workload called BBOC001 that requires fast response times. Also, you want to give work that runs under your manager's user ID (DBOOZ) slower response times. Finally, all remaining work requests should run under the default service class, CBCLASS.

Table 61. Classification rules example

Type column	Name column	Service column	Goal
CN	BBOC001	CBFAST	90% complete in 2 seconds
UI	DBOOZ	CBSLOW	Velocity 50, importance = 3
(default)	(blank)	CBCLASS	Discretionary

You could set the following performance goals through IWMARIN0:

1. Issue IWMARIN0 and choose option 4:

```
  File  Utilities  Notes  Options  Help
-------------------------------------------------------------------------------
Functionality LEVEL003         Definition Menu         WLM Appl LEVEL004
Command ===> ____________________________________________________________

Definition data set  . . : 'CB.MYCB.WLM'

Definition name  . . . . . CB390      (Required)
Description  . . . . . . . WLM Setup for WebSphere for z/OS
Select one of the
following options. . . . . 4__  1.  Policies
                               2.  Workloads
                               3.  Resource Groups
                               4.  Service Classes
                               5.  Classification Groups
                               6.  Classification Rules
                               7.  Report Classes
                               8.  Service Coefficients/Options
                               9.  Application Environments
                              10.  Scheduling Environments
```

2. Create a service class called CBFAST and specify that it be 90% complete in 2 seconds.

   Note: The example assumes you have defined a workload called ONLINE.

```
  Service-Class  Notes  Options  Help
------------------------------------------------------------------------------
                        Create a Service Class              Row 1 to 2 of 2
Command ===> ____________________________________________________________

Service Class Name . . . . . . CBFAST    (Required)
Description  . . . . . . . . . Quick CB transactions
Workload Name  . . . . . . . . ONLINE    (name or ?)
Base Resource Group  . . . . . ________  (name or ?)

Specify BASE GOAL information.  Action Codes: I=Insert new period,
E=Edit period, D=Delete period.

        ---Period---  ----------------------Goal---------------------
Action  #  Duration   Imp.  Description

  __
  __    1             1    90% complete within 00:00:02.000
****************************** Bottom of data ********************************


  .--------------------------------------------------------------------------.
  | Press EXIT to save your changes or CANCEL to discard them. (IWMAM970) |
  '--------------------------------------------------------------------------'
```

3. Save the service class. You see the following:

```
  Service-Class  View  Notes  Options  Help
------------------------------------------------------------------------------
                       Service Class Selection List          Row 1 to 14 of 21
Command ===> ____________________________________________________________

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar


Action  Class     Description                      Workload
  __    CBFAST    Quick CB Transactions            ONLINE
****************************** Bottom of data ********************************
```

4. Repeat these steps for the CBSLOW service class.
5. Create classification rules using the new service class. Choose option 6 on the main panel:

```
  File  Utilities  Notes  Options  Help
------------------------------------------------------------------------------
Functionality LEVEL003         Definition Menu         WLM Appl LEVEL004
Command ===> ____________________________________________________________

Definition data set  . . : 'CB.MYCB.WLM'

Definition name  . . . . . CB390      (Required)
Description  . . . . . . . WLM Setup for WebSphere for z/OS

Select one of the
following options. . . . . 6__  1.  Policies
                               2.  Workloads
                               3.  Resource Groups
                               4.  Service Classes
                               5.  Classification Groups
                               6.  Classification Rules
                               7.  Report Classes
                               8.  Service Coefficients/Options
                               9.  Application Environments
                              10.  Scheduling Environments
```

6. Create a set of rules for your service classes:

```
  Subsystem-Type  Xref  Notes  Options  Help
------------------------------------------------------------------------------
               Create Rules for the Subsystem Type        Row 1 to 2 of 2
Command ===> _____________________________________________ SCROLL ===> PAGE

Subsystem Type . . . . . . . . CB    (Required)
Description  . . . . . . . . . WebSphere classification
Fold qualifier names?  . . . . Y  (Y or N)

Action codes:  A=After    C=Copy         M=Move     I=Insert rule
               B=Before   D=Delete row   R=Repeat   IS=Insert Sub-rule
          -------Qualifier-------------             -------Class--------
Action    Type       Name     Start                 Service     Report
                                                DEFAULTS: CBCLAS      ________
  ____  1  CN         BBOC001  ___                      CBFAST      ________
  ____  1  UI         DBOOZ    ___                      CBSLOW      ________
****************************** BOTTOM OF DATA ******************************
```

In this example, all work for BBOC001, except for work running under the user ID DBOOZ, gets classified as CBFAST. Work for DBOOZ gets classified as CBSLOW. All other work, such as work coming from clients outside the cell and including the work for WebSphere for z/OS run-time servers, gets classified as CBCLASS.

Configuring your systems for test and production

Overview

Sharing resources between a production workload and a test workload potentially can expose the production workload to a set of error conditions to which it would not be exposed if the production and test workloads ran in different cells. For this reason, you should run production and test workloads in separate cells on your system.

Testing and production phases

Before explaining the test and production configurations for WebSphere for z/OS, you must understand which test phase should be done on the z/OS platform and which should be done on other platforms. Configuring your systems for test and production shows the test and production phases. The sections that follow explain the phases.

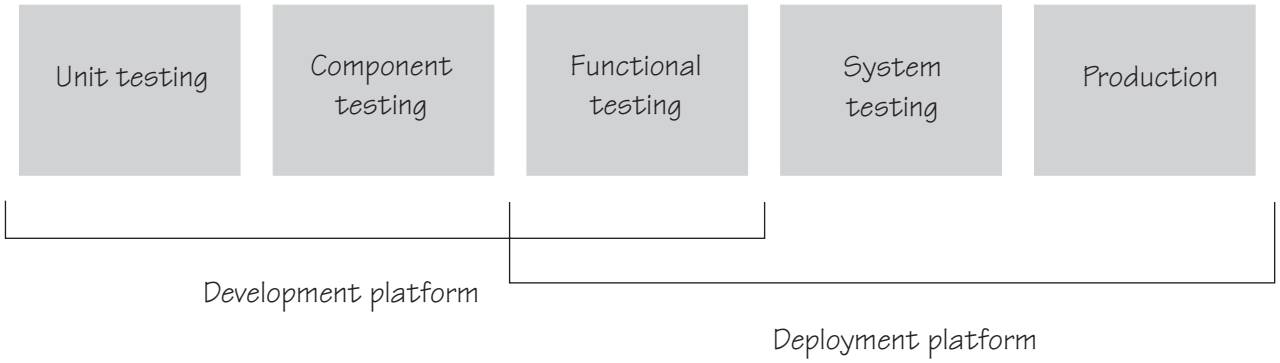


Figure 19. Test and production phases

Unit test phase

The development platform for WebSphere for z/OS is a WebSphere distributed system (for example, Windows or Linux Intel). The development environment

includes tools such as WSAD for Web content delivery. The IBM tooling solution assumes that you develop enterprise beans in WSAD and perform unit (basic) testing of the business logic in the WebSphere Test Environment.

Component test phase

Component testing involves the joining together of several beans into logical components, providing them with access to data, and testing them together. While this can be done on WebSphere for z/OS, most of our current installations perform this level of testing using a distributed platform such as Windows 2000 running WebSphere Application Server Advanced Edition. This allows a small team of developers to join the pieces of code that they have developed together and test the interactions. This testing does not test z/OS platform functions and features directly, but focuses on the individual beans and the relationships between them.

Function test phase

Function testing involves joining the various components together, connecting them to test data in the target database, and validating the function that the application provides. Where this test is performed is dependent on what the function is, and what its data requirements are. If the target deployment platform is z/OS, then it may make sense to do this level of testing there. This is possible by setting up one or more **test servers** into which the application is installed.

When the application is installed into the test server, the installer defines where in the JNDI directory the references to the application will be stored. The test clients will need to be configured with this information that tells the clients the location of the test application. The test clients will then drive requests against the test server to perform the functional testing. You can also use remote debugging tools to diagnose problems you encounter along the way.

System test phase

Before you put an application into production on z/OS, you should deploy the code into a WebSphere for z/OS server for testing. You may bring up the application and simulate a real load on the application. The important point here is that the code needs to run on z/OS before it goes into production. To do this, you need to define an additional **test server** (on a whole new cell dedicated to the test system) and install the application into it. When installed, beans that are part of the application should be registered in a different subtree of the JNDI directory (this occurs by default). The test clients need to be configured to the version of the application that is being tested and the tests run.

Production phase

You can install the application in a production WebSphere for z/OS cell after you are satisfied with the functional and system testing. The difference between a production cell and a test cell is whether the remote debugger is allowed to be attached. Normally, it is not acceptable for a production workload to stop because someone flowed a remote debugging request to it.

Test cell and production cell configuration

As Figure 20 on page 209 shows, placing test and production servers into separate cells eliminates all local sharing between test and production and provides the highest risk reduction possible. If you require complete availability of your production system, this configuration eliminates the risk of including production and test in the same cell.

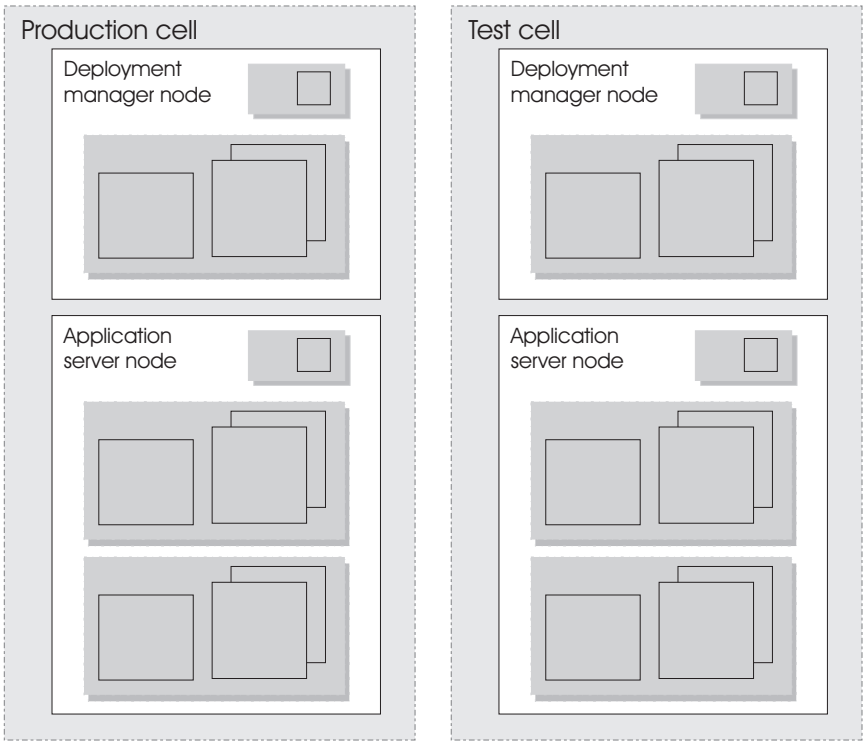


Figure 20. Test and production separated by different cells

Chapter 6. Installing new releases and maintenance levels of WebSphere for z/OS

IBM provides functions and methods to meet the need of migrating from one functional level of WebSphere for z/OS to another with as little disruption as possible. These functions and methods include the following:

- Documenting types of migration methods.
- Providing a function to off-load WebSphere for z/OS configuration data and later reload that data into a new or existing configuration.
- Managing WebSphere variables in a central location, the system management database, so that there is no confusion about where to go for authoritative configuration data.
- Supporting differing functional levels of WebSphere for z/OS within the same network or within the same z/OS sysplex while you perform an orderly migration of the WebSphere for z/OS run-time from one functional level to another. We assume this migration happens over a relatively short period of time, perhaps a few weeks.

You can install new functional levels of WebSphere for z/OS without disrupting service to your clients, provided you have the proper HFS structure in a sysplex and you use what we call a rolling upgrade. Through the rolling upgrade method you can upgrade the WebSphere for z/OS host cluster by upgrading each clustered host instance one at a time, allowing you to keep service to clients available while you do the upgrade. Availability of service continues because only one system is removed from the host cluster, allowing the other clustered host instances to keep running.

Note: This chapter doesn't apply to you if you have WebSphere for z/OS running in a monoplex or on a single system in a sysplex. In your case, installing a new code level requires shutting down WebSphere for z/OS, and you have no choice except to disrupt service to your clients. You can also ignore this chapter if you have the luxury of being able to disrupt service to your clients (whether or not you have WebSphere for z/OS running in a sysplex) and it is not a problem.

There are two methods you can employ to perform a rolling upgrade of WebSphere for z/OS. One utilizes a version-specific HFS structure while the other utilizes an alternate HFS structure. This chapter will describe each and the reasons you would use one over the other.

Note: Keep in mind whether or not you want to use a shared HFS, as your decision will impact which method you use.

Using a version-specific HFS structure to upgrade WebSphere for z/OS

Overview of creating the version-specific HFS structure for upgrades

To use the rolling upgrade method with a version-specific HFS structure, you need to have two different versions of WebSphere for z/OS in use in the sysplex at the same time, each version installed on a version-specific HFS. In the conventional

sysplex environment, you would have only one version-specific HFS and all systems in the sysplex would share it. But since you want to be able to have two different versions of WebSphere for z/OS in use in the sysplex at the same time, you must create a second version-specific HFS. See Figure 21.

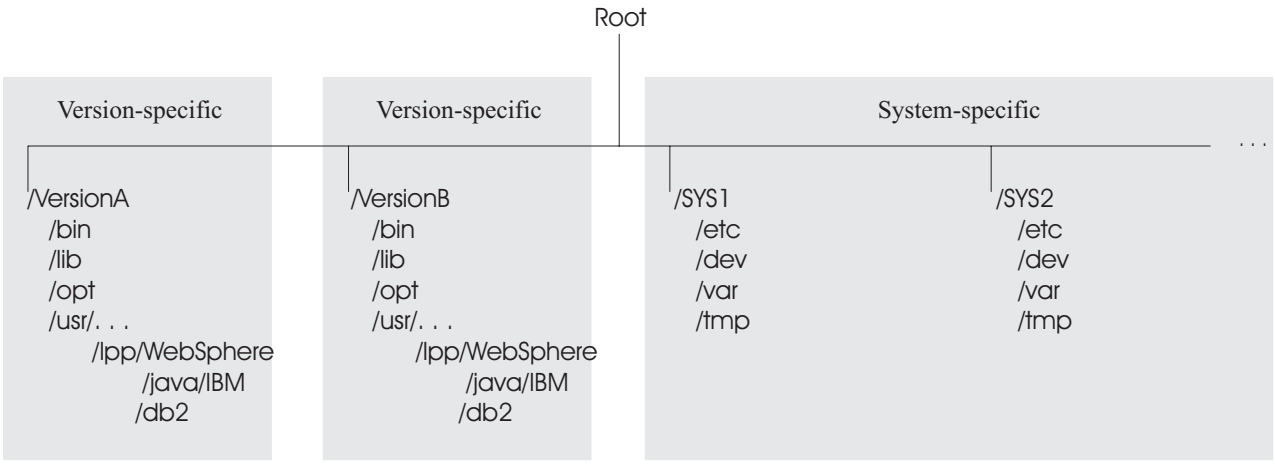


Figure 21. HFS structure for the rolling upgrade method

Because of code level interdependencies between products, both version-specific HFSes require the /usr directory (containing a version-specific level of WebSphere for z/OS and Java) and the /db2 directory (containing a version-specific level of JDBC).

With the dual HFS structure in place, you can mount a code level of WebSphere for z/OS on one mount point and run the host cluster from that mount point while upgrading the other mount point.

Example: Assume you have a version-specific HFS for one service level (PTF 10) mounted at /VersionA and another version-specific HFS for service level (PTF 15) mounted at /VersionB.

```
mount omvs.ptf10.was.hfs  at /VersionA/usr/lpp/WebSphere
mount omvs.ptf10.java.hfs at /VersionA/usr/lpp/java/IBM

mount omvs.ptf15.was.hfs  at /VersionB/usr/lpp/WebSphere
mount omvs.ptf15.java.hfs at /VersionB/usr/lpp/java/IBM
mount omvs.ptf15.jdbc.hfs at /VersionB/usr/lpp/db2
```

See Figure 22 on page 213.

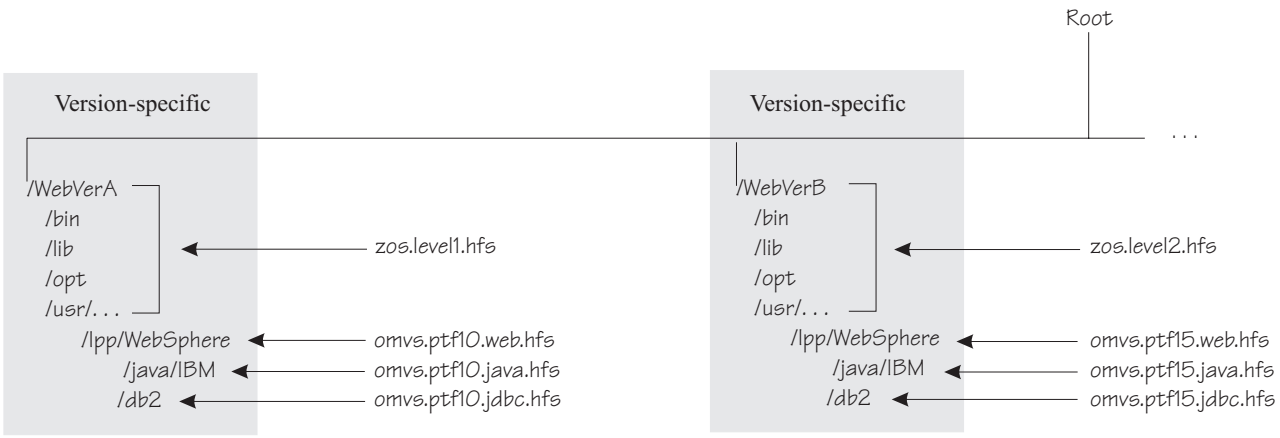


Figure 22. Mount point configuration for WebSphere for z/OS

Through the symbolic link

```
/usr --> $VERSION/usr
```

you can determine which code version any system in the sysplex addresses. You can control how $VERSION is resolved with the SETOMVS command. In this example, $VERSION for each system in the sysplex is set initially to VersionA, so all references to /usr by all systems are actually resolved to /VersionA/usr through the symbolic link.

If you wanted any system in the sysplex to use the HFSes associated with PTF15, you would change the value of $VERSION on that system (and only on that system) to VersionB. Accordingly, any references on that system to /usr are actually revolved to /VersionB/usr through the symbolic link.

To switch the code level for a given clustered host instance, you would:

- Install the new code, copy it to a new data set, and mount the data set at the VersionB mount point.
- Shut down all application servers and WebSphere for z/OS on that clustered host instance
- Use SETOMVS to change $VERSION to VersionB
- Using SET PROG, load the LPA modules from data sets associated with the new level
- Change the start procedures to address the new code level load libraries
- Restart WebSphere for z/OS and the application servers.

By repeating this process for each clustered host instance, one at a time, you can upgrade the code level of WebSphere for z/OS throughout the sysplex without disrupting service to your clients.

Each code level of WebSphere for z/OS is designed to tolerate an older code level, so differing levels of WebSphere for z/OS can coexist compatibly within the sysplex during the upgrade process. In cases when WebSphere for z/OS introduces new functions, all members of the host cluster run in compatibility mode during this upgrade process. Then, when all clustered host instances are at the new code level, you restart each instance, one by one, which enables the new function.

Using an alternate HFS structure to upgrade WebSphere for z/OS

Overview of creating the alternate HFS structure for upgrades

The alternate HFS structure accomplishes the same objective as the version-specific HFS structure, but it does not mount product HFSes directly off the version-specific subdirectories (referenced by the $VERSION symbolic link). Rather, the version-specific subdirectories refer to the system-specific subdirectories by using symbolic links with the $SYSNAME symbol. In turn, the system-specific subdirectories refer to program product subdirectories through symbolic links. The alternate HFS structure is depicted in Figure 23.

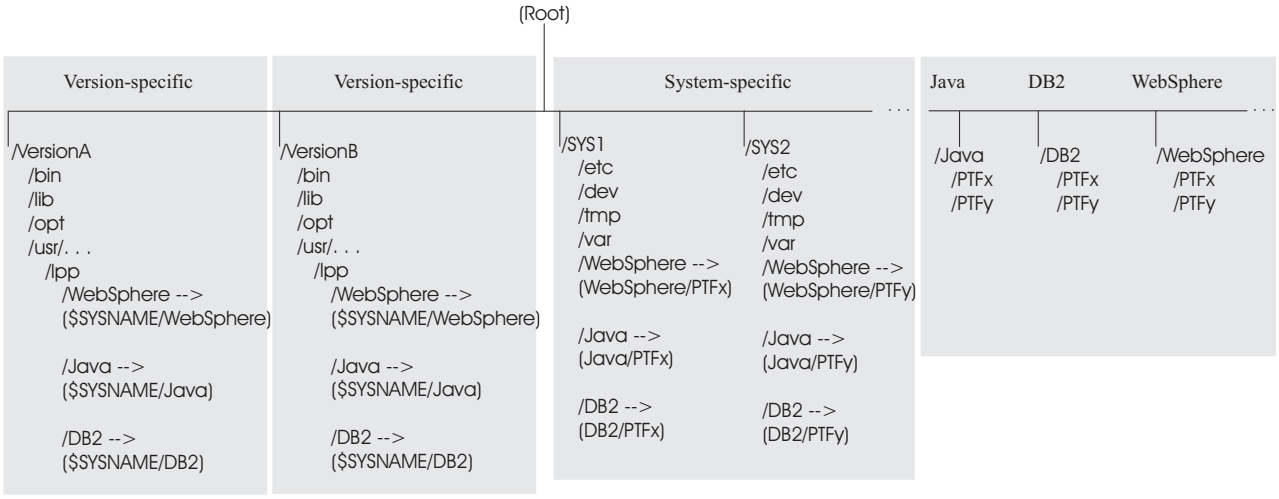


Figure 23. Alternate HFS structure

The alternate HFS structure has:

- Version-specific subdirectories that allow systems in the sysplex to refer to differing versions of system code. However, the WebSphere for z/OS, Java, and DB2 for OS/390 (JDBC) subdirectories do not contain product code. Those subdirectories contain symbolic links to system-specific subdirectories through the use of the $SYSNAME symbol. As far as WebSphere for z/OS is concerned, you do not have to change these symbolic links. You should still, however, plan for creating version-specific structures for future system upgrades.
- System-specific subdirectories that contain symbolic links to WebSphere for z/OS, Java, and DB2 for OS/390 (JDBC) subdirectories in the sysplex root. The symbolic links point to specific code levels (for example, `WebSphere/PTFx`). When you want to change the code level that a system uses, you change these symbolic links.
- Individual subdirectories for WebSphere for z/OS, Java, and DB2 for OS/390 (JDBC) components. Each of these subdirectories can have one or more subdirectories for a specific code level.
- Shared subdirectories, such as the `WebSphere390` subdirectory.

With the alternate HFS structure in place, you can mount one or more code levels of WebSphere for z/OS, Java, or DB2 for OS/390 (JDBC) under their individual component subdirectories. Each system-specific subdirectory uses symbolic links to component code levels and can refer to new code levels by changing those symbolic links.

There are certain advantages to the alternate HFS structure:

- This alternative HFS structure gives you the flexibility to stage product upgrades and service in a sysplex environment with minimal impact to availability. You can stage product upgrades or service without applying it to all products at the same time.
- By placing the level of control at the system-specific subdirectories and linking to those subdirectories through the $SYSNAME symbol, you do not need to duplicate another version-specific ($VERSION) structure when all you are doing is upgrading one product. It is, however, beneficial to plan for a second version-specific structure so you are prepared for future system upgrades.
- The version-specific subdirectories can remain read-only, benefiting performance. The changes are being done at the system-specific ($SYSNAME) subdirectory, which is read/write.
- This structure saves DASD space because you do not need to duplicate version-specific HFSes just for program product upgrades.

Example: Assume you have an individual component subdirectory for WebSphere for z/OS, Java, and DB2 for OS/390 (JDBC) and each contains two subdirectories, one for PTFx and one for PTFy. Also, the code for each component update is in its own HFS data set (OMVS.PTFX.WEB.HFS, OMVS.PTFX.JAVA.HFS, and so forth). The mount commands would be:

```
MOUNT FILESYSTEM('OMVS.PTFX.WEB.HFS')   MOUNTPOINT('/WebSphere/PTFx') TYPE(HFS) MODE(RDWR)
MOUNT FILESYSTEM('OMVS.PTFX.JAVA.HFS')  MOUNTPOINT('/Java/PTFx')      TYPE(HFS) MODE(RDWR)
MOUNT FILESYSTEM('OMVS.PTFX.JDBC.HFS')  MOUNTPOINT('/DB2/PTFx')       TYPE(HFS) MODE(RDWR)

MOUNT FILESYSTEM('OMVS.PTFY.WEB.HFS')   MOUNTPOINT('/WebSphere/PTFy') TYPE(HFS) MODE(RDWR)
MOUNT FILESYSTEM('OMVS.PTFY.JAVA.HFS')  MOUNTPOINT('/Java/PTFy')      TYPE(HFS) MODE(RDWR)
MOUNT FILESYSTEM('OMVS.PTFY.JDBC.HFS')  MOUNTPOINT('/DB2/PTFy')       TYPE(HFS) MODE(RDWR)
```

System SYS1 refers to the PTFx levels of code through these symbolic links:

```
/WebSphere --> /WebSphere/PTFx
/Java      --> /Java/PTFx
/DB2       --> /DB2/PTFx
```

If you want system SYS1 in the sysplex to use the HFSes associated with PTFy, change the symbolic links for /WebSphere, /Java, and /DB2:

```
/WebSphere --> /WebSphere/PTFy
/Java      --> /Java/PTFy
/DB2       --> /DB2/PTFy
```

Thus, to switch the code level for the WebSphere for z/OS clustered host instance on SYS1, you would:

- Install the new code for WebSphere for z/OS, Java, and DB2 for OS/390 (JDBC), copy each component to its own data set, and mount the data under its component subdirectory.

  Note: WebSphere for z/OS, Java, and DB2 for OS/390 (JDBC) code levels are usually interdependent, so keep the level of each component coordinated with the others.
- Shut down all application servers and the WebSphere for z/OS clustered host instance on SYS1.
- Change the symbolic links for the system-specific subdirectories for SYS1.
- Load new run-time modules into LPA and update the link list. You can do this dynamically, but IBM recommends you re-IPL the system.
- Change the start procedures to address the new code level load libraries.

- Restart WebSphere for z/OS and the application servers.

By repeating this process for each clustered host instance, one at a time, you can upgrade the code level of WebSphere for z/OS throughout the sysplex without disrupting service to your clients.

Appendix A. Default server values for WebSphere Application Server for z/OS V5

The following table lists the default server values for WebSphere Application Server for z/OS V5.

Table 62. Default server values for WebSphere Application Server for z/OS V5

Server	Server name (long)	Server short name	Cluster transition name / Application environment name	Run-time controller start procedure name	Run-time servant start procedure name	Subsystem type	Start parameter	Limit on starting server address space for a subsystem instance
Application Server	server1	BBOS001	BBOC001	BBO5ACR	BBO5ASR	CB	JOBNAME= &IWMSSNM.S, ENV=*cellname .nodename.* &IWMSSNM	No limit
Deployment Manager (ND only)		BBODMGR	BBODMGR	BBO5DCR	BBO5DSR	CB	JOBNAME= &IWMSSNM.S, ENV=*cellname .nodename.* &IWMSSNM	No limit
Location service daemon	BBODMNB (base) or BBODMNC (ND)			—	—	—	—	—
JMS server		BBOJ001		—	—	—	—	—
Node agent (ND only)		BBON001		—	—	—	—	—

Appendix B. z/OS port assignments

The following table lists the z/OS port assignments.

Table 63. z/OS port assignments

Port	Base location service daemon	Base Application Server	ND location service daemon	ND Application Server	Node Agent	JMS Server	Deployment Manager
HTTP		9080		9080			9090
HTTP/S		9443		9443			9043
Bootstrap		2809		9810	2809	2810	9809
ORB	5655	2809	5755	9810	2809	2810	9809
ORB SSL	5656	0	5756	0	0	0	0
SOAP/JMX		8880		8880	8878	8876	8879
DRS		7873		7873	7888		7989
JMS Queued		5558				5558	
JMS Direct		5559				5559	
Node Discovery					7272		
Node Multi-cast Discovery					5000		
Cell Discovery							7277

Appendix C. Variables and default values

The following table lists the variables you encounter when installing and configuring WebSphere for z/OS, along with each value's default and a short description. This information is also located in the BBOWVARS file in the Customization Dialog.

Table 64. Variables and their default values

Variable	Default value	Description
BBOVER	5.0	WebSphere for z/OS version
BBOPATH	NONE	which path: AS, DM, IJP
JVMHEAP	256	jvm heapsize
SYSNAME	""	system name
SYSPLEX	""	sysplex name
ERRLOG	WAS.ERROR.LOG	error logstream name
ERRVOL	123456	volume for error ds
ERRHLQ	LOGGER	logger HLQ
ERRDATCL	STANDARD	Error log stream data class
ERRSTGCL	""	Error log stream storage class
ERRLGLS	3000	log stream LS_SIZE
ERRLGSS	3000	log stream STG_SIZE
CFSTWAS	WAS_STRUCT	WebSphere Structure
CFERR	Y	CF logstreams?
RRSVOL	123456	volume for RRS data sets
RRSHLQ	LOGGER	logger HLQ
CFSTRRS	RRS.STRUCT	RRS Structure
CFRRS	Y	CF logstreams?
RRSGRP	&SYSPLEX	RRS group name
RRSDATCL	STANDARD	RRS log stream data class
RRSSTGCL	""	RRS log stream storage class
RRSLGLS	3000	RRS log stream LS_SIZE
RRSLGSS	3000	RRS log stream STG_SIZE
RRSCOPY	Y	Create RRS PROC?
CFDATCLS	STANDARD	log stream Data class
CBCONFIG	/WebSphere/V5R0M0	mount point of config HFS
JAVAHOME	/usr/lpp/java/IBM/J1.3	location of java

Table 64. Variables and their default values (continued)

Variable	Default value	Description
SMPEHOME	/usr/lpp/zWebSphere/V5R0M0	location of SMP/E
SYS1HLQ	SYS1	standard HLQ for OS390 data sets
MCAT	SYS1.MASTERCAT	master catalog name
UCAT	SYS1.USERCAT	user catalog name
PROCLIB	&SYS1HLQ..PROCLIB	proclib to update
PARMLIB	&SYS1HLQ..PARMLIB	parmlib to update
SYSEXEC	&SYS1HLQ..SYSEXEC	SYSEXEC to update
SYSEXEC	""	SYSEXEC to update
LDPLIB	GLD.SGLDLNK	name of SGLDLNK
LLDAP	Y	SGLDLNK in linklist?
SCELIB	CEE.SCEERUN	Name of SCEERUN
LSCE	Y	SCEERUN in linklist?
BBOHLQ	BBO	WebSphere for z/OS AS data set HLQ
BBOLPA	&BBOHLQ..SBBOLPA	name of SBBOLPA
LBBOLPA	Y	SBBOLPA in the linklist ?
BBOLOAD	&BBOHLQ..SBBOLOAD	name of SBBOLOAD
LBBOLOAD	Y	SBBOLOAD in the linklist?
BBOLOD2	&BBOHLQ..SBBOLD2	name of SBBOLOD2
LBBOLOD2	Y	SBBOLOD2 in the linklist?
BBOMSG	&BBOHLQ..SBBOMSG	name of SBBOMSG
BBOMIG	&BBOHLQ..SBBOMIG	name of SBBOMIG
LBBOMIG	Y	SBBOMIG in the linklist ?
BBODBRM	&BBOHLQ..SBBODBRM	name of SBBODBRM
BBOEXEC	&BBOHLQ..SBBOEXEC	name of SBBOEXEC
CNFGHFS	OMVS.WAS.CONFIG.HFS	Config HFS data set name
CNFGPRI	250	primary space
CNFGSEC	100	secondary space
CNFGVOL	"*"	volser or SMS

Table 64. Variables and their default values (continued)

Variable	Default value	Description
CBROOT	o=BOSS,c=US	RDN for CORBA name space
WASROOT	o=WASNaming,c=US	RDN for WebSphere for z/OS name space
WASPRIN	″- not used -″	principal name for WebSphere for z/OS name space access
WASCRED	secret	principal's password
RESPORT	900	resolve IP port
RESIPNAM	″″	resolve IP name
Language variables		
WASLANG	ENUS	default language
ALLGRPC	WSCTL1	main controller group name
ALLGIDC	2511	main controller GID
ALLGRPS	WSSR1	main servant group name
ALLGIDS	2501	main servant GID
ALLUSRD	WSGUEST	
ALLUIDD	2402	
ALLGRPD	WSCLGP	
ALLGIDD	2502	
CNFGGRP	WSCFG1	WebSphere for z/OS config group
CNFGGID	2500	WebSphere for z/OS config GID
WASADMN	WSADMIN	default administrator name
WASPW	WSADMIN	default administrator's password
WASUID	2403	default UID for administrator
WASGRP	WSADMGP	default group for administrators
WASGID	2503	default GID for adminstrators
location service daemon for the base Application Server variables		
DMNHOMED	&CBCONFIG./location service daemon	location service daemon Home Directory
DMNIPNAM	″″	location service daemon IP name, fully qualified

Table 64. Variables and their default values (continued)

Variable	Default value	Description
DMNIPPRT	5655	location service daemon IP port
DMNSLPRT	5656	location service daemon SSL port
DMNINSN	""	location service daemon Instance Name -- hardwired to be system name
DMNNAME	""	location service daemon Name (base) -- hardwired to be cell short name
DMNJNAME	BBODMNB	location service daemon Job Name (base)
DMNPRCC	BBO5DMN	location service daemon controller proc
DMNUSRC	WSDMNCR1	location service daemon controller userid
DMNUIDC	2411	location service daemon controller UID
DMNGIDC	&ALLGIDC	location service daemon controller GID
location service daemon for the Deployment Manager variables		
DMNHOME2	&CBCONFIG./location service daemon	location service daemon Home Directory
DMNIPNA2	""	location service daemon IP name, fully qualified
DMNIPPR2	5755	location service daemon IP port
DMNSLPR2	5756	location service daemon SSL port
DMNINSN2	""	location service daemon Instance Name -- hardwired to be system name
DMNNAME2	""	location service daemon Name -- hardwired to be cell short name
DMNJNAM2	BBODMNC	location service daemon Job Name
DMNPRCC2	BBO5DMN	location service daemon controller proc
DMNUSRC2	WSDMNCR1	location service daemon controller userid
DMNUIDC2	2411	location service daemon controller UID
DMNGIDC2	&ALLGIDC	location service daemon controller GID
Deployment Manager variables		
DMWASH	&CBCONFIG./DeploymentManager	Deployment Manager home directory
DMCENL	""	Deployment Manager cell name long
DMCENS	""	Deployment Manager cell name short
DMNONL	""	Deployment Manager node name long
DMNONS	""	Deployment Manager node name short
DMCTN	BBODMGR	Deployment Manager cluster transition name (default to server short name)

Table 64. Variables and their default values (continued)

Variable	Default value	Description
DMSSNL	dmgr	Deployment Manager server name long
DMSSNS	BBODMGR	Deployment Manager server name short
DMPRCC	BBO5DCR	Deployment Manager controller proc
DMUSRC	DMCR1	Deployment Manager controller userid
DMUIDC	2421	Deployment Manager controller UID
DMGRPC	&ALLGRPC	Deployment Manager controller group
DMGIDC	&ALLGIDC	Deployment Manager controller GID
DMPRCS	BBO5DSR	Deployment Manager servant proc
DMUSRS	DMSR1	Deployment Manager servant userid
DMUIDS	2422	Deployment Manager servant UID
DMGRPS	&ALLGRPS	Deployment Manager servant group
DMGIDS	&ALLGIDS	Deployment Manager servant GID
DMSPORT	8879	Deployment Manager scripting port/SOAP_CONNECTOR_ADDRESS port
DMNOHOST	""	Deployment Manager node host name.
DMOLHN	"*"	Deployment Manager ORB Listener host name
DMOLADDP	9809	Deployment Manager ORB_LISTENER_ADDRESS port
DMOSSLAP	0	Deployment Manager ORB SSL Listener Address port
DMCDADDP	7277	Deployment Manager CELL_DISCOVERY_ADDRESS port
DMDCADDP	7989	Deployment Manager DRS_CLIENT_ADDRESS port
DMVHHA1P	9090	Deployment Manager virtualhosts.xml: HTTP port
DMVHHA2P	9043	Deployment Manager virtualhosts.xml: HTTP SSL port
DMVHHAHN	"*"	Deployment Manager virtualhosts.xml: HTTP transport host.
Application Server variables		
ASWASH	&CBCONFIG./AppServer	Application Server home directory
ASCENL	""	Application Server cell name long
ASCENS	""	Application Server cell name short
ASNONL	""	Application Server node name long
ASNONS	""	Application Server node name short

Table 64. Variables and their default values (continued)

Variable	Default value	Description
ASCTN	BBOC001	Application Server cluster transition name (default to server short name)
ASSSNL	server1	Application Server server name long
ASSSNS	BBOS001	Application Server server name short
ASPRCC	BBO5ACR	Application Server controller proc
ASUSRC	ASCR1	Application Server controller userid
ASUIDC	2431	Application Server controller UID
ASGRPC	&ALLGRPC	Application Server controller group
ASGIDC	&ALLGIDC	Application Server controller GID
ASPRCS	BBO5ASR	Application Server servant proc
ASUSRS	ASSR1	Application Server servant userid
ASUIDS	2432	Application Server servant UID
ASGRPS	&ALLGRPS	Application Server servant group
ASGIDS	&ALLGIDS	Application Server servant GID
ASSPORT	8880	Application Server scripting port/SOAP_CONNECTOR_ADDRESS port
ASNOHOST	""	Application Server node host name. (IIOP transport)
ASOLHN	"*"	Application Server ORB Listener host name
ASOLAP	2809	Application Server ORB Listener Addressport
ASOSSLAP	0	Application Server ORB SSL Listener Address port
ASDCADDP	7873	Application Server DRS_CLIENT_ADDRESS port
ASJQADDP	5558	Application Server JMSSERVER_QUEUED_ADDRESS port
ASJDADDP	5559	Application Server JMSSERVER_DIRECT_ADDRESS port
ASVHHA1P	9080	Application Server virtualhosts.xml: HTTP port
ASVHHA2P	9443	Application Server virtualhosts.xml: HTTP SSL port
ASVHHAHN	"*"	Application Server virtualhosts.xml: HTTP transport host.
IVT variables		
IV1USRL	WSIVT	default local userid
IV1UIDL	2409	default local uid
IV1USRR	WSIVT	default remote userid

Table 64. Variables and their default values (continued)

Variable	Default value	Description
IV1UIDR	2409	default remote uid
IV1GRPU	WSIVTGP	default user group name
IV1GIDU	2509	default user group gid
IVT1USR	WSIVT	userid to run IVT1
IVT1UID	2409	uid for user to run IVT1
IVT1PW	WSIVT	password for user
IV1SCR	/tmp	location of script
CTRACE variables		
CTRPRCC	BBOWTR	CTRACE writer proc
CTRUSRC	STCRACF	CTRACE writer userid
CTRGRP	SYS1	CTRACE writer group
CTRTRCDS	SYS1.&SYSNAME..WAS390.CTRACE	trace data set name
CTRVOL	"*"	trace data set volume or SMS
CTRPRI	10	trace data set primary
CTRSEC	0	trace data set primary
RACF variables		
UEJBR	N	EJBROLE class
UOPER	N	OPERCMDS class
UKERB	N	Kerberos
USSLB	N	SSL Basic Auth
USSLC	N	SSL client Certificates
USPKT	N	Use passticket
PTKTNAM	CBS390	PTKTDATA prof name
KEYMASK	""	Keymask for passtickets
SWITCH	testit	testit or doit
TESTCA	"WAS TestCertAuth"	
IV1KR	WASKeyring	key ring for IVT1
ASKEYR	WASKeyring	key ring for AppServer

Table 64. Variables and their default values (continued)

Variable	Default value	Description
IV1KRCL	WASKeyring	key ring for IVT1 client
Home variables used for BBOMCFG		
HOMEDMN	""	home for the location service daemon
HOMEAS	""	home for the Application Server
HOMEDM	""	home for the Deployment Manager
HOMEWS	"*"	home for Web services updates
Dialog switch settings - used inside dialog		
ZBC1	""	
ZBC2	""	
ZBC3	""	
ZBC4	""	
ZBS1	""	
ZBS2	""	
ZBS3	""	
ZBS4	""	
ZBJ1	""	
ZBJ2	""	
ZBJ3	""	
ZBJ4	""	
ZBW1	""	
Integral JMS Provider settings		
MQSSID	"WMQX"	4 byte MQ Subsystem ID
MQCPF	"+"	1 byte MQ Command Prefix
CSQHLQ	"CSQ531"	
CSQAUTH	&CSQHLQ..SCSQAUTH	name of the CSQAUTH dsn
CSQANLX	&CSQHLQ..SCSQANLE	name of the CSQANLx dsn
CSQLINK	&CSQHLQ..SCSQLINK	name of the CSQLINK dsn
CSQLOAD	&CSQHLQ..SCSQLOAD	name of the CSQLOAD dsn

Table 64. Variables and their default values (continued)

Variable	Default value	Description
CSQMACS	&CSQHLQ..SCSQMACS	name of the CSQMACS dsn
CSQMVR1	&CSQHLQ..SCSQMVR1	name of the CSQMVR1 dsn
CSQMVR2	&CSQHLQ..SCSQMVR2	name of the CSQMVR2 dsn
CSQPROC	&CSQHLQ..SCSQPROC	name of the CSQPROC dsn
CSQSNLX	&CSQHLQ..SCSQSNLE	name of the CSQSNLx dsn
CSQZPARM	&CSQHLQ..CSQZPARM	name of the CSQZPARM loadmod dsn
CSQC37S	&CSQHLQ..SCSQC37S	name of the CSQC37S loadmod dsn
MACLIBDS	SYS1.MACLIB	name of the MACLIB dsn
CSQBV1	""	volser of BSDS01 dsn
CSQBV2	""	volser of BSDS02 dsn
CSQL11	""	volser of volume 1 of logcopy1
CSQL12	""	volser of volume 1 of logcopy2
CSQL21	""	volser of volume 2 of logcopy1
CSQL22	""	volser of volume 2 of logcopy2
LCSQAUTH	Y	SCSQAUTH in the linklist?
LCSQANLX	Y	SCSQANLC in the linklist?
LCSQLINK	Y	SCSQLINK in the linklist?
LCSQMVR1	Y	SCSQMVR1 in the linklist?
LCSQMVR2	Y	SCSQMVR2 in the linklist?
CSQP00	""	volser of PAGE00 dsn
CSQP01	""	volser of PAGE01 dsn
CSQP02	""	volser of PAGE02 dsn
CSQP03	""	volser of PAGE03 dsn
CSQP04	""	volser of PAGE04 dsn
CSQP05	""	volser of PAGE05 dsn
CSQP06	""	volser of PAGE06 dsn
CSQP07	""	volser of PAGE07 dsn
CSQP08	""	volser of PAGE08 dsn

Table 64. Variables and their default values (continued)

Variable	Default value	Description
CSQCUID	″″	default userid for commands
CSQRDAYS	″9999″	archive retention period (days)
CSQSTATT	″30″	statistic interval (secs)
CSQRAUD	″Y″	resource level auditing turned on
CSQSAF	″Y″	install SAF exit for Queue Manager
CSQSMPEH	″/usr/lpp/mqm/V5R3M1″	SMP/E home of java client feature
JES3	N	installation is running JES3 - required for some procs
IJPJSSP	5557	JMSSEVER_security_port

Appendix D. Notices

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

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

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, 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
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
USA

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 information 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 measurement 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 is for planning purposes only. The information herein is subject to change before the products described become available.

Examples in this book

The examples in this book are samples only, created by IBM Corporation. These examples are not part of any standard or IBM product and are provided to you solely for the purpose of assisting you in the development of your applications. The examples are provided "as is." IBM makes no warranties express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose, regarding the function or performance of these examples. IBM shall not be liable for any damages arising out of your use of the examples, even if they have been advised of the possibility of such damages.

These examples can be freely distributed, copied, altered, and incorporated into other software, provided that it bears the above disclaimer intact.

Programming interface information

This publication documents information that is NOT intended to be used as Programming Interfaces of WebSphere for z/OS.

Trademarks

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

APPN
CICS
DB2
DFSMS
ES/3090
ES/4381
ES/9000
ESA/390
IBM
IMS
IMS/ESA
Language Environment
Multiprise
MVS
Open Class
OS/390
RACF
RETAIN
RMF
RS/6000
S/390
S/390 Parallel Enterprise Server
SecureWay
System/390
VisualAge
VTAM
WebSphere
z/OS

The term CORBA used throughout this book refers to Common Object Request Broker Architecture standards promulgated by the Object Management Group, Inc.

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

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

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

The Duke logo is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries.

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

Glossary

For more information on terms used in this book, refer to one of the following sources:

- *IBM Glossary of Computing Terms*, located on the Internet at:

  `http://www.ibm.com/ibm/terminology/`
- Sun Microsystems Glossary of Java Technology-Related Terms, located on the Internet at:

  `http://java.sun.com/docs/glossary.html`
- The Sun Web site, located on the Internet at:

  `http://www.sun.com/`

Index

T

W

Program Number: 5655–I35

Printed in the United States of America

GA22-7910-00