WebSphere® Application Server V4.0.1 for z/OS and OS/390

Installation and Customization

WebSphere® Application Server V4.0.1 for z/OS and OS/390

IBM

Installation and Customization

> **Note**
>
> Before using this information and the product it supports, be sure to read the general information under
> "Appendix F. Notices" on page 365.

Contents

Figures

Tables

About this book

WebSphere Application Server V4.0.1 for z/OS and OS/390: 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 *WebSphere Application Server V4.0.1 for z/OS and OS/390: Migration*, GA22-7860. You should begin your migration planning with that manual.

- Set up WebSphere for z/OS in advanced system configurations, such as a sysplex.

Included are instructions for setting up requisite z/OS or OS/390 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 V4.0.1 for z/OS and OS/390," 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 or OS/390 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:

1. The base z/OS or OS/390 system. You must prepare various z/OS or OS/390 subsystems and your network prior to setting up WebSphere for

z/OS. For instance, you must perform such tasks as setting up security controls, defining workload management (WLM) workloads, and setting up DB2. See "Chapter 2. Preparing the base z/OS or OS/390 environment" on page 9 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 (also known as bootstrap jobs). See "Chapter 3. Installing and customizing your first run time" on page 55 for details.

"Chapter 4. Performing post-installation tasks" on page 207 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 219.

"Chapter 6. Installing new releases and maintenance levels of WebSphere for z/OS" on page 285 provides general information and procedures for migrating WebSphere for z/OS from one release or service level to another.

Note: You should begin your migration planning with *WebSphere for z/OS: Migration*.

Following the last chapter are these appendixes:
- "Appendix A. Environment files" on page 303 describes the WebSphere for z/OS environment variables.
- "Appendix B. Migrating from WebSphere for z/OS V4.0 to V4.0.1 without the customization dialog" on page 337 has procedures for migrating to WebSphere for z/OS V4.0.1 without using the customization dialog.
- "Appendix C. Using an alternate HFS structure for product upgrades" on page 345 introduces an alternate HFS structure and appropriate procedures for performing WebSphere for z/OS code upgrades.
- "Appendix D. Configuring the name space" on page 353 describes how to configure the WebSphere for z/OS naming space.
- "Appendix E. Setting up DCE" on page 359 describes how to set up DCE security.
- "Appendix F. Notices" on page 365 contains various legal notices.

"Glossary" on page 369 tells you where to find information on terms used in this manual.

Where to find related information, tools, and supplements

This is a list of books that are in the WebSphere for z/OS library. They can be found at the following Web site:

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

- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Program Directory*, GI10-0680, describes the elements of and the installation instructions for WebSphere for z/OS.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: License Information*, LA22-7855, describes the license information for WebSphere for z/OS.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834, describes the planning, installation, and customization tasks and guidelines for WebSphere for z/OS.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Messages and Diagnosis*, GA22-7837, provides diagnosis information and describes messages and codes associated with WebSphere for z/OS.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Operations and Administration*, SA22-7835, describes system operations and administration tasks.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836, describes how to develop, assemble, and install J2EE applications in a WebSphere for z/OS J2EE server.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling CORBA Applications*, SA22-7848, describes how to develop, assemble, and deploy CORBA applications in a WebSphere for z/OS (MOFW) server.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838, describes the system administration and operations tasks as provided in the Systems Management User Interface.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management Scripting API*, SA22-7839, describes the functionality of the WebSphere for z/OS Systems Management Scripting API product.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Migration*, GA22-7860, describes migration procedures for WebSphere for z/OS.

Here are some other WebSphere Application Server books on that Web site that you might find particularly helpful:

- *WebSphere Application Server for OS/390 V3.5 Standard Edition Planning, Installing, and Using*, GC34-4835, provides information about running the WebSphere for z/OS plug-in within the HTTP Server address space. You can use this configuration if you want to continue running

non-J2EE-compliant Web applications in the V4.0.1 WebSphere for z/OS plug-in within the HTTP Server address space while migrating to the full WebSphere for z/OS run time.

- *Getting Started with WebSphere Application Server*, SC09-4581, provides an overview of the WebSphere Application Server family of products.
- *Building Business Solutions with WebSphere*, SC09-4432

The integrated WebSphere Application Server Advanced Edition and WebSphere Application Server Enterprise Edition InfoCenter includes CORBA (MOFW) information you need to code CORBA (MOFW) components. Go to:

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

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

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

You might also 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. 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, 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.

Summary of changes

**Summary of changes
for GA22–7834–02
WebSphere for z/OS V4.0.1
as updated, October 2001**

This book contains information previously presented in GA22–7834–01, which
supports WebSphere for z/OS. The following is a summary of changes to this
information:

- Additional overview information about EJBROLE support was added.
- Information about the customization dialog has changed.
- Information about release migrations (in Chapter 4 of the previous edition)
  has moved to *WebSphere for z/OS: Migration*.
- Information about setting up WebSphere for z/OS in a sysplex now
  includes recommendations for HFS structures. This information is in
  "Chapter 5. Performing advanced tasks" on page 219, "Chapter 6. Installing
  new releases and maintenance levels of WebSphere for z/OS" on page 285,
  and "Appendix C. Using an alternate HFS structure for product upgrades"
  on page 345.
- Minor technical and editorial changes have also been made.

Change bars in the left margin indicate a technical change to information.

**Summary of changes
for GA22–7834–01
WebSphere for z/OS
as updated, June 2001,
service level W400017**

This book contains information previously presented in GA22–7834–00, which
supports WebSphere for z/OS. The following is a summary of changes to this
information:

- Information about the customization dialog and the changed installation
  verification programs has been added in this edition, introduced through
  APAR PQ48858 (PTF UQ900028, service level W400017) and APAR PQ49216
  (PTF UQ90029, service level W400017).
- The information about classes specified on the CLASSPATH statement has
  changed, introduced through APAR PQ48859 (PTF UQ54362, service level
  W400012).

- The information about environment variables has changed, introduced through APAR PQ47185 (PTF UQ53185, service level W400007).
- Minor technical and editorial changes have also been made.

Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Chapter 1. Overview of installation and customization

WebSphere Application Server V4.0.1 for z/OS and OS/390, hereafter referred to as WebSphere for z/OS, brings together the functions of WebSphere Application Server for OS/390 Version 3 Standard Edition and Enterprise Edition into a single product.

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. The tasks include 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 install and customize WebSphere for z/OS initially.
- Provides a picture and description of your run-time environment after the initial installation and customization. The initial installation and customization is performed on a monoplex or a single system in a sysplex.
- Provides a checklist of things 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 sysplex

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

1. The base z/OS or OS/390 system. You must prepare various z/OS or OS/390 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 or OS/390 environment" on page 9 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 (also known as bootstrap jobs). See "Chapter 3. Installing and customizing your first run time" on page 55 for details.

If you already have a release of WebSphere installed and customized, you can migrate the release to WebSphere for z/OS. See *WebSphere Application Server V4.0.1 for z/OS and OS/390: Migration*, GA22-7860.

After installation and customization, you can install application development environments for your application developers and client environments for your business applications. More information about this, see *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836.

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 connecting your business applications to an IMS or CICS database. These and other topics are in "Chapter 5. Performing advanced tasks" on page 219.

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

Figure 1 on page 3 depicts the WebSphere for z/OS run-time configuration after you install the product initially on a monoplex system or a single system in a sysplex.
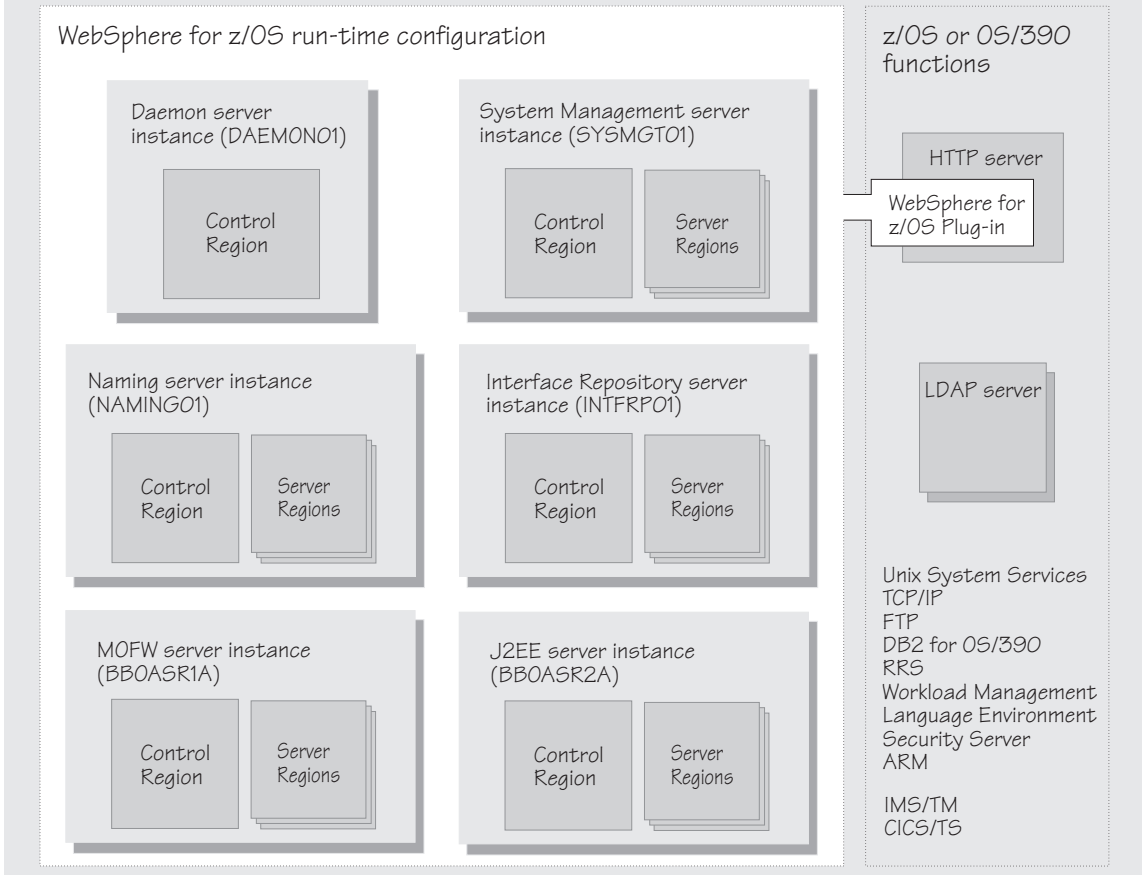
Figure 1. WebSphere for z/OS run time on a monoplex system

Before we continue, let us explain some terminology, especially the use of the word *server*. In WebSphere for z/OS, the functional component on which applications run is called a *server instance*. Server instances comprise address spaces that actually run code.

A *server*, on the other hand, is a *logical grouping* of replicated server instances. Why is that? Servers allow you to partition workloads into separate server instances, but still refer to them as a single unit. This is particularly important in sysplex environments, where each system in the sysplex might be running a replicated server instance, but clients outside the sysplex address them as a single server. The client does not know which server instance is actually doing

work on its behalf; in fact, a subsequent work request from the client may, due to workload balancing, be served by a different server instance in the sysplex.

Within each server instance are two kinds of address spaces: control regions and server regions. A *control region* runs system authorized programs and manages things such as communication for the server instance. Each server instance has one control region. A *server region* runs unauthorized programs, such as business applications. Depending on the workload, a server instance has one or more server regions running at a time (except for the Daemon, which is a specialized server instance and which has no server regions). When work builds up, additional server regions are dynamically started to meet the demand.

As Figure 1 on page 3 shows, a full WebSphere for z/OS run time includes the Daemon, System Management, Naming, and Interface Repository server instances. Though not directly part of WebSphere for z/OS, the run time requires a Lightweight Directory Access Protocol (LDAP) server. We also include two general-purpose application server instances:

- A J2EE server instance (BBOASR2A), used by the J2EE installation verification programs (IVPs) to test J2EE component support. You can use this server instance as a pattern for your servlet, Java server pages, or enterprise (EJB) bean server instances.
- A MOFW server instance (BBOASR1A), used by the MOFW portion of our installation verification program to test MOFW component support. MOFW (Managed Object Framework) is WebSphere for z/OS's implementation of CORBA-compliant components. You can use this server instance as a pattern for your MOFW components.

The run-time server instances use other z/OS or OS/390 functions, as indicated in Figure 1 on page 3, 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 or OS/390 environment" on page 9).

J2EE 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 WebSphere for z/OS run time includes a plug-in routine (pictured in Figure 1 on page 3) that runs in the HTTP server address space and routes HTTP requests to Web applications running in a Web container.

The server instances you see in Figure 1 on page 3 are automatically created during the installation on the first z/OS or OS/390 image. Table 1 on page 5 lists the default servers and their corresponding server instance and server

names.

Table 1. Server instance and server names

Server	Server instance name	Server name
Daemon	DAEMON01	CBDAEMON
System Management	SYSMGT01	CBSYSMGT
Naming	NAMING01	CBNAMING
Interface Repository	INTFRP01	CBINTFRP

During installation and customization, you will set up an LDAP server. You will also create either the MOFW server instance, BBOASR1A, and its corresponding application server, BBOASR1, or the J2EE server instance, BBOASR2A, and its corresponding server, BBOASR2, or both, depending on which IVPs you want to run.

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 or OS/390 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 server instances on one system, then replicate them on other systems as you expand into a sysplex. 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 or OS/390 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:

✔	Item	For more information, see . . .
☐	Determine the skills you need.	"Determining your skill needs" on page 9
☐	Determine WebSphere for z/OS system requirements.	"Determining WebSphere for z/OS system requirements" on page 10

✔	Item	For more information, see . . .
☐	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 19
☐	Set up workload management environments for WebSphere for z/OS run-time servers.	"Setting up workload management (WLM)" on page 35
☐	Customize resource recovery services for use by WebSphere for z/OS.	"Recommendations for resource recovery services" on page 41
☐	Plan for your performance and monitoring systems.	"Guideline for RMF and other monitoring systems" on page 42
☐	Plan for DB2 and LDAP changes.	"DB2 database and LDAP" on page 42
☐	Follow recommendations for memory utilization.	"Recommendations for using memory" on page 49
☐	Plan and define your problem diagnosis procedures.	"Planning for problem diagnosis" on page 51
☐	Consider automatic restart management before you install WebSphere for z/OS.	"Tip on automatic restart management (ARM)" on page 54

_____

2. Install and customize WebSphere for z/OS.

✔	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 55
-or-		
☐	Upgrade code levels to a new release or service level of WebSphere for z/OS.	*WebSphere Application Server V4.0.1 for z/OS and OS/390: Migration*, GA22-7860, and "Chapter 6. Installing new releases and maintenance levels of WebSphere for z/OS" on page 285

_____

3. Perform various post-installation tasks.

✔	Item	For more information, see . . .
☐	Plan and define your system backup procedures.	"Guidelines for backup of the WebSphere for z/OS system" on page 207
☐	Update the LDAP access control list, if necessary.	"Adding a new administrator for the Administration application" on page 208
☐	Plan and define your software service procedures.	"Overview of product service" on page 211
☐	Set up RACF protection for DB2, if desired.	"Setting up RACF protection for DB2" on page 212
☐	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 214

4. Plan for your application development and client environments.

✔	Item	For more information, see . . .
☐	Review WebSphere for z/OS requirements for application development and client environments.	*WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836, and *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling CORBA Applications*, SA22-7848

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

✔	Item	For more information, see . . .
☐	Plan to deploy WebSphere for z/OS in a sysplex.	"Enabling WebSphere for z/OS on a sysplex" on page 219
☐	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 238
☐	Implement advanced security controls such as SSL and Kerberos	"Implementing advanced security" on page 241
☐	Tune system performance.	"Implementing advanced performance controls" on page 263

✔	Item	For more information, see . . .
☐	Use data in your IMS database. You have two options: a. Use the OTMA interface. b. Use APPC.	• "Guidelines for the IMS-OTMA Procedural Application Adapter (CORBA applications)" on page 273 • "Setting up the IMS-APPC Procedural Application Adapter" on page 275
☐	Access CICS from CORBA applications.	"Setting up the CICS-EXCI Procedural Application Adapter for CORBA applications" on page 271

_____

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

Chapter 2. Preparing the base z/OS or OS/390 environment

Some z/OS or OS/390 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 or OS/390 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 55.

In either case, this chapter gives you background information about WebSphere for z/OS's use of z/OS or OS/390 functions and 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.

You can get started with WebSphere for z/OS by assembling a team with the following system skills:
- z/OS UNIX System Services and the hierarchical file system (HFS)
- eNetwork Communications Server (TCP/IP) or equivalent
- Lightweight Directory Access Protocol (LDAP)
- DB2
- Workload management (WLM)
- System logger and resource recovery services (RRS)
- SMP/E and JCL
- Security Server (RACF), or the security product you use

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 the automation you use
- Sysplex, if you plan to use WebSphere for z/OS in a sysplex
- Secure Sockets Layer (SSL), Kerberos, or Distributed Computing Environment (DCE), if you plan to have security in a distributed network
- RMF or other performance measurement systems
- Webserver, if you plan to support HTTP clients
- C++ or Java

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 2 Platform, Enterprise Edition (J2EE) and the Enterprise JavaBeans (EJB) component architecture
- If you plan to use CORBA components, knowledge of Common Object Request Broker Architecture (CORBA)
- Knowledge of the application development tool you use, such as VisualAge for Java and IBM WebSphere Studio.
- Windows skills
- Network File System (NFS) or File Transfer Protocol (FTP) skills

Determining WebSphere for z/OS system requirements

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

z/OS or OS/390 hardware requirements

The hardware requirements for this product are any hardware that supports OS/390 Version 2 Release 8 or z/OS and later releases of those products. 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 WebSphere for z/OS run-time and initial application servers require a minimum of 512 MB of real storage. You many need to increase the real storage size depending on the size and number of application servers you deploy.

z/OS or OS/390 software requirements for WebSphere for z/OS

The following are software requirements for WebSphere for z/OS. Consult the Program Directory or PSP bucket for the required corrective service.

- OS/390 Version 2 Release 8 (or later) 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.
- z/OS or OS/390 UNIX System Services (z/OS UNIX) with a hierarchical file system (HFS). For details, see *z/OS UNIX System Services Planning*, GA22-7800.

   Note: The WebSphere for z/OS System Management Server requires a read/write HFS. If you plan to deploy WebSphere for z/OS in a sysplex, you must establish some means of sharing the HFS in read/write mode across the sysplex. For OS/390 Version 2 Release 8, you must use the Network File System. For OS/390 Version 2 Release 9 or later and z/OS, you can choose either the Network File System or use the shared HFS function.

- 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.
- DB2 Version 7.1.

  Notes:

  1. If you run WebSphere for z/OS on more than one system in the sysplex and share workloads, you must configure DB2 in data sharing mode, which requires the Coupling Facility. For details, see *DB2 Data Sharing: Planning and Administration*, SC26-9935.
  2. If you run DB2 in a monoplex, you do not need to run in data sharing mode.
- Workload management (WLM) set up in goal mode. For details, see *z/OS MVS Planning: Workload Management*, SA22-7602.
- z/OS or OS/390 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 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 SecureWay Security Server RACF Migration*, GA22-7690.
- LDAP, a component in z/OS or OS/390 Security Server. For details, see *z/OS Security Server LDAP Server Administration and Use*, SC24-5923.
- 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 1.3.0.

  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:

If you plan to use . . .	Then you need . . .	Notes . . .
Secure Sockets Layer (SSL) security	Cryptographic Services System SSL, a component of Cryptographic Services Base, an element of z/OS or OS/390.	See *z/OS System Secure Sockets Layer Programming*, SC24-5901

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 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.
DCE security	DCE component of the Security Server, an optional element of z/OS or OS/390	See *z/OS DCE Administration Guide*, SC24-5904.
EJB roles to secure Web components or enterprise beans	z/OS V1R2 Security Server (RACF) or equivalent	For releases of z/OS or OS/390 earlier than z/OS V1R2, install the appropriate APAR for this support.
Java Message Service (JMS)	• MQSeries for OS/390 Version 5 Release 2 • JMS Support Pack MA88	MQSeries for OS/390 does not provide a message broker that supports the publish/subscribe domain. You must install a message broker on another platform and point the Queue Manager to that message broker. For example, you could install a message broker such as MQseries Integrator for the MQSeries Support pack MA0C and point the Queue Manager there. For details on MQSeries, see *MQSeries Using Java*.
Enterprise beans that access transactions and data under CICS or IMS	CICS and IMS connectors on the WebSphere Web page: `http://www.ibm.com/software/ webservers/appserv/`	The download package includes instructions, runtime code, and other materials that enable you to develop J2EE applications that use the connector support. For more information about the connector support, see the readme file.

If you plan to use . . .	Then you need . . .	Notes . . .
WebSphere for z/OS IMS-OTMA or IMS-APPC Procedural Application Adapter support	IMS/TM 6.1.0	See "Guidelines for the IMS-OTMA Procedural Application Adapter (CORBA applications)" on page 273
WebSphere for z/OS CICS-EXCI Procedural Application Adapter support	CICS/TS 1.3	See "Setting up the CICS-EXCI Procedural Application Adapter for CORBA applications" on page 271
The IBM Distributed Debugger and Object Level Trace	C/C++ with the Debug Tool feature of z/OS or OS/390	See *z/OS Planning for Installation*, GA22-7504, and *Debug Tool User's Guide and Reference*, SC09-2137

Workstation requirements

The Administration and Operations applications are shipped with WebSphere for z/OS. They require the following:

Processor
> 400 MHz (minimum)

Memory
> 256 MB (minimum)

Disk 20 MB of available hard disk space (minimum configuration)

> 50 MB of available hard disk space (with all configuration options)

Temporary disk space
> 50 MB (deleted after installation)

Display
> 800x600 capable display (minimum)

Operating system
> Microsoft Windows NT 4.0 (with service pack 3), Microsoft Windows 95 (with service pack 1 or 2), Windows 98 or Windows 2000

Communications
> TCP/IP (provided by the operating system)

Web browser
> HTML 3.2 capable (such as Netscape Navigator 4.0 or Microsoft Internet Explorer 4.0)

Java Virtual Machine
> IBM Java Runtime Environment 1.3 or higher (included with installation package)

Increasing processor speed and memory may improve your workstation performance.

Software requirements for developing WebSphere for z/OS applications
The required products for your application development environment depend on whether you are developing J2EE components or CORBA (MOFW) components. MOFW is the Managed Object Framework, IBM's implementation of the CORBA standard.

Requirements for J2EE components: If you are developing J2EE components, you need the following on your workstation:

Table 2. Software requirements for Java 2 Enterprise Edition application components

J2EE application component	Software to use
Enterprise beans	**For development:** • VisualAge for Java 3.5.3 **Recommendation:** For new application development, use VisualAge for Java 4.0. • As an alternative to using VisualAge for Java, you may use 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:
	• The IBM EJB Development Environment feature of VisualAge for Java, or
	• This combination of products:
	  – IBM or Sun Microsystems Java 2 Standard Edition (J2SE) Software Development Kit (SDK) V1.3
	  – WebSphere Application Server Advanced Edition, V3.5, or Advanced Single Server Edition, V4.0
	
	(Optional) DB2 Universal Database Version 7.1, required only for testing beans that require the use of a persistent datastore.

	For assembly: The WebSphere for z/OS Application Assembly tool Download the latest copy from the WebSphere Application Server web site (go to `http://www.ibm.com/software/webservers/appserv/` and click `Download` on the left frame).
	For installation in a J2EE server: The WebSphere for z/OS Administration application

Table 2. Software requirements for Java 2 Enterprise Edition application components (continued)

J2EE application component	Software to use
Servlets and JavaServer Pages (JSPs)	**For development and testing:** • WebSphere Studio 3.5.2 **Recommendation:** For new application development, use WebSphere Studio 4.0. • IBM or Sun Microsystems Java 2 Standard Edition (J2SE) Software Development Kit (SDK) V1.3
	For assembly: The WebSphere for z/OS Application Assembly tool
	Download the latest copy from the WebSphere Application Server web site (go to `http://www.ibm.com/software/webservers/appserv/` and click `Download` on the left frame).
	For installation in a J2EE server: The WebSphere for z/OS Administration application

Recommendation for developing EJB components: If you use the IBM EJB Development Environment feature of VisualAge for Java for developing and testing beans, servlets, and JSPs, you do not need to install the Java 2 Standard Edition (J2SE) Software Development Kit (SDK), or a WebSphere for z/OS run-time. This feature of VisualAge for Java provides a code-generated test client and a WebSphere for z/OS run-time that you can use to simulate the bean deployment environment on your workstation. The IBM EJB Development Environment 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 that servlets and JSPs can be served up to a Web browser as if they were going through an HTTP and Application Server.

Recommendation for developing Web components: Use IBM WebSphere Studio to develop servlets or JSPs, and automatically package them into Web application archive (WAR) files. (If you use other tools, you might have to create the WAR files manually.) You can then use the WebSphere for Multiplatforms to test these Web components before assembling and installing them in a J2EE server.

For J2EE components, you need the following on z/OS or OS/390:
• An FTP server that can write to the Hierarchical File System (HFS)

Requirements for CORBA (MOFW) components: If you are developing CORBA (MOFW) components, you need the following on your workstation:

- Component Broker for Windows NT 3.5
- VisualAge C++
- If developing for procedural application adaptors, VisualAge for Java Enterprise Edition 3.5

For CORBA (MOFW) components, you need the following on z/OS or OS/390:

- C/C++ IBM Open Class Library (an optional feature of z/OS or OS/390. Required for compiling code but not at run time). See *z/OS Language Environment Customization*, SA22-7564, and *z/OS Planning for Installation*, GA22-7504.

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 "Running the customization dialog" on page 61).

Tips on TCP/IP and WebSphere for z/OS

Consider the following for your TCP/IP network.

On z/OS or OS/390:

- You can get started with a simple Domain Name Service (DNS) name server and a single z/OS or OS/390 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 Daemon IP Name, while dynamically routing network traffic to replicated server instances. 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.

  You have your choice of DNS and router implementations on or off z/OS or OS/390:
  - Non round-robin DNS name servers.
  - Round robin DNS name servers.

– Connection optimization, a technique used by z/OS or OS/390 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 or OS/390. For more information, see "Connection optimization" on page 238.

– Network routers, such as the IBM Network Dispatcher. For more information, see "IBM Network Dispatcher" on page 239.

- **Select the Daemon IP name for the Daemon Server carefully.** You can choose any name you want, but, once chosen, it is difficult to change. Also, you cannot change the Daemon IP name in the middle of installation and customization.

  You must define the DAEMON_IPNAME environment variable at installation time, before you start the Daemon bootstrap process. For the value, use the Daemon IP name you chose. See "Appendix A. Environment files" on page 303.

  The bootstrap process sets, among other things, the Daemon IP name in the system management database. After bootstrap, WebSphere for z/OS uses the value in the system management database and ignores the value in the environment file. It is possible that, after bootstrap, the value of the DAEMON_IPNAME environment variable could change to a value other than what is in the system management database. If this happens, an error message is issued, but the Daemon initializes with the value from the system management database.

- Select the port for the Daemon Server and do not change it. Object references also include the port—if you change the port, existing objects will no longer be accessible. WebSphere for z/OS uses port 5555 as a default.

- In WebSphere for z/OS, the System Management Server handles the Resolve Port. Because clients are configured with a Resolve IP Name and the server returns such items as the Naming Server root, or an Interface Repository reference, the server is more resilient to change.

  Recommendation: CORBA and IBM recommend a default port 900 for the Resolve Port. If you use another port for the Resolve Port, you must change it everywhere in your distributed network.

  You may configure the bootstrap server locally on z/OS or OS/390 (it is actually the System Management Server in WebSphere for z/OS) or on another system. You can configure ports other than 900 to facilitate multiple ORBs on z/OS or OS/390.

- You can set fixed port numbers for all connections 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.

- All other ports are dynamically obtained.
- Establish a TCP/IP host address for the root naming context.
- Other TCP/IP-related activities include setting up NFS, LDAP, WebServer (optional), Kerberos (optional) and DCE (optional).

  For LDAP, we recommend you set up an LDAP server exclusively for WebSphere for z/OS even if you already have an LDAP server on your system. This exclusive LDAP server needs its own port (we suggest it be 1389).
- If you use the DNS on z/OS or OS/390, you may wish to change the refresh timer interval (-t value) associated with the named daemon. The -t value specifies the time (nn, in seconds) between refreshes of sysplex 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 DAEMON_IPNAME and RESOLVE_IPNAME 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.

On the workstation that runs the Administration and Operations applications:

The Administration and Operations applications, clients of the System Management Server that run on Windows NT, need TCP/IP setup. You must define the bootstrap server IP name and the naming server IP name in your domain name server (DNS) or your workstation HOSTS file.
- The bootstrap server IP name is the name associated with the initial connection to the host. It is defined by the RESOLVE_IPNAME parameter of the WebSphere for z/OS environment file.
- The naming server IP name is a generic name associated with your naming server and is defined by the DAEMON_IPNAME parameter of the WebSphere for z/OS environment file. If you have more than one name server (federated name space) you must ensure that all the name servers' host names needed by the workstation can be resolved.

Your workstation may have a HOSTS file, which is used to associate TCP/IP host names with TCP/IP addresses. Ordinarily, TCP/IP addresses are associated with host names by the domain name server (DNS) for your system. Your workstation uses the HOSTS file when a host name cannot be resolved using your domain name server. "Steps for updating the workstation Hosts file" on page 112 gives you instructions about how to update your HOSTS file.

Setting up security

WebSphere for z/OS supports access to resources by clients and servers 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 servers to the base operating system services in z/OS or OS/390. These services include RACF security, database management, and transaction management.
  - For the servers, you must distinguish between control regions and server regions. Control regions run authorized system code, so they are trusted. Server regions run application code and are given access to resources, so you should carefully consider the authorizations you give server regions.
  - You must also distinguish between the level of authority run-time servers and your own application servers have. For example, the System Management server needs the authority to start other servers, while your own application servers do not need this authority.
- You must authorize clients (users) to servers and objects within servers. 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.
  - What kind of objects will the client access? Enterprise beans and CORBA objects have differing authorization mechanisms.

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

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

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
- Distributed Computing Environment (DCE)

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 219. 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 55, 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

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 control region, server region, 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 server or from a server to a server, WebSphere for z/OS passes the user identity (client or server) 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 3 on page 21 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 3. Summary of controls and authorizations

Control	Authorization
Access control lists in LDAP	LDAP-controlled access to WebSphere for z/OS naming and interface repository data
CBIND class	Access to a server
DATASET class	Access to data sets
DCEUUIDS and FACILITY classes	Mapping DCE credentials to RACF user IDs
DSNR class	Access to DB2
EJBROLE 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
File permissions	Access to HFS files
GRANTs (DB2)	DB2 access to plans and database
LOGSTRM class	Access to log streams
OPERCMDS class	Start and stop servers by Daemon
PTKTDATA class	Passticket enabling in the sysplex
SERVER class	Access to control region by a server region
SOMDOBJS class	Access to methods in CORBA objects
STARTED class	Associate user ID (and optionally group ID) to start procedure
SURROGAT class (*.DFHEXCI)	Access to EXCI for CICS access

Server authorizations
Figure 2 on page 22 shows the kinds of authorization checking WebSphere for z/OS does for servers.
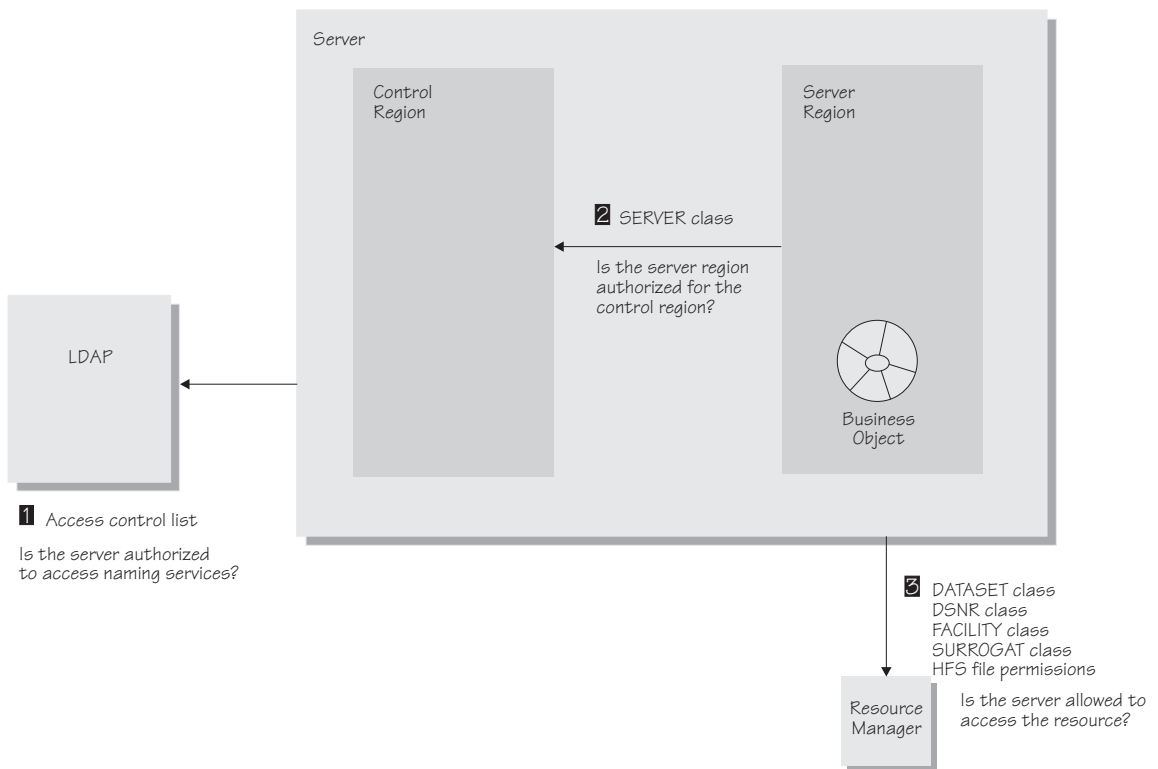
Figure 2. Server authorization checking

The following explains the numbered items in Figure 2.

1. LDAP can be set up to use access control lists (ACLs) for its objects, in which case your Naming Server and System Management Server need to be authorized to update these objects. For more information, see *z/OS Security Server LDAP Server Administration and Use*, SC24-5923.

2. Server regions must have access to profiles in the RACF SERVER class. This controls whether a server region can call authorized routines in the control region.

   Control regions do not require such access control. Only authorized programs, loaded from Authorized Program Facility (APF) libraries, run in control regions.

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

   When resource controls are used by DB2, all control regions and server regions 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 server authorization checking: To control access to WebSphere for z/OS resources:

- As a rule of thumb, give greater authority to control regions and less authority to server regions.

Table 4. Level of trust and authority for regions

Region	Level of trust and access authority
Control region	Contains WebSphere for z/OS system code. Trusted, deals with multiple users. Greater authorization. Runs APF-authorized.
Server region	Contains application code. Untrusted. Other than having authorization to get work and to attach to data stores, should run unauthorized.

- Regarding the WebSphere for z/OS run-time servers, the rule of thumb is to give less authority to the Daemon and Naming Server, and greater authority to the System Management Server, as explained in the table below:

Table 5. Assigning authorities to WebSphere for z/OS run-time server control and server regions

Run-time Server	Region	Required Authorities
Daemon Server	Control	STARTED class, access to WLM services, access to DNS, OPERCMDS access to START, STOP, CANCEL, FORCE and MODIFY other servers
Naming Server	Control	STARTED class, access to WLM services
	Server	STARTED class, READ authority to the SERVER class, DBADM for the LDAP database
System Management Server	Control	STARTED class
	Server	STARTED class, READ authority to the SERVER class, OPERCMDS access to START, STOP, CANCEL, FORCE and MODIFY other servers
Interface Repository Server	Control	STARTED class
	Server	STARTED class, READ authority to the SERVER class, DBADM for the LDAP database

- Remember to protect the RRS log streams. By default, UACC is READ.

- Protect the WebSphere for z/OS environment files, especially if they contain passwords. For more information about the environment files, see "Appendix A. Environment files" on page 303.

Client authorizations

Figure 3 shows the kinds of authorization checking WebSphere for z/OS does for clients.
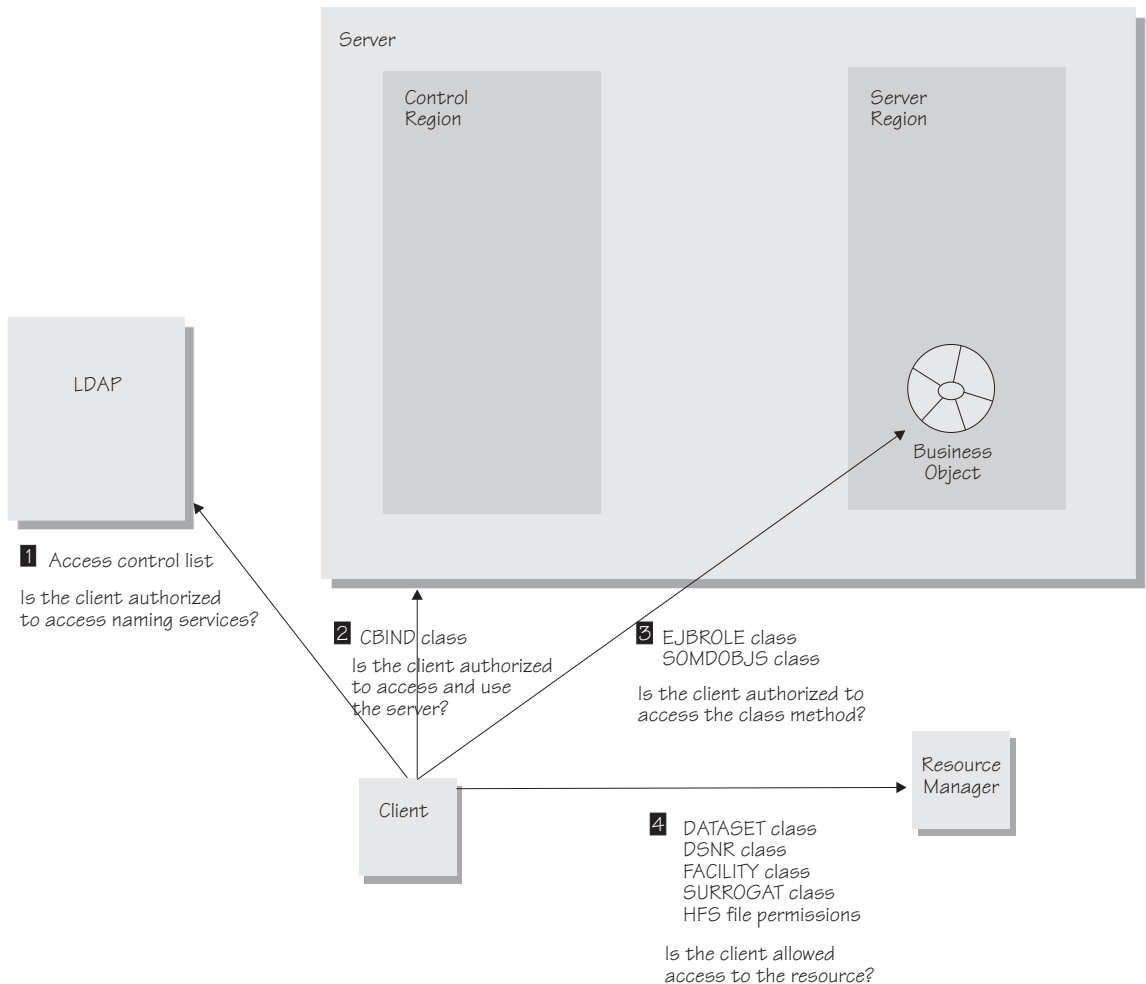
Figure 3. Client authorization checking

The following explains the numbered items in Figure 3.

1. LDAP uses access control lists to control client access to naming services. Usually, you set up a general ANYBODY user identity with read access to the LDAP name space, allowing any client to access naming services.

2. You can use the CBIND class in RACF (optional) to restrict a client's ability to access servers, 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 servers. The name of the profile has this form:

   CB.BIND.**server_name**

   where *server_name* is the name of the server.

- One that controls whether a client can use objects in a server. The name of the profile has this form:

   CB.**server_name**

   where *server_name* is the name of the server.

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

Example: CBADMIN needs read authority to the CB.BBOASR1 and CB.BIND.BBOASR1 profiles:

```
PERMIT CB.BBOASR1      CLASS(CBIND) ID(CBADMIN) ACCESS(READ)
PERMIT CB.BIND.BBOASR1 CLASS(CBIND) ID(CBADMIN) ACCESS(READ)
```

3. 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 userids 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, see *WebSphere
Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE
Applications*, SA22-7836.

- Use the SOMDOBJS class in RACF to control a client's access to CORBA
  objects. Profile names in SOMDOBJS have the form:

  server_name.home.method

  where

  server_name
    Is the server 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 server and home name lengths.

    Example: If the server 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.

4. 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 4 on page 28 shows the kinds of user identification and authentication WebSphere for z/OS uses within and across systems.
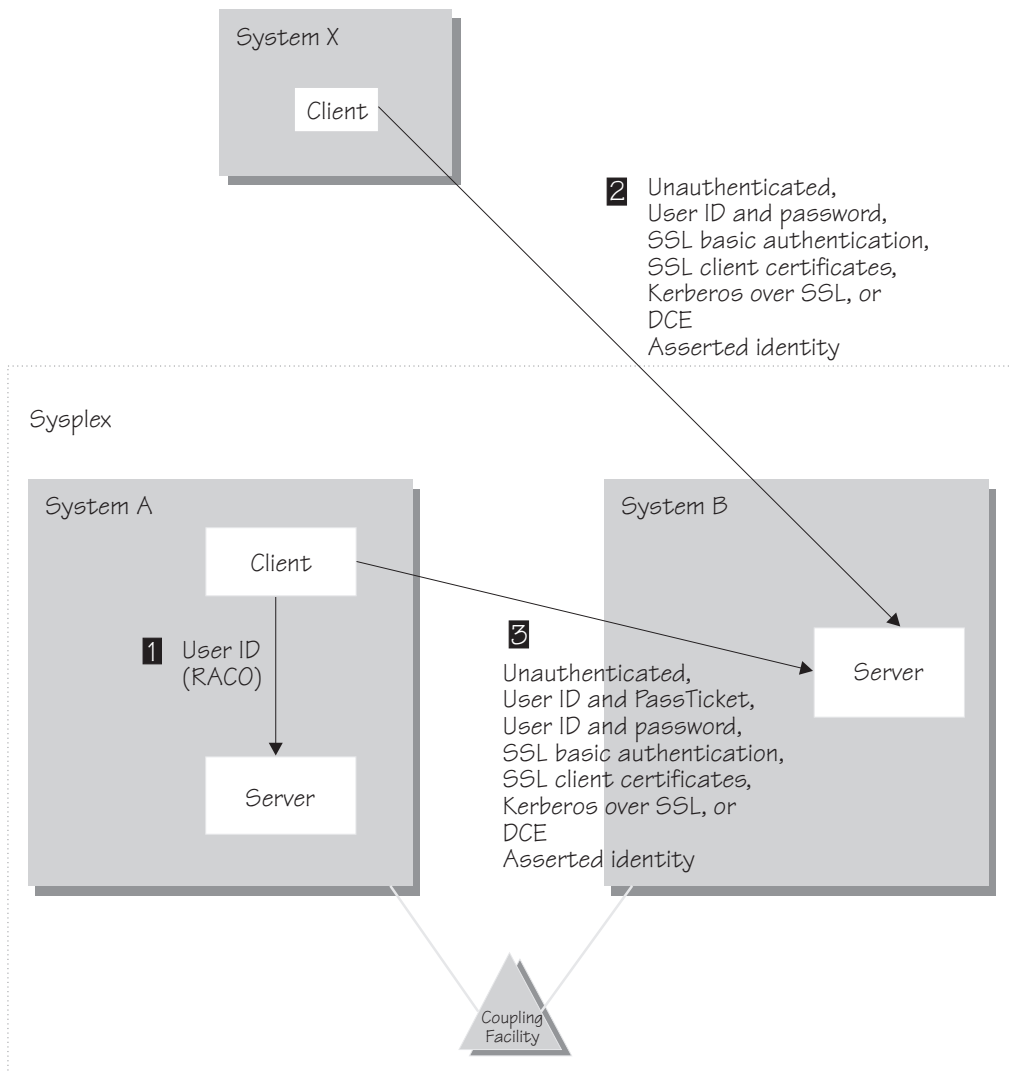
Figure 4. Identification and authentication

The following explains the numbered items in Figure 4.

1. Local clients and servers 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 servers 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 servers, message confidentiality, and message integrity become important issues. A client may want to be sure that it is receiving a service from a legitimate server and a server 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:

   - You can create a network with no security by configuring your server to accept unauthenticated clients. When you configure the server this way, every request without an identity is run under a default identity established by the server.

   - From a WebSphere for z/OS client, you can use user ID/password security, which validates the client but which offers no message protection and no guarantee that a server is authentic. User ID/Password security should never be used in an untrusted network because user IDs and passwords can easily be intercepted and reused to gain entry into the system.

   - If you want the added security of protected communications and user authentication in a network, you can use Secure Sockets Layer (SSL) security. 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 methods to prove the identities of the parties communicating. Through SSL support on WebSphere for z/OS, there are these ways to prove the identities of servers and clients:

     – Basic authentication (also known as SSL Type 1 authentication), in which a server proves its identity by passing a digital certificate to the client, much like a person presents a passport to enter another country. A client proves its identity to the server by passing a user identity and password known by the target server.

     – Client certificate support, in which both the server and client supply digital certificates to prove their identities to each other.

     – Kerberos over SSL is another authentication mechanism you can use. In WebSphere for z/OS, Kerberos client authentication is used in conjunction with SSL to provide a complete authentication mechanism, in which SSL provides message security and

authenticates the server to the client. Kerberos itself provides the ability for a server to authenticate the client.

– Asserted identity, in which an intermediate server can send the identities of its clients to a target server in a secure yet efficient manner. This function requires client certificate support to establish the intermediate server as the owner of the 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.

SSL and Kerberos support is optional: running WebSphere for z/OS without them affects the encryption and authentication functions only. You can still use other authentication mechanisms.

For details on SSL, see "Setting up SSL security for WebSphere for z/OS" on page 243. For details on Kerberos, see "Setting up Kerberos security for WebSphere for z/OS" on page 259.

- Distributed Computing Environment (DCE) security is another option you can use for clients and servers on different systems in an untrusted network. DCE uses a third-party verification technique that verifies that clients are communicating with the correct servers and servers are communicating with the correct clients. DCE also allows you to encrypt messages and check for message tampering.

DCE support is optional: running WebSphere for z/OS without installing DCE affects the DCE encryption and authentication functions only. If you do not install and activate DCE, WebSphere for z/OS cannot use DCE to authenticate remote clients.

For details on the use of DCE and its requirements, see "Appendix E. Setting up DCE" on page 359.

3. Within the sysplex, all security protocols (except for RACO) are supported between clients and servers 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 server, part of the connection includes a negotiation between the client and server 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 servers negotiate security protocols" on page 241.

Specifics about identification and authentication

For identification, each control region and server region start procedure must have its own user ID and you must define it in the STARTED class. Control regions are trusted, while server regions 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 control regions and server regions.

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 control regions and server regions.
- A user ID for the installation verification program and its application server. Our RACF sample uses CBIVP.
- A user ID called CBADMIN used by the Administration application.
- A default local and remote user ID associated with each server through the Administration application. We use CBGUEST.

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 server by using the START command and the control region 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 servers, do so through the OPERCMDS class in RACF.

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 servers, 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 19 provides an overview of WebSphere for z/OS security.

Follow these steps to choose the security you need:

1. Decide whether 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 servers to allow unauthenticated requests through the Administration application 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 6).

Table 6. Recommended security mechanisms based on your trust in the network

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

Table 6. Recommended security mechanisms based on your trust in the network (continued)

Type of network	Non-SSL Security				SSL-based Security[a]			
	local	Pass Ticket	User ID/ Pass– word	DCE	Basic Auth– tication	Kerb– eros	Client certifi– cates	Aserted identity

Notes:

a. While SSL generally causes encryption to be done, the level of encryption is negotiated by server 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 server. See "Setting up SSL security for WebSphere for z/OS" on page 243.

b. The management of asserted identities requires trust to be conferred administratively on intermediate servers.

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 the Administration application connects from the workstation to WebSphere for z/OS through user ID and password.

---

3. If your application has a server component (enterprise beans or CORBA components) that issue requests to remote servers, consider a security mechanism that provides for an authenticated identity to be transmitted to the remote servers. Some mechanisms enable the client identity to be propagated (delegated) to a remote server and some mechanisms transmit the intermediate server's identity (Table 7).

Table 7. Recommended security mechanisms based on the need to propagate a user identity

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

---

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 servers (Table 8 on page 34).

Table 8. Recommended security mechanisms based on the software configuration and client characteristics

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

Note:

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

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

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

In this example, you deploy two J2EE servers (CBSRV1 and CBSRV2) in a sysplex. Clients communicate with the system through CBSRV1 and CBSRV1 propagates client identities to CBSRV2 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 6 on page 32).

2. Your application requires that the client identity be propagated to other servers. You may use PassTicket, asserted identities, Kerberos, or DCE (see Table 7 on page 33).

   - 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.

   - Asserted identity security requires the client's MVS identity be defined on both MVS systems. You must define SSL certificates and key rings for CBSRV1 and CBSRV2 through RACF. Also, you must define a trust relationship between CBSRV1 and CBSRV2 by giving CBSRV1 RACF CONTROL authority for the CB.BIND.CBSRV2.* profile.

   - Kerberos security is the most robust of the security mechanisms in WebSphere for z/OS. Kerberos is scalable and delegates Kerberos network identities securely. However, you must install and configure Kerberos and SSL, which is a significant task.

   - DCE security is an option if you already have DCE security implemented.

   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 SSL basic authentication for network interactions because WebSphere Application Server Enterprise Edition (distributed) supports that security mechanism.

In this example, you would define PassTicket and SSL Type 1 (basic authentication) for CBSRV1 and PassTicket security for CBSRV2.

Setting up workload management (WLM)

WebSphere for z/OS uses the workload management (WLM) function in z/OS or OS/390 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 219.

Setting up workload management (WLM) in goal mode

WebSphere for z/OS requires that z/OS or OS/390 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 *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836.

Overview of workload management and servers

You need to define application environments for the System Management Server, Naming Server, and Interface Repository Server (you do not define an application environment for the Daemon Server). Without these definitions, WebSphere for z/OS will not start.

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 263.

Because the installation verification programs need servers, you must also define an application environment for the MOFW application server, the J2EE application server, or both, depending on whether you plan to use MOFW or J2EE components. We include those servers in the tables below.

Just like servers for your business applications, the WebSphere for z/OS run-time servers (with the exception of the Daemon) have a control region and one or more server regions. The regions are started by the start procedures shown in Table 9.

Table 9. Start procedures for run-time control and server regions

Server	Server name	Control region start procedure	Server region start procedure
Naming Server	CBNAMING	BBONM	BBONMS
System Management Server	CBSYSMGT	BBOSMS	BBOSMSS
Interface Repository Server	CBINTFRP	BBOIR	BBOIRS
MOFW application server	BBOASR1	BBOASR1	BBOASR1S

Table 9. Start procedures for run-time control and server regions (continued)

Server	Server name	Control region start procedure	Server region start procedure
J2EE application server	BBOASR2	BBOASR2	BBOASR2S

For business application servers, you have to start the control regions yourself. For the WebSphere for z/OS run-time servers, however, you need only start the Daemon, which in turn starts the control regions for the System Management Server, Naming Server, and Interface Repository Server. Workload manager dynamically starts the server regions as work requests arrive. Thus, you must create WLM application environments that name **server** region start procedures to start, as shown in Table 10. For example, specify BBOASR1S as the start procedure name that workload management starts for the BBOASR1 server.

Each new server that you create for a business application also needs to be defined to workload management. See *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836.

Step for defining workload management policies for the run-time servers
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:
- Use the ISPF application IWMARIN0 to define WLM application environments according to the following table:

Table 10. Application environment specifications for run-time servers

Run-time server	Application environment	Subsystem type	Procedure name for the run-time server region	Start parameter	Limit on starting server address space for a subsystem instance[1]
Naming Server	CBNAMING	CB	BBONMS	IWMSSNM=&IWMSSNM	No limit
System Management Server	CBSYSMGT	CB	BBOSMSS	IWMSSNM=&IWMSSNM	No limit
Interface Repository Server	CBINTFRP	CB	BBOIRS	IWMSSNM=&IWMSSNM	No limit

Table 10. Application environment specifications for run-time servers (continued)

Run-time server	Application environment	Subsystem type	Procedure name for the run-time server region	Start parameter	Limit on starting server address space for a subsystem instance[1]
MOFW application server	BBOASR1	CB	BBOASR1S	IWMSSNM=&IWMSSNM	Single address space per system[2]
J2EE application server	BBOASR2	CB	BBOASR2S	IWMSSNM=&IWMSSNM	No limit

Notes:

1. You can specify "No limit", or "Single address space per system." You cannot specify "Single address space per sysplex."

2. The MOFW installation verification program runs in BBOASR1 and is an example of a program that makes the state of transient objects available to other transactions, which requires that all transactions run in the same address space (server region). If all transactions do not run in the same server region, one transaction may process in one server region and a second transaction that depends on the state of a transient object may process in a different server region. However, the state of the transient object would not be available to the second transaction. To set up a server like BBOASR1, you must do the following:

   a. Set up only one server instance for the server. You cannot replicate server instances because that would result in more than one server region (address space).

   b. Set the workload management "Limit on starting server address space for a subsystem instance" to "Single address space per system." You cannot use "No limit" because that could result in more than one server region (address space).

   c. Using the Administration application, set the following server attributes for your application server:
      – Check the Production check box
      – Set the Isolation policy to multiple transactions per server region.

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 BBOASR1. You must perform the steps in the example for each server in Table 10 on page 37.

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 BBOASR1 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  . . . BBOASR1_____    Required
 Description  . . . . . . . . . CB IVP Server_____
 Subsystem Type . . . . . . . . CB__                             Required
 Procedure Name . . . . . . . . BBOASR1S
 Start Parameters . . . . . . . IWMSSNM=&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) |
     '------------------------------------------------------------------'
```

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
    __      BBOASR1                          CB IVP 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.

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 55.

When setting up RRS, consider the following:

1. You may already have configured RRS for z/OS or OS/390 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. In general, WebSphere for z/OS performs a modified distributed commit even for those protected resources that are accessed or modified in a single server region, and you may observe these global transactions in the in-doubt state. In-doubt is a very short-lived state when the transaction is local to a given application server. However, because the transaction does enter the in-doubt state, RRS logs hardened data in the DELAYED.UR log.

     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 still may 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.

DB2 database and LDAP

This section explains how WebSphere for z/OS uses DB2 and LDAP (Lightweight Directory Access Protocol), provides guidelines for these two functions, describes DB2 operational considerations, and discusses rules about LDAP security.

After installation and customization is complete, you may wish to use RACF to protect DB2 resources. For more information, see "Setting up RACF protection for DB2" on page 212.

Overview

This section describes the relationships between WebSphere for z/OS, DB2, and LDAP.

For WebSphere for z/OS, the LDAP component of the z/OS or OS/390 Security Server provides the directory services for the Java Naming and

Directory Interface (JNDI) and CORBA (MOFW) naming and interface repository services. The contents of the directory are stored in DB2 tables.

At run time, your J2EE components require an LDAP server to be running for the JNDI name services. We recommend that you use the LDAP server you create during installation and customization for this purpose. CORBA (MOFW) components do not require an LDAP server to be running because they rely on the Naming Server, which runs the LDAP DLLs in its own address space. In both cases, you need an LDAP server for administrative purposes, such as adding users to the LDAP access control list.

Recommendation: Even if you already have an LDAP server on your system, create a new LDAP server and database for WebSphere for z/OS. The reasons are:
- The data you put in the database is of interest only to WebSphere for z/OS and accessible through WebSphere for z/OS services.
- An exclusive LDAP server and database helps you keep the WebSphere for z/OS databases synchronized.

Note: If you have an existing WebSphere Application Server Enterprise Edition for OS/390 V3.02 LDAP database, schema changes require that you migrate that database using an unload/reload operation.

For in-depth instructions about setting up LDAP, refer to *z/OS Security Server LDAP Server Administration and Use*, SC24-5923.

During installation and customization, you must create an LDAP server (or use an existing LDAP server), create the LDAP database, run bind jobs, set DB2 grants, and initialize the LDAP directories. You will find these instructions in the customized instructions produced by the customization dialog (for more information about the customization dialog, see "Running the customization dialog" on page 61).

Structure of the LDAP configuration files
The main LDAP configuration file, *system*.bboslapd.conf, uses include statements to include the other configuration files. Typical LDAP configuration files also include a dsnaoini statement, which points to the DSNAOINI data set, the DB2 initialization file. However, in order to place our version of DSNAOINI into the HFS, the start procedures for LDAP, the Naming server region, and the Interface Repository server region must point to DSNAOINI through a DD statement (our samples do that for you). When you use such a DD statement in the start procedures, you do not need to use the dsnaoini statement in the LDAP configuration file. Thus, we comment out the dsnaoini statement in bboslapd.conf.

The structure looks like this:
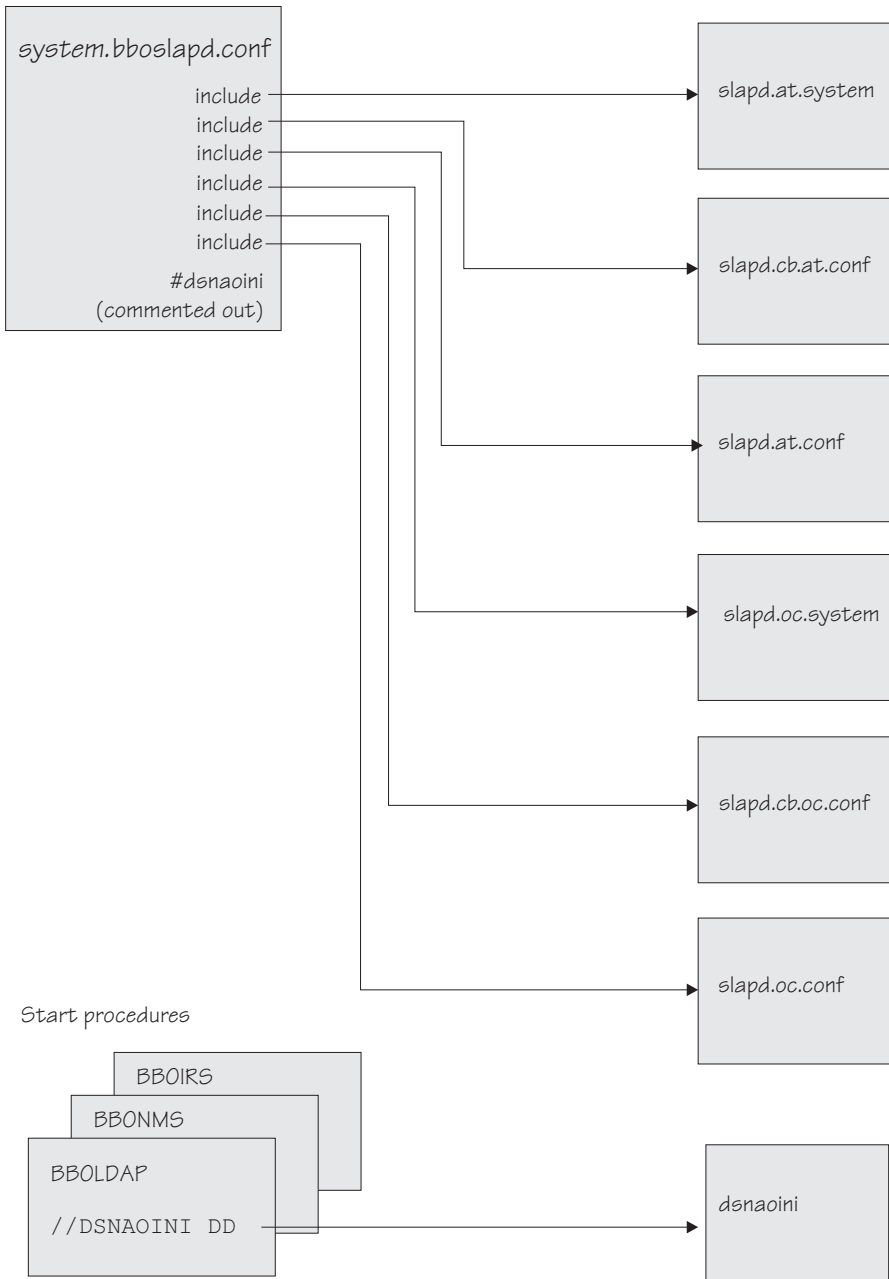
Figure 5. LDAP configuration file structure

Guidelines, rules, and recommendations for DB2 and LDAP

Follow these guidelines and recommendations to set up DB2 and LDAP:

- The VARCHAR FROM INDEX option must be set to NO. This option is set through the DB2 installation panels (see the bold print in the figure).

```
   1  DATE FORMAT          ===> ISO      ISO, JIS, USA, EUR, LOCAL
   2  TIME FORMAT          ===> ISO      ISO, JIS, USA, EUR, LOCAL
   3  LOCAL DATE LENGTH    ===> 0        10-254 or 0 for no exit
   4  LOCAL TIME LENGTH    ===> 0         8-254 or 0 for no exit
   5  STD SQL LANGUAGE     ===> NO       NO or YES
   6  CURRENT DEGREE       ===> 1        1 or ANY
   7  CACHE DYNAMIC SQL    ===> NO       NO or YES
   8  OPTIMIZATION HINTS   ===> NO       Enable optimization hints.  NO or YES
   9  VARCHAR FROM INDEX   ===> NO       Get VARCHAR data from index. NO or YES
  10  RELEASE LOCKS        ===> YES      Release cursor with hold locks. YES,NO
```

- The CORBA (MOFW) support in WebSphere for z/OS does not support DBCS. You must configure DB2 with the DSNHDECP parameter MIXED=NO.

- LDAP RDBM support in WebSphere for z/OS requires you configure DB2 with DSNHDECP parameter MIXED=NO.

- If you are configuring a J2EE server and you would like to support DBCS data, you should configure DB2 with the DSNHDECP parameter MIXED=YES and you must use the LDAP TDBM backend. The RDBM backend does not support MIXED=YES.

- Check the size of your DB2 logs. They might need to be larger because of the number of transactions WebSphere for z/OS generates.

- Increase the BP32K buffer pools to at least 100.

- Check the size of your DSNDB07 database.

- Check the 32K temporary work space for DB2. Your installation may not have had use for this work space before, but WebSphere for z/OS uses it. You must run a DB2 job called DSNTIJTM (in *hlq*.SDSNSAMP) during DB2 installation to allocate the work space. If this allocation is not large enough, you may get an SQL -904 return code when bringing up the LDAP server, the System Management Server, or the Naming Server.

- Take note of the fact that WebSphere for z/OS uses row-level locking and Type 2 indexes.

- If possible, keep the WebSphere for z/OS LDAP tables separate from other LDAP tables. The reason for keeping the sets of LDAP tables separate is that you need to back up the WebSphere for z/OS LDAP tables with the WebSphere for z/OS system management database as a unit. Performing such a coordinated backup is easier if the WebSphere for z/OS LDAP tables are separate from other LDAP tables. Additionally, if you need to restore the WebSphere for z/OS environment, restoring the WebSphere for z/OS LDAP tables will not interfere with LDAP tables used by other applications.

- Access to naming services is controlled and managed by LDAP access control lists. The sample LDIF file we provide (bboldif.cb) provides two LDAP access IDs with write access to the name space: CBAdmin and WASAdmin. Because they have write access, you may want to change the administrative password in the LDIF file.

  If you change the password for CBAdmin (you can do this through the customization dialog at installation and customization time), you must update the LDAPBINDPW environment variable for the Naming server and the LDAPIRBINDPW environment variable for the Interface Repository server. Update the environment variable in the current.env file for each server. For more information, see "Appendix A. Environment files" on page 303.

  Note: General run-time name lookup requires read access to the name space. The sample LDIF file provides an access ID with read access called ANYBODY, which allows any user to access name services.

- DB2 counts RRSAF threads in the Batch thread bucket. IBM recommends you increase the MAX USERS and MAX BATCH CONNECT settings as follows. In DSN710.SDSNSAMP(DSNTIJUZ):
  - Increase the CTHREAD parameter, used to increase MAX USERS
  - Increase the IDBACK parameter, used to increase MAX BATCH CONNECT

  Example:
  ```
  DSN6SYSP AUDITST=NO,
           BACKODUR=5,
           CHKFREQ=50000,
           CONDBAT=64,
           CTHREAD=700,
           DBPROTCL=DRDA,
           DLDFREQ=5,
           DSSTIME=5,
           EXTRAREQ=100,
           EXTRASRV=100,
           EXTSEC=NO,
           IDBACK=500,
           IDFORE=40,
           IDXBPOOL=BP0,
           LBACKOUT=AUTO,
  ```

  For more information, see *DB2 Installation Guide*, GC26-9936.

- **Attention:** During installation and customization, you are instructed to run a job called BBO1JCL. **If you have already run this job, or if the DSNACLI plan already exists on your system, do not run it again because this will destroy all GRANT privileges established for DB2.**

If you are not a DB2 expert, contact one to determine if BBO1JCL has already been run or if DSNACLI already exists. To determine this, run the following SPUFI query, which tests to see whether the DSNACLI plan has already been bound:

```
select * from sysibm.sysplan where name='DSNACLI';
```

If you get SQLCODE=100, DSNACLI has not been bound. You may safely run BBO1JCL.

If BBO1JCL has already been run or DSNACLI already exists, you have some alternatives:

– Bind the plan again specifying RETAIN so that existing privileges are not lost.
– Find out who has execute privileges on the plan, run BBO1JCL again, then re-grant the privileges. To find out who has execute privileges on the plan, run the following SPUFI query:

```
select  * from sysibm.sysplanauth where name='DSNACLI';
```

– Create a new plan name (for example, BBOACLI), update the dsnaoini file used by BBOLDAP and WebSphere for z/OS with the new plan name, then bind the new plan using the same package names and DBRMs as in BBO1JCL. Then update the execute permissions appropriately for BBOLDAP, BBOIRS, and BBONMS, or PUBLIC (depending on your installations policies).

Guidelines for Java Database Connectivity and static SQL

Java Database Connectivity (JDBC) provides an interface for Java application programs to access relational data in a database by using dynamic SQL. Static SQL (SQLJ) provides support for embedded static SQL in Java applications and applets. DB2 supports these application programming interfaces. For complete information about JDBC, SQLJ, 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 and SQLJ.

• You may use JDBC (dynamic SQL) and SQLJ (static SQL) in your server applications.
• As part of JDBC customization, you will create a SQLJ/JDBC run-time properties file, db2sqljjdbc.properties. If you customize this file, you may want to keep the customized version in a separate directory such as /etc. In any case, because you will need this information during the WebSphere for z/OS customization process, note the name and location of this file.
• All J2EE servers and the System Management server 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 J2EE servers and the System Management server. If you use DB2 secondary authorization IDs, then you can grant the authority to the groups to which the server IDs belong.

Note: During installation and customization, you use the BBOCBGRT job (produced by the customization dialog) to grant various user IDs authority to access DB2. This GRANT job issues:

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

You may want to alter or remove the statement.

- You must use the RRSAF attachment interface (not CAF).

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

Notes on planning for DB2 operations

When planning for operations, note the following:

- WebSphere for z/OS uses DB2 for its control information. Thus, DB2 must be running for the WebSphere for z/OS run-time servers to run. If you plan to stop DB2 in order to do maintenance, you must also stop WebSphere for z/OS. Also, you must stop LDAP before DB2 will shut down.
- When displaying DB2 threads with the -dis thd(*) command, the correlation ID is CB390. The Authid column contains the user id of the active/last request.

Example:

```
NAME    ST A   REQ   ID         AUTHID   PLAN     ASID TOKEN
RRSAF   T      9     CB390      DINGES   ?RRSAF   0045 436
RRSAF   T      841   CB390      CBNAMSR1 ?RRSAF   0044 435
RRSAF   T      1457  CB390      CBNAMSR1 ?RRSAF   0031 434
RRSAF   T      83    CB390      CBINTSR1 ?RRSAF   001E 433
RRSAF   T      221   CB390      CBIVP    ?RRSAF   0015 432
RRSAF   T      3709  CB390      CBNAMSR1 ?RRSAF   0038 431
RRSAF   T      1923  CB390      CBSYMCR1 ?RRSAF   0040 12
RRSAF   T      2078  CB390      CBSYMCR1 ?RRSAF   0040 13
RRSAF   DI     2300  CB390      CBSYMCR1 ?RRSAF   0040 14
RRSAF   T      1285  CB390      CBSYMCR1 ?RRSAF   0040 350
RRSAF   T      452   CB390      CBDMNCR1 ?RRSAF   003F 10
RRSAF   T      31    CB390      CBDMNCR1 ?RRSAF   003F 11
```

For JDBC connections the correlation id is the name of the job.

Example:

```
NAME    ST A   REQ   ID         AUTHID   PLAN     ASID TOKEN
RRSAF   T  *   3     BBOASR1S   CBASRU1  DSNJDBC  0039 438
```

Rules for LDAP security

You can control access to LDAP directories, subdirectories, or entries by means of access control lists (ACLs). ACLs specify which users are allowed access to each LDAP entry and which types of operations those users may perform. For details, see *z/OS Security Server LDAP Server Administration and Use*, SC24-5923. Follow these rules regarding LDAP security:

- For CORBA (MOFW) components, IBM has configured the LDAP DLLs to run within the Naming and Interface Repository server instances, thus eliminating the need to have a separate LDAP server running with the WebSphere for z/OS run time.

  If you do not follow the standard configuration of running the LDAP DLLs in the Naming and Interface Repository server instances and rely on an LDAP server running with WebSphere for z/OS run time, do not implement ACL-based access control to WebSphere for z/OS data. If you do implement ACL-based access control with such a configuration, WebSphere for z/OS will not be able to access its data.

- You can use RACF user IDs in LDAP Access Control Lists.

  Example: If USER1 is a RACF user id, use the following ACL statement. It gives USER1 the maximum access rights to the specified LDAP entry.

  ```
  aclSource: cn=DEPT_A, o=IBM, c=US
  aclEntry: access-id:USER1:object:ad:normal:rwsc
  ```

  You cannot, however, use RACF group names in this way. For more information about this and how LDAP can access the RACF database, see *z/OS Security Server LDAP Server Administration and Use*, SC24-5923. If you use group names, your installation must place WebSphere for z/OS libraries, DB2 libraries, and SYS1.LINKLIB under program control.

  Recommendation: For your initial LDAP configuration, we recommend you do not set up LDAP with RACF group names.

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 or OS/390'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:

1. For real storage requirements, see "z/OS or OS/390 hardware requirements" on page 10.

2. We recommend you dynamically load the run time in the link pack area (LPA) because the size of the load modules are large, and many address spaces need to refer to those load modules. The load modules for the run time comprise about 200 MB in size.

   Because you are using dynamic LPA, you will run out of ECSA after an IPL if you do not increase CSA at IPL time. You should monitor ECSA after dynamically loading the run time into LPA. Remember to increase the size of your CSA page data set accordingly.

3. If you choose to place the load modules in steplib or in the link list, you must allow for the additional 200 MB as part of each address space's region. A typical WebSphere for z/OS basic installation consists of 9 address spaces, each of which reference most of the 200 MB of load modules.

4. In addition to placing the load modules in the link pack area, give each address space a dynamic area of at least 128 MB.

5. 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 328 MB (200 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 change, if necessary.

Additional information about tuning your application's memory usage is in "Implementing advanced performance controls" on page 263.

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 to display trace data in trace data sets. WebSphere for z/OS identifies itself to CTRACE with the component name "SYSBBOSS". 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.
- Allow trace data to wrap or not wrap, allowing better management of system resources.
- 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 for servers and server instances. 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 V4.0.1 for z/OS and OS/390: Messages and Diagnosis*, GA22-7837. 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 51
Plan for and set up the WebSphere for z/OS error log stream	*z/OS MVS Setting Up a Sysplex*, SA22-7625 Table 18 on page 77
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
Define access authorization to system logger resources for the WebSphere for z/OS error log stream	Table 18 on page 77
Define the WebSphere for z/OS error log stream	Table 18 on page 77
View the WebSphere for z/OS error log stream	*WebSphere Application Server V4.0.1 for z/OS and OS/390: Messages and Diagnosis*, GA22-7837
Learn about how Java applications can log messages and trace data in the error log stream	*WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836

For details about problem diagnosis, see *WebSphere Application Server V4.0.1 for z/OS and OS/390: Messages and Diagnosis*, GA22-7837.

Post-installation notes on the error log

After the installation bootstrap is complete, use the Administration application to change the log stream name or create new log stream names for servers or server instances.

Notes:

1. A server error log stream setting overrides the general WebSphere for z/OS setting, and a server instance setting overrides a server setting. Thus, you can set up general error logging, but direct error logging for servers or server instances to specific log streams.

2. If you create a new log stream name through the Administration application, you must configure a new log stream on z/OS or OS/390 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 Administration application, 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 or OS/390 client to be recorded in an error log stream, code the CLIENTLOGSTREAMNAME environment variable in its environment file, then initialize the client. For more information about CLIENTLOGSTREAMNAME, see "Appendix A. Environment files" on page 303.

Our RACF sample BBOCBRAK gives UPDATE authority to the run-time control and server region user IDs for the log stream you created (it requires 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 BBOCBRAK.

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 environment 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 V4.0.1 for z/OS and OS/390: Messages and Diagnosis*, GA22-7837.

Recommendation for dumps

Plan as you would normally for system dumps. Due to the sized 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 214.

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 57 tells you about things 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 59 tells you where to find information about installing the product code.
3. "Running the customization dialog" on page 61 explains how to run the customization dialog.
4. "Following the customized instructions" on page 106 explains how to follow the instructions generated by the customization dialog to complete the system setup and run the bootstrap jobs.
5. "Installing the Administration and Operations applications" on page 111 provides information about installing the Administration and Operations applications, which are workstation programs used to define and operate servers.
6. "Defining application servers for the installation verification programs" on page 114 gives you instructions on how to run the Administration application to create the BBOASR2 and BBOASR1 servers used for the installation verification programs.
7. "Running the WebSphere for z/OS installation verification programs (IVPs)" on page 192 gives you instructions on how to run the installation verification programs.

If you encounter problems during installation and customization, refer to *WebSphere Application Server V4.0.1 for z/OS and OS/390: Messages and Diagnosis*, GA22-7837, 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, install the Administration and Operations applications, define application servers for the installation verification programs, and run the installation verification programs.

You can find background information about preparing z/OS or OS/390 subsystems in "Chapter 2. Preparing the base z/OS or OS/390 environment" on page 9.

For information about installing the product code through SMP/E, see *WebSphere Application Server V4.0.1 for z/OS and OS/390: Program Directory*, GI10-0680.

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.

The Administration and Operations applications are workstation-based applications that allow you to define and operate servers. During the installation and customization process, you will install these applications, then use them to define servers that run the installation verification programs.

IBM provides installation verification programs that test Web applications and server components, such as enterprise beans. At the end of installation and customization, you will run one or more of these programs.

The following table outlines the installation and customization process:

Stage	Description
1	Install prerequisite products, such as DB2 Version 7. 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 V4.0.1 for z/OS and OS/390: Program Directory* (if you use CBPDO) or *ServerPac: Installing Your Order* (if you use ServerPac).
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. The instructions take you through a process we call the bootstrap. When you finish, you have a complete WebSphere for z/OS run-time configuration.
5	Install the workstation-based Administration and Operations applications (also called the System Management User Interface).

Stage	Description
6	With the Administration application, create the J2EE or CORBA (MOFW) server definitions. These servers are used by the installation verification programs and are examples of application servers you will create for your own applications.
7	Run the installation verification programs to verify 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 monoplex 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 or OS/390 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 or OS/390 subsystems before you start customization.

Steps for preparing your z/OS or OS/390 subsystems

Before you begin: Read "Chapter 1. Overview of installation and customization" on page 1.

Follow these steps:

1. Prepare your z/OS or OS/390 subsystems (see "Chapter 2. Preparing the base z/OS or OS/390 environment" on page 9). 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 10.
   - 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 19.
   - Workload manager (WLM). See "Setting up workload management (WLM)" on page 35.
   - Resource Recovery Services. See "Recommendations for resource recovery services" on page 41.
   - DB2. For background, guidelines, and rules about DB2 and LDAP (which you will install in this chapter, should you not have one installed), see "DB2 database and LDAP" on page 42.

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). The user ID must have the ability to create DB2 tables.

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

Follow the *WebSphere Application Server V4.0.1 for z/OS and OS/390: Program Directory*, GI10-0680 (if you use CBPDO) or *ServerPac: Installing Your Order* (if you use ServerPac), to install the code through SMP/E.

Notes:

1. You can change the high-level qualifier of the installed data sets (not recommended) or the middle-level qualifier. In this book we use 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.

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/WebSphere, or at similar mount points of your choice.

Steps for collecting information for customization

This procedure prepares you for the WebSphere for z/OS customization dialog. By recording important information in the worksheets later in this chapter, you will make important decisions about the information you will enter into the dialog.

Before you begin: You must know or be able to find the system characteristics for the system on which WebSphere for z/OS will run.

Perform these steps:

1. Read through the section "Running the customization dialog" on page 61 (through "L Load Customization Variables" on page 105) to get acquainted with the customization dialog and the information you need to supply to it.

   _____

2. Fill in the worksheets in "Running the customization dialog" on page 61 (through "L Load Customization Variables" on page 105) with the customization values you will use.

   Note: The worksheets follow the order of the customization dialog panels. Titles in the worksheet match panel titles in the dialog.

   _____

You are done when you have completed filling in the worksheets.

Running the customization dialog

The customization dialog is intended for the system programmer or administrator responsible for installing and customizing WebSphere for z/OS. The dialog is intended to be used only once, for the first time you customize the product.

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 system management HFS structure and the initial environment file
- Create and customize the LDAP server
- Set up WebSphere for z/OS security controls (RACF)
- Define the WebSphere for z/OS run-time configuration (systems management server, naming server, interface repository server, daemon server)
- Run the installation verification programs (IVPs)

Steps for running 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 a minimum of 32 rows by 80 columns (32 x 80) in order for the ISPF customization dialog to run.
- You cannot use the split screen function in ISPF. Using split screen will obscure lines at the bottom of the dialog panels.

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.

hlq(*value***)**
> Specifies the data set qualifier(s) for the WebSphere for z/OS product data sets. The default value is BBO.

appl(*value***)**
> Specifies the ISPF application name. The default value is BBO.

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

Example:

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

Result: You see the splash screen:

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


     WebSphere Application Server V4.0.1 for z/OS and OS/390
     Licensed Material - Property of IBM

     5655-F31 (C) Copyright IBM Corp. 2001
     All Rights Reserved.
     U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
     Disclosure restricted by GSA-ADP schedule contract with IBM Corp.

     Status  = H28W401

     Version = 4.01.001


                         Press ENTER to continue.

```

2. Press Enter.

   Result: You see the following panel:

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

  Use this dialog to customize WebSphere for z/OS for the first time
  or to migrate releases. Specify an option and press ENTER.


  1  New customization. If you are customizing WebSphere for
     z/OS for the first time, use this option.

  2  Migration with saved variables. If you have previously saved the
     customization variables using the dialog, use this option to
     migrate from WebSphere for z/OS V4.0 to V4.0.1.

  3  Migration without saved variables. If you have never run the
     customization dialog, or have not previously saved the
     customization variables, use this option to migrate from
     WebSphere for z/OS V4.0 to V4.0.1.

```

3. Choose option 1 and press Enter.

   Result: You see the main dialog panel:

```
-----------------      WebSphere for z/OS Customization     -----------------
Option  ===>                                                Appl: BBO
 New Customization
   Use this dialog to define WebSphere for z/OS variables and generate
   customization jobs for your installation.  Specify the HLQ for
   WebSphere product data sets, 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 the generated customization instructions.



   Options for WebSphere 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.
```

You have finished starting the customization dialog.

Steps for allocating the target data sets

Before you begin: You need to start the customization dialog.

Perform the following steps to allocate the target data sets:

1. On the main dialog panel, type 1 in the `Option` field.

   _____

2. If you did not specify a high-level qualifier when you started the customization dialog, type one in the `HLQ for WebSphere product data sets` field.

   _____

3. Press Enter.

   _____

4. On the Allocate Target Data Sets panel, type in the information from "1 Allocate Target Data Sets" on page 65, then press Enter.

   _____

You are done when the data set allocation succeeds.

1 Allocate Target Data Sets

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.

Steps for defining variables

Before you begin: You must start the customization dialog.

Perform the following steps to define variables:

1. On the main dialog panel, type 2 in the Option field.

   _____

2. Press Enter.

   Note: If this is the first time through the dialog, you see the Load
   Customization Variables panel. Press enter to load the default
   WebSphere for z/OS variable settings from the product data sets
   and continue.

   Result: You see:

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

   Define Variables

      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.


                                           Changed?
      1 - System Locations (directories, HLQs, etc)
      2 - WebSphere Customization
      3 - Server Customization
      4 - IVP Customization
      5 - LDAP Customization
      6 - Security Customization
   ```

   _____

3. Follow the options in order and enter information from "2 Define
   variables" on page 67.

   _____

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

2 Define variables

System locations (directories, HLQs, etc):

Table 14. System Locations (1 of 3)

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)	
SGLDLNK	GLD.SGLDLNK	
		In link list or LPA?
SCEERUN	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?
SDSNLOAD	DSN710.SDSNLOAD	
		In link list or LPA?
SDSNLOD2	DSN710.SDSNLOD2	
		In link list or LPA?
SDSNEXIT	DSN710.SDSNEXIT	
		In link list or LPA?
SDSNDBRM	DSN710.SDSNDBRM	
RUNLIB.LOAD	DSN710.RUNLIB.LOAD	
SBBOLPA	BBO.SBBOLPA	
SBBOULIB	BBO.SBBOULIB	
SBBODBRM	BBO.SBBODBRM	
SBBOEXEC	BBO.SBBOEXEC	
SBBOMSG	BBO.SBBOMSG	

Worksheet

This panel asks you for information about your base operating system, HFS-resident components, and DB2 subsystem.

System name
> The system name for the target z/OS or OS/390 system on which WebSphere for z/OS is installed.

Sysplex name
> The sysplex name for the target z/OS or OS/390 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 or OS/390 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 or OS/390 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 LDAP, Language Environment, DB2, 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.

SGLDLNK
Your existing LDAP run-time load module library.

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.

SDSNLOAD
Your existing DB2 run-time load module library PDS.

SDSNLOD2
Your existing DB2 run-time load module library PDSE.

SDSNEXIT
Your existing DB2 installation exits load module library.

Specify the following DB2 and WebSphere for z/OS libraries so they can be accessed by the customized job streams the dialog produces. These data sets must be cataloged.

SDSNDBRM
Your existing DB2 DBRM library.

RUNLIB.LOAD
Your existing DB2 sample application load module library.

SBBOLPA
WebSphere for z/OS data set you installed through SMP/E. Its members must go into the LPA.

SBBOULIB
WebSphere for z/OS unauthorized load module library you installed through SMP/E.

SBBODBRM
WebSphere for z/OS DBRM library you installed through SMP/E.

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

Worksheet

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

Table 15. System Locations (2 of 3)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Locations of HFS resident components		
Java home directory	/usr/lpp/java/IBM/J1.3	
LDAP home directory	/usr/lpp/ldap	
DB2 home directory	/usr/lpp/db2/db2710	
WebSphere home directory	/usr/lpp/WebSphere	
NLS Path	/usr/lib/nls/msg/En_US.IBM-1047/%N	

Locations of HFS resident components:

Java home directory
> Your existing Java SDK library path.

LDAP home directory
> Your existing LDAP library path.

DB2 home directory
> Your existing DB2 library path.

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

NLS Path
> NLS library path used in the NLSPATH environment variable. The directory must either be /usr/lib/nls/msg/En_US.IBM-1047/%N or /usr/lib/nls/msg/C/%N.

Worksheet

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
DB2 information		
DB2 version number	7	
DB2 subsystem name	DB2	
DB2 location name	LOC1	
Tablespace buffer pool	BP0	
Index buffer pool	BP0	
32K tablespace buffer pool	BP32K	
DB2 jdbc properties file	/usr/lpp/db2/db2710/classes /db2sqljjdbc.properties	
System Management Database		
Database name	BBOMDB01	
First STOGROUP for database	BBOMG01	
Volume for first STOGROUP	*	
Second STOGROUP for database	BBOMG02	
Volume for second STOGROUP	*	
Stateful Session Bean Database		
Database name	BBOJDB01	
First STOGROUP for database	BBOJG01	
Volume for first STOGROUP	*	
Second STOGROUP for database	BBOJG02	
Volume for second STOGROUP	*	

DB2 Information

DB2 version number
> The version of DB2 you use for WebSphere for z/OS. Currently, this version must be version 7.

DB2 subsystem name
> The name of your DB2 subsystem.

DB2 location name
> Your DB2 data source location name. This is set in the DB2 installation job DSNTIJUZ.

Tablespace buffer pool
> Name of your DB2 tablespace buffer pool.

Note: If your installation does not allow user data to be placed in BP0, specify an appropriate buffer pool. Ask your DB2 administrator for the correct value.

Index buffer pool
    Name of your DB2 index buffer pool.

Note: If your installation does not allow user data to be placed in BP0, specify an appropriate buffer pool. Ask your DB2 administrator for the correct value.

32K tablespace buffer pool
    Name of your DB2 virtual 32K buffer pool.

DB2 jdbc properties file
    Specify the path and name of your SQLJ/JBDC run-time properties file. If you customized this file, you may want to keep the customized version in a separate directory, such as /etc. If so, override the default setting.

    Example:

    `/usr/lpp/db2/db2710/classes/db2sqljjdbc.properties`

    This file is shipped as part of the DB2 JDBC feature, and must be modified during JDBC customization. See *DB2 for OS/390 Application Programming Guide and Reference for Java*.

System Management Database

Database name
    Specifies the database name for the WebSphere for z/OS system management database.

First STOGROUP for database
    Specifies the first storage group for the database.

Volume for first STOGROUP
    Specifies either the DASD volume serial number that will contain 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.

Second STOGROUP for database
    Specifies the second storage group for the database.

Volume for second STOGROUP
    Specifies either the DASD volume serial number that will contain 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.

Stateful Session Bean Database

Database name
>   The database name for the database that supports stateful session beans.

First STOGROUP for database
>   Specifies the first storage group for the database.

Volume for first STOGROUP
>   Specifies either the DASD volume serial number that will contain 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.

Second STOGROUP for database
>   Specifies the second storage group for the database.

Volume for second STOGROUP
>   Specifies either the DASD volume serial number that will contain 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.

WebSphere customization:

Table 17. WebSphere Customization (1 of 4)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
WebSphere configuration HFS information		
Mount point	/WebSphere390/CB390	
Name	OMVS.WAS.CONFIG.HFS	
Volume, or '*' for SMS	*	
Primary allocation in cylinders	40	
Secondary allocation in cylinders	20	
WebSphere DB2 information		
VCAT value	DSN710	
Data volume, or '*' for SMS	*	
Index volume, or * for SMS	*	
WebSphere Namespace Information		
CORBA name space root	o=BOSS,c=US	
JNDI name space root	o=WASNaming,c=US	

WebSphere configuration HFS Information

Mount point
> Read/write HFS directory mount point where application data and environment files will be written. The customization process creates this mount point.

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

Volume, or '*' for SMS
> Specifies either the DASD volume serial number that will contain 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 40 cylinders (3390).

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

Worksheet

WebSphere DB2 information

VCAT value

High-level qualifier for DB2 table spaces. This existing VCAT value is used to create storage groups.

Data volume, or '*' for SMS

Specify either the DASD volume serial number for the WebSphere for z/OS DB2 tables storage group 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.

Index volume, or '*' for SMS

Specify either the DASD volume serial number for the WebSphere for z/OS DB2 tables storage group 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.

WebSphere Namespace Information

Rules:

- For the following name space roots, you must specify at least the "o=..." parameter.
- Values can be mixed case. Use lowercase characters for the parameter names.

CORBA name space root

Root naming context that will be created for CORBA (MOFW) components.

JNDI name space root

Starting point of the WsnName tree context that will be created for J2EE components.

Table 18. WebSphere 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	(sysplex 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 for z/OS error log stream that will be created.

> **Rules:**
> - The name must be 26 characters or less.
> - 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 will be created.

Worksheet

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 sysplex 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".

Table 19. WebSphere for z/OS Customization (3 of 4)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
WebSphere for z/OS Common Groups and User IDs		
Control region group for ALL servers	CBCTL1	
Control region GID for ALL servers	2211	
Server region group for base servers	CBSR1	
Server region GID for base servers	2201	
Unauthenticated User Definitions for Base Servers		
User ID	CBGUEST	
UID	2102	
Group	CBCLGP	
GID	2202	
WebSphere for z/OS Application Installer Group Information		
Group	CBCFG1	
GID	2300	
WebSphere for z/OS Administrator Information		
User ID	CBADMIN	
UID	2103	
Password	CBADMIN	
Group	CBADMGP	
GID	2203	

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 will be at the group level rather than the user ID level. In a later panel, the dialog asks for user IDs for the run-time server instances. These user IDs will be connected to their proper RACF groups.

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

Worksheet

On the other hand, server regions 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 run-time servers only. (You will have to create these RACF definitions for your own application server regions.)

Rules:

- User IDs and groups must be unique names (one to seven 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 Common Groups and User IDs

Control region group for ALL servers
    A group name that the dialog uses for all control regions.

Control region GID for ALL servers
    A group identifier that the dialog uses for all control regions.

Server region group for base servers
    A group name that the dialog uses for the WebSphere for z/OS run-time server regions.

Server region GID for base servers
    A group identifier that the dialog uses for the WebSphere for z/OS run-time server regions.

Unauthenticated User Definitions for Base Servers

Userid If you allow unauthenticated client requests, this is the default user ID under which those requests run.

UID   The user identifier for the unauthenticated user.

Group   The group for unauthenticated users.

GID   The group identifier for unauthenticated users.

WebSphere Application Installer Group Information

Group   The group name for all users that install applications. This group allows you to manage application installers' authorities more easily.

GID   The group identifier for application installers.

WebSphere Administrator Information

Userid The user ID for your initial administrator. This administrator uses the Administration and Operations applications.

UID The user identifier for the initial administrator.

Password
  The password that the initial administrator uses to log onto the Administration and Operations applications.

Group The group name for all your administrators.

GID The group identifier for all your administrators.

Worksheet

Table 20. WebSphere for z/OS 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.*system*.WAS390.CTRACE	
Volume, or "*"for SMS	*	
Primary space in cylinders	20	
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 PROCLIB.

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.

>   **Rule:** 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.

Worksheet

Server Customization: The WebSphere for z/OS run time requires four base system servers and their server instances: Daemon, System Management, Naming, and Interface Repository. The panels corresponding to Table 21 through Table 24 on page 89 set up the names, start procedures, and user IDs for the base servers.

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, the start procedure for each base server control region and server region must have a user ID and will be defined in the STARTED class. For more information, see "Server authorizations" on page 21.

Table 21. Server Customization (1 of 4)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Daemon definitions		
Server name	CBDAEMON	
Server instance	DAEMON01	
Procedure name	BBODMN	
Userid	CBDMNCR1	
UID	2111	
Port	5555	
IP name	(null)	
SSL Port	5556	

The 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 sysplex.

The **Server name** is the generic server name used for all Daemon instances in the sysplex. The server name is used in security profiles to control a client's access to this server.

The **Server instance** name is the specific server instance name for the Daemon on your target z/OS or OS/390 system. The server instance is associated with a single control region, which in turn has a **Procedure name** and a control region **Userid** and **UID**.

Specify the TCP/IP **Port** and **IP name** at which the Daemon listens for incoming connections. The port and IP name are sometimes called the *Daemon port* and *Daemon IP name*. For advice on appropriate settings for the port and

IP name, see "Updating your TCP/IP network" on page 16. Also specify an additional **SSL port** at which the Daemon listens for incoming SSL connections.

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

Worksheet

Table 22. Server Customization (2 of 4)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
System Management server definitions		
Server name	CBSYSMGT	
Server instance	SYSMGT01	
Control region information		
Procedure name	BBOSMS	
Userid	CBSYMCR1	
UID	2112	
Server region information		
Procedure name	BBOSMSS	
Userid	CBSYMSR1	
UID	2104	
Resolve IP port	900	
Resolve IP name	(same as Daemon)	

The System Managment server manages configuration data for all servers and interacts with the Administration and Operations applications (SM GUI) on Windows to install and configure application servers.

The **Server name** is the generic server name used for all system management instances in the sysplex. The **Server instance** name is the specific server instance name for the system management server on your target z/OS or OS/390 system.

The server instance consists of a single control region and one or more server regions. The control region is asssociated with a **Procedure name** and a control region **Userid** and **UID**. The server regions share a common **Procedure name**, server region **Userid** and **UID**.

Specify a TCP/IP port for the **Resolve IP port** and an IP name for the **Resolve IP name** at which the system management server listens for incoming connections. For advice on appropriate settings for the Resolve port and IP name, see "Updating your TCP/IP network" on page 16. By default, the customization dialog sets the Resolve IP name to be the same as the Daemon IP name.

Note: Select the IP name and port number for the SM server carefully. You can choose any name you want, but once chosen, it is difficult to

change, even in the middle of customization.

Worksheet

Table 23. Server Customization (3 of 4)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Interface Repository server definitions		
Server name	CBINTFRP	
Server instance	INTFRP01	
Control region information		
Procedure name	BBOIR	
Userid	CBINTCR1	
UID	2114	
Server region information		
Procedure name	BBOIRS	
Userid	CBINTSR1	
UID	2106	

The Interface Repository server manages the inventory of CORBA business object interfaces for predicate evaluation queries.

The **Server name** is the generic server name used for all interface respository instances in the sysplex. The **Server instance** name is the specific server instance name for the interface respository server on your target z/OS or OS/390 system.

The server instance consists of a single control region and one or more server regions. The control region is asssociated with a **Procedure name** and a control region **Userid** and **UID**. The server regions share a common **Procedure name**, server region **Userid** and **UID**.

Table 24. Server Customization (4 of 4)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Naming server definitions		
Server name	CBNAMING	
Server instance	NAMING01	
Control region information		
Procedure name	BBONM	
Userid	CBNAMCR1	
UID	2113	
Server region information		
Procedure name	BBONMS	
Userid	CBNAMSR1	
UID	2105	
Keyring	CBKeyring	

The Naming server provides applications with the capability to register and to find names for object references. Services are implemented through LDAP tables and JNDI services.

The **Server name** is the generic server name used for all naming instances in the sysplex. The **Server instance** name is the specific server instance name for the naming server on your target z/OS or OS/390 system.

The server instance consists of a single control region and one or more server regions. The control region is asssociated with a **Procedure name** and a control region **Userid** and **UID**. The server regions share a common **Procedure name**, server region **Userid** and **UID**.

Specify the name of the *client's* **Keyring** used in SSL processing. This key ring must reside in RACF. For more information, see "Setting up SSL security for WebSphere for z/OS" on page 243.

Worksheet

IVP Customization: This part of the dialog (corresponding to Table 25 on this page and Table 26 on page 92) asks you for information about the application servers used for the installation verification programs (IVPs). Though the dialog asks you for information for both the CORBA and J2EE application servers, you can choose later to run either IVP or both, depending on which server type you plan to have.

Table 25. IVP Customization (1 of 2) - CORBA Server

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Server name	BBOASR1	
Server instance	BBOASR1A	
Control region information		
Procedure name	BBOASR1	
Userid	CBACRU1	
UID	2107	
Server region information		
Procedure name	BBOASR1S	
Userid	CBASRU1	
UID	2110	
Group	CBASR1	
GID	2205	
Unauthenticated user information		
Local userid	CBIVP	
UID	2109	
Remote userid	CBIVP	
UID	2109	
Group	CBIVPGP	
GID	2209	
User ID to run the IVP		
Userid	CBIVP	
UID	2109	
Password	CPIVP	
Server key ring	CBKeyring	
Client key ring	CBKeyring	
Location of script	/tmp	

The installation verification program for the CORBA (MOFW) run-time application server consists of a simple business object called Policy that uses the following server definitions.

The **Server name** is the generic server name used for all instances of the CORBA IVP application server in the sysplex. The **Server instance** name is the specific server instance name for the CORBA IVP server on your target z/OS or OS/390 system.

The server instance consists of a single control region and one or more server regions. The control region is asssociated with a **Procedure name** and a control region **Userid** and **UID**. The server regions share a common **Procedure name**, server region **Userid** and **UID**, and server region **Group** and **GID**.

Unauthenticated clients on the same z/OS or OS/390 system as WebSphere for z/OS run under the **Local userid**, with its associated **UID**. Unauthenticated clients running on remote systems run under the **Remote userid** and associated **UID**. Both user IDs are associated with a **Group** and **GID**.

The CORBA (MOFW) IVP runs as a batch job. For the user ID that runs the IVP, specify the **Userid**, **UID** and **Password** to be used for running this job.

Specify a **Server Keyring** and **Client Keyring** to be used for SSL processing. Both key rings must reside in the RACF database. For more information on SSL, see "Setting up SSL security for WebSphere for z/OS" on page 243.

Specify a read/write HFS directory where the dialog will place the IVP shell script. You may want to specify a different directory, such as /tmp/CBIVP, to segregate the files for the CORBA IVP. If you do specify a different directory, you must create this directory before running the WebSphere for z/OS customization batch jobs. Make sure the directory is owned by the user ID used to run the IVP and give the directory file permissions of 755.

Worksheet

Table 26. IVP Customization (2 of 2) - J2EE Server

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Server name	BBOASR2	
Server instance	BBOASR2A	
Control region information		
Procedure name	BBOASR2	
User ID	CBACRU2	
UID	2115	
Server region information		
Procedure name	BBOASR2S	
User ID	CBASRU2	
UID	2116	
Group	CBASR2	
GID	2216	
Unauthenticated user information		
Local user ID	CBIVP2	
UID	2117	
Remote user ID	CBIVP2	
UID	2117	
Group	CBIVPGP2	
GID	2217	
User ID to run the IVP		
User ID	CBIVP2	
UID	2117	
Password	CPIVP2	
Server key ring	CBKeyring	
Client key ring	CBKeyring	
Location of script	/tmp	

The installation verification program for the J2EE run-time application server consists of a simple business object called Policy that uses the following server definitions.

The **Server name** is the generic server name used for all instances of the J2EE IVP application server in the sysplex. The **Server instance** name is the specific server instance name for the J2EE IVP server on your target z/OS or OS/390 system.

The server instance consists of a single control region and one or more server regions. The control region is asssociated with a **Procedure name** and a control region **Userid** and **UID**. The server regions share a common **Procedure name**, server region **Userid** and **UID**, and server region **Group** and **GID**.

Unauthenticated clients on the same z/OS or OS/390 system as WebSphere for z/OS run under the **Local userid**, with its associated **UID**. Unauthenticated clients running on remote systems run under the **Remote userid** and associated **UID**. Both user IDs are associated with a **Group** and **GID**.

One of the J2EE IVPs runs as a batch job. For the user ID that runs the IVP, specify the **Userid**, **UID** and **Password** to be used for running this job.

Specify a **Server Keyring** and **Client Keyring** to be used for SSL processing. Both key rings must reside in the RACF database. For more information on SSL, see "Setting up SSL security for WebSphere for z/OS" on page 243.

Specify a read/write HFS directory where the dialog will place the IVP shell script. You may want to specify a different directory, such as /tmp/CBIVP2, to segregate the files for the J2EE IVP. If you do specify a different directory, you must create this directory before running the WebSphere for z/OS customization batch jobs. Make sure the directory is owned by the user ID used to run the IVP and give the directory file permissions of 755.

Worksheet

LDAP Customization: LDAP (Lightweight Directory Access Protocol) provides the directory services for the Java Naming and Directory Interface (JNDI) and CORBA naming and interface repository services. For the WebSphere for z/OS implementation, LDAP stores its naming data in a DB2 database.

The dialog provides most of the LDAP configuration data, but you must specify the LDAP server definitions and some configuration information on this panel.

Table 27. LDAP Customization (1 of 1)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
LDAP Server Definitions		
Procedure name	BBOLDAP	
User ID	CBLDAP	
UID	2119	
Group	CBLDAPGP	
GID	2219	
LDAP Configuration Information		
IP name	(same as Daemon)	
IP port	1389	
Administrator user DN	cn=CBAdmin	
Administrator user pw	secret	
DB2 STOGROUP value	BBOLDSTO	
DB2 database name	BBOLDAP	
Authid for DB2 tables	BBOLDAP	
LDAP Tablespace Information		
Tablespace for LDAP entry table	BBOENT	
Tablespace for 32K tables	BBO32K	
Tablespace for 4K tables	BBO4K	
Tablespace for 4K mutex table	BBOMUTX	

LDAP Server Definitions

Procedure name
> The name of the procedure to be created in your procedure library that starts the LDAP server.

Userid   The SAF-defined user ID to be created and associated with the LDAP start procedure.

UID   The user identifier associated with this user ID.

Group   The SAF-defined group name for the LDAP server.

GID   The group identifier associated with this group.

LDAP Configuration Information

IP Name
> The fully-qualified IP host name of the system on which the LDAP server will run.

IP Port
> An existing IP port to be used by the LDAP server.

Administrator user DN
> The distinguished name of the LDAP administrator to be created.

Administrator user pw
> The password to be created for this user.

DB2 STOGROUP value
> The name of the default storage group to be created for the LDAP database.

DB2 database name
> The name of the LDAP database to be created in DB2.

Authid for DB2 tables
> The user ID to be created and granted access to the LDAP tables.

LDAP Tablespace Information

Tablespace for LDAP entry table
> Specifies the partitioned table space name which is to be used when creating the LDAP entry table.

Tablespace for 32K tables
> Specifies the segmented table space name that is to be used when creating 32K tables.

Tablespace for 4K tables
> Specifies the segmented table space name that is to be used when creating 4K tables.

Worksheet

Tablespace for 4K mutex table
　　　Specifies the nonsegmented table space name that is to be used when
　　　creating the LDAP 4K mutex table.

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 19.

Table 28. Security Customization (1 of 1)

Item	Value in the dialog after you load IBM defaults	Your value (Fill in the blanks)
Enable SETROPTS GRPLIST	Y	
Use DSNR class to control DB2 access	N	
Use SOMDOBJS class to control CORBA method access	N	
Use EJBROLE class to control EJB method access	N	
Use OPERCMDS to control commands	N	
Use DCE for authentication and encryption	N	
Use Kerberos over SSL	N	
Use SSL basic authentication	N	
Use SSL client certificates	N	
Test certificate authority label	WAS TestCertAuth	
PassTicket profile name	CB390	(cannot change)
PassTicket KEYMASK value	0123456789ABCDEF	

In the following, specifying "Y" (yes) tells the dialog to define the profile or control in RACF. Specifying "N" (no) tells the dialog not to define the profile or control.

Enable SETROPTS GRPLIST
> Enables the GRPLIST option in RACF.

> **Note:** Without the GRPLIST option enabled, the customized RACF job produced by the dialog will not run successfully and there will be problems accessing HFS files needed for configuration of the servers. If you cannot enable GRPLIST on your system, contact IBM service for help in defining the authorizations you need.

Use DSNR class to control DB2 access
> Specify "Y" to create the RACF DSNR class. The class helps you centralize DB2 security management through RACF.

Worksheet

Use SOMDOBJS class to control CORBA method access
> Specify "Y" to create the RACF SOMDOBJS class. The class controls method accesses for CORBA objects.

Use EJBROLES class to control EJB method access
> Specify "Y" to create the RACF EJBROLES class. EJBROLES controls method access 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 DCE for authentication and encryption
> Specify "Y" to create DCE (Distributed Computing Environment) security. You can use DCE to create secure communications in a network.

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.

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

Steps for generating customization jobs

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 102.

Perform the following steps to generate the customization jobs:

1. On the main dialog panel, type 3 in the `Option` field.

   _____

2. Press Enter.

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

   _____

3. Fill in the job card information according to "3 Generate customization jobs" on page 100, then press Enter.

   _____

You are done when all the jobs are generated.

Worksheet

3 Generate customization jobs

Table 29. 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.

Steps for viewing the generated customization instructions

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.

   _____

2. Press Enter.

   _____

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.

Steps for saving the customization variables

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.

   _____

2. Press Enter.

   Result: You see the Save Customization Variables panel.

   _____

3. Type in the information from "S Save Customization Variables" on page 103, then press Enter.

   _____

You are done when you successfully save the variables.

S Save Customization Variables

Table 30. 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.

Steps for loading customization variables

When you first run the customization dialog, you must load the customization variables. IBM provides member BBOWVARS in the SBBOEXEC data set, which allows you to prime the dialog with default values.

You may also follow these instructions to re-load variables you have saved.

Before you begin: You must start the customization dialog.

Perform the following steps to load customization variables:
1. On the main dialog panel, type L in the Option field.

   _____

2. Press Enter.
   Result: You see the Load Customization Variables panel.

   _____

3. Type in the information from "L Load Customization Variables" on page 105, then press Enter.

   _____

You are done when you successfully load the variables.

L Load Customization Variables

Table 31. 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 instructions

The customization dialog generates instructions tailored for your system. The procedure in this section explains how to follow those instructions to customize your system and run the bootstraps.

Rules:

1. If you created the target data sets (*.CNTL and *.DATA) on another (driving) system, you must copy them to the **target system** and give them the same data set names.

2. **You must perform the instructions on your target system.**

The major steps of the customization process are:

Stage	Description
1	Make a variety of configuration changes to your z/OS or OS/390 system configuration (PARMLIB, TCP/IP, workload management, and so forth.). The customized instructions provide details and pointers to relevant documentation.
2	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.
3	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 BBOWBRAK of the *hlq*.DATA data set. Job BBOCBRAK executes these RACF commands. Later, you can use the RACF commands saved in BBOWBRAK to help in defining security for additional servers or users.
4	Create a customized set of RACF commands for the LDAP server security setup using job BBOLDRAJ. The RACF commands are saved in member BBOLDRAK of the *hlq*.DATA data set. Job BBOLDRAK executes these RACF commands.
5	Allocate and mount the WebSphere for z/OS system management 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 and files in the WebSphere for z/OS system management HFS. For more information about the resulting HFS structure, see "Overview of BBOMCFG" on page 107.
6	Copy customized PARMLIB and PROCLIB members, as well as customized HFS files, into their proper locations using job BBOWCPY1.

Stage	Description
7	Define the DB2 tables and objects used by the WebSphere for z/OS system management server through job BBOMCRDB. DB2 and RRS must be running before you run BBOMCRDB. Job BBOBIND creates package bindings for this database.
8	Define the DB2 tables needed by the WebSphere for z/OS LDAP server through job BBOLDTBC. Jobs BBO1JCL and BBO2JCL bind packages and plans for this database. In certain cases, job BBO1JCL should NOT be run. For details, see the discussion about BBO1JCL on page 46. If you need to recreate the LDAP tables with BBOLDTBC, run job BBODTBD first to delete the old ones. Running BBOLDTBD requires you to redo the entire customization process again.
9	Set up DB2 authorizations for the WebSphere for z/OS system management and LDAP databases through jobs BBOCBGRT and BBOLDGRT.
10	Prime the LDAP tables with the initial entries needed for WebSphere for z/OS setup using job BBOLD2DB.
11	Run the bootstrap process. This process consists of bringing up the WebSphere for z/OS run–time servers for the first time, restarting them, running two configuration jobs (BBONMC and BBOIRC), and restarting the run time once more. Each phase of this bootstrap causes additional information to be stored in the WebSphere for z/OS databases and HFS. Once the bootstrap process is complete, you have a working WebSphere for z/OS run time that you will use to run the installation verification programs (IVPs) and your own applications.

Overview of BBOMCFG

The BBOMCFG job creates the system management HFS structure on a mount point for a WebSphere for z/OS file system. The mount point is specified by a variable called -TARGETDIR. The default -TARGETDIR is /WebSphere390/CB390.

Rules:

1. TARGETDIR must be a read/write directory. If you plan to set up WebSphere for z/OS in a sysplex, this directory must be shared, so you must establish some means of sharing the HFS in read/write mode across the sysplex. For OS/390 Version 2 Release 8, you must use the Network File System. For OS/390 Version 2 Release 9 or later and z/OS, you can choose either the Network File System or use the shared HFS function.

2. The System Management group (default CBCFG1) and user ID (default CBSYMSR1) must own each directory and subdirectory in TARGETDIR. If the System Management group and user ID do not own TARGETDIR, use the chown command to make them the owner of each directory and subdirectory in TARGETDIR. Thus, if you use the default TARGETDIR,

you must use the chown command to give the System Management group
and user ID ownership of /WebSphere390 and /WebSphere390/CB390.

Example:

```
chown -R CBSYMSR1:CBCFG1 /WebSphere390
```

The entire subdirectory structure looks like this:

```
/TARGETDIR
   /controlinfo
      /envfile
         /SYSPLEX
            /DAEMON01
               current.env -> /TARGETDIR/SYSPLEX/initial/configuration.env
            /INTFRP01
               current.env -> /TARGETDIR/SYSPLEX/initial/configuration.env
            /NAMING01
               current.env -> /TARGETDIR/SYSPLEX/initial/configuration.env
            /SYSMGT01
               current.env -> /TARGETDIR/SYSPLEX/initial/configuration.env
   /SYSPLEX
      /conversations
         /cb302
         /current
            configuration.xml -> /TARGETDIR/SYSPLEX/initial/configuration.xml
      /etc
         /ldap
               SYS1.bboldif.cb
               SYS1.bboslapd.conf
               SYS1.dsnaoini
      /initial
         configuration.env
         configuration.xml
         WebContainerDB2.xml
      /resources
         /templates
            DB2datasource.xml
            MQQueue.xml
            MQQueueConnectionFactory.xml
            MQRRSQueueConnectionFactory.xml
            MQRRSTopicConnectionFactory.xml
            MQTopic.xml
            MQTopicConnectionFactory.xml
      /temp
   /apps
      /SYSPLEX
```

where

TARGETDIR

Is the mount point you specify using a job variable called -TARGETDIR in
BBOMCFG.

SYSPLEX
Is the name of the sysplex on which your WebSphere for z/OS system runs. You specify the sysplex name in a job variable called -SYSPLEX.

The directories that are important for installation and customization are:

- *TARGETDIR/SYSPLEX*/initial. Environment files for the run-time servers are placed in this directory.
- *TARGETDIR/SYSPLEX*/etc/ldap. Custom LDAP configuration files are in this directory.

The following are variables in BBOMCFG. The customization dialog supplies values for these variables using values you supply in the dialog.

Table 32. Variables in job BBOMCFG

Variable	Explanation
-INSTALLDIR	The name of the directory where WebSphere for z/OS files reside after SMP/E installation
-TARGETDIR	The name of the WebSphere for z/OS mount point. -TARGETDIR is used as the base directory under which BBOMCFG sets up a directory structure that will hold all HFS-related configuration and application data. The value of -TARGETDIR must be the same as the value of the CBCONFIG environment variable specified in environment files and the CBCONFIG JCL variable used in the start procedures to startup WebSphere for z/OS servers. The value of -TARGETDIR should **not** be the same as -INSTALLDIR.
-SYSPLEX	The name of the monoplex or sysplex on which WebSphere for z/OS runs. You can obtain this value by entering the command D SYMBOLS on the system console.
-SYSNAME	The name of the z/OS or OS/390 system on which WebSphere for z/OS runs. You can obtain this value by entering the command D SYMBOLS on the system console.
-DM_NAME	The name of your initial Daemon server instance that will be used for the bootstrap
-IR_NAME	The name of your initial Interface Repository server instance that will be used for the bootstrap
-NM_NAME	The name of your initial Naming server instance that will be used for the bootstrap
-SM_NAME	The name of your initial System Management server instance that will be used for the bootstrap

Table 32. Variables in job BBOMCFG (continued)

Variable	Explanation
-OWNER	The user ID associated with the System Management server. It will be the owner of the HFS files.
-GROUP	The RACF group name for the HFS files. BBOCBRAK creates this group (the default is CBCFG1). The purpose of the group is to allow application installers to manage these HFS files without needing to be in the same RACF groups as the run-time server user IDs, particularly the system management server region user ID (CBSYMSR1), which owns the HFS directories.

Steps for following the customized instructions

Before you begin: You must run the customization job and generate the customized jobs.

Perform the following steps to follow the customized instructions.

1. Within the customization dialog, enter option 4 (View the generated customization instructions).

   _____

2. View or print the instructions.

   _____

3. Follow the instructions in the order they are presented.

   _____

You know you are done when you have finished the WebSphere for z/OS bootstrap process. The customized instructions will tell you to return to this manual to continue.

Installing the Administration and Operations applications

The procedures in this section tell you how to install the Administration and Operations applications and, if your workstation does not use a domain name server (DNS), how to update the workstation Hosts file.

Steps for installing the Administration and Operations applications

In these steps, you download and install the Administration and Operations applications package to your Windows workstation. The program package is a self-extracting exe file.

Before you begin: Check the workstation requirements in "Determining WebSphere for z/OS system requirements" on page 10.

Perform the following steps to install the Administration and Operations applications:

1. Open a command prompt and change directories to a directory into which you will download the program package.

   Example:
   ```
   C:\>cd temp

   C:\TEMP>
   ```

   _____

2. Issue the ftp command to the system on which WebSphere for z/OS runs. Log onto the system. You can log on with any user ID with an OMVS segment defined. Our example uses CBGUEST, but we suggest you use your own user ID.

   Example:
   ```
   C:\TEMP>ftp boss.my.com
   Connected to boss.my.com.
   220-FTPD1 IBM FTP CS V2R8 at OS390CBSERIES, 15:18:44 on 2000-04-18.
   220 Connection will close if idle for more than 5 minutes.
   User (boss.my.com:(none)): cbguest
   331 Send password please.
   Password:
   230 CBGUEST is logged on.  Working directory is "CBGUEST.".
   ```

   _____

3. Change directories to the directory where the program package resides (default is /usr/lpp/WebSphere/bin).

   Example:
   ```
   ftp> cd /usr/lpp/WebSphere/bin
   250 HFS directory /usr/lpp/WebSphere/bin is the current working directory
   ```

   _____

4. Issue the bin command and get the program package.

   Example:

   ```
   ftp> bin
   200 Representation type is Image
   ftp> get bboninst.exe
   200 Port request OK.
   125 Sending data set /usr/lpp/WebSphere/bin/bboninst.exe
   250 Transfer completed successfully.
   16725648 bytes received in 35.16 seconds (475.70 Kbytes/sec)
   ```

   _____

5. Quit ftp.

   Example:

   ```
   ftp> quit
   221 Quit command received. Goodbye.
   ```

   _____

6. From the Start menu, click Run, then use Browse to find the program package. Click OK.

   _____

7. Follow the InstallShield wizard to complete the installation.

   _____

You know you are done when the InstallShield wizard completes successfully.

Steps for updating the workstation Hosts file

If the workstation on which the Administration and Operations applications run is not connected to a Domain Name Server (DNS) or is not in the same domain as WebSphere for z/OS, you must update the workstation Hosts file. Through the Hosts file, your workstation can find the system on which WebSphere for z/OS runs. If your workstation is connected to a DNS, you can skip this procedure.

Before you begin: You must be on a running Windows system.

Perform the following steps to update the Hosts file on Windows:

1. Find the Hosts file. On Windows NT, it is usually in c:\winnt\system32\drivers\etc. On Windows 95, it is usually in c:\windows.

   Tip: If you do not have a Hosts file, you can create one using any text editor and placing it in the appropriate directory. You may have a sample Hosts file, Lmhosts.sam, that you can use to model your new Hosts file.

   _____

2. Make an association between a TCP/IP host name and an address by adding an entry to the file. Each entry in the Hosts file consists of an IP address, followed by a fully-qualified IP name and, optionally, one or more aliases. The fully-qualified name should be first after the IP address to assure proper address resolution. Each entry must be surrounded by blanks and on a single line.

   Example:
   ```
   #
   #  The following entries allow the workstation to access CB on OS390 without
   #  the workstation being in the same domain.
   #
   9.82.93.2 wsccb.washington.ibm.com  wsccb #CB Daemon_IPname and alias
   #
   #  The CB Resolve_IPname is the same for this installation or it, too, must
   #  be added.
   #
   ```

   _____

3. Save your Hosts file and test it. You can test your changes by opening a command window and issuing the ping command with the name you just added.

   Example:
   ```
   ping wsccb
   ```

   _____

You know you are done when you get a response from the ping command.

Defining application servers for the installation verification programs

Use the Administration application to define the BBOASR2 server, the BBOASR1 server, or both. The BBOASR2 server is a J2EE server. Three of the installation verification programs (IVPs) use BBPASR2 to test J2EE component support. The BBOASR1 server is a MOFW server. One IVP uses BBOASR1 to test MOFW component support. Besides allowing you to run the IVPs, these servers provide examples of how to set up your own business application servers.

Each of the following two main sections of this topic describe how to start the Administration application and a add new conversation. A *conversation* is a system management object that allows you to display and modify a WebSphere for z/OS configuration. (For information about conversations and the Administration application, see *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838.) Then each section describes how to define an application server, install the IVP application, then activate the conversation. Activating a conversation means that your server configuration has been updated for use by the system management function in WebSphere for z/OS.

You may define either server or both, depending on the component types you plan to use. Base your choice on which component type server you want:

If you want to set up:	Then define the server according to instructions in:
J2EE servers	"Defining the BBOASR2 J2EE server" on page 115
MOFW servers	"Defining the BBOASR1 MOFW server" on page 149
J2EE servers and MOFW servers	"Defining the BBOASR2 J2EE server" on page 115 followed by "Defining the BBOASR1 MOFW server" on page 149 (use two distinct conversations)

When a new conversation is created, all objects in the currently-active conversation are automatically added to the new conversation. Therefore, creating a J2EE server conversation first, followed by a MOFW conversation, results in the second conversation containing definitions for both J2EE and MOFW servers.

You can now perform the steps for the decision you have made.

The Administration application interacts with the System Management Server to do its work. You may find these interactions take some time to complete.

Defining the BBOASR2 J2EE server

If you plan to use J2EE components, do the steps in this section to set up BBOASR2, the J2EE server that the IVPs use to test J2EE component support.

Steps for starting the Administration application

Before you begin: You must initialize the WebSphere for z/OS run-time server instances and have the Administration application installed.

Perform these steps to start the Administration application:

1. On your workstation, click Start, then Programs, then IBM WebSphere for z/OS Administration.

   _____

2. Fill in the dialog with the Bootstrap server IP name, port 900, the user ID cbadmin, and password (for the password, see our RACF sample BBOCBRAK). Click OK.

   Recommendations:

   a. We strongly recommend that you **not** use the same administrator ID to log on to multiple concurrent sessions of the application, from either a single workstation or from more than one workstation. For example, if you start the Administration application on your workstation using CBADMIN as the user ID, you should not start another session using CBADMIN from either your own or a different workstation.

   b. If you define several administrator user IDs, they all may be logged on simultaneously, but only **one** should update and activate a conversation at a time.

      If more than one administrator attempts to activate a conversation, unexpected results will occur. When an administrator starts a new conversation, a copy of the currently active conversation is used as the base level. If more than one administrator creates a new conversation based on the same currently active conversation, the first administrator to activate will be successful. All others who try to activate will fail, since their changes are not based on the currently active conversation (the currently active conversation has changed out from under them). The second and subsequent administrators will have to start over again using the new current conversation. Depending on the amount of change, this can be very disruptive. Thus, while one administrator is updating and activating a conversation, the others should use the administration application only for read or display functions.

_____

You know you are done when the main window appears showing the
bootstrap conversation. If you have trouble connecting, check the Help system
or *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management
User Interface*, SA22-7838, for more information.

Steps for starting a new conversation

Before you begin: You must start the Administration application by logging in.

Perform these steps to start a new conversation:

1. Select the Conversations folder with the left mouse button. Then, using the right mouse button, click the Conversations folder, then select Add.

   _____

2. In the properties form (right panel), name your new conversation. For example, we named the conversation "BBOASR2 SERVER DEFINITION." Add a description (optional).

   _____

3. Click the save (diskette) icon. The words "Adding... Conversations" appear in the tree.

   _____

You know you are done when the following message appears in the status bar:

```
BBON0515I Conversation BBOASR2 SERVER DEFINITION was added.
```

The screen looks like this:

Steps for adding the BBOASR2 J2EE server

Before you begin: You must be working on the current conversation.

Perform these steps to add the new server:

1. Expand your new conversation tree by clicking the node to the left of the conversation name.

   _____

2. Expand Sysplexes, then your sysplex.

   _____

3. Select the J2EE server folder with the left mouse button. Then, using the right mouse button, Click the J2EE server folder, then select Add.

   _____

4. In the properties form, enter values or make selections as appropriate for your installation.

Server name	BBOASR2
Server description	Optional server description
Control region identity	The user ID under which the control region runs. This must match an entry in the RACF STARTED class and have appropriate RACF authorizations for a control region. The default value in BBOCBRAK is CBACRU2.
Server region identity	The user ID under which the server region runs. This must match an entry in the RACF STARTED class and have appropriate RACF authorizations for a server region. The default value in BBOCBRAK is CBASRU2.
Server region stack size (in bytes)	0
Production J2EE server	Select the check box
Debugger allowed	Leave unchecked
Object Level Trace hostname	Leave blank
Object Level Trace port	Leave the default
Isolation policy	One transaction per server region
Replication policy	One per server
Local identity	The user ID you chose as the local unauthenticated user ID on the IVP customization panel in the customization dialog. By default, the customization dialog uses CBIVP2.
Remote identity	The user ID you chose as the remote unauthenticated user ID on the IVP customization panel in the customization dialog. By default, the customization dialog uses CBIVP2.

Register transaction factory	Clear the check box[*]

* A server that registers as a transaction factory must be available at all times. Because BBOASR2 is available only during installation verification, this server should not register as a transaction factory.

The Naming Server is defined as a transaction factory. If you remove the Naming Server from the configuration, you need to make another server into a transaction factory. You can have more than one transaction factory, but remember that such servers must be available at all times.

Allow server region recycling	Select the check box
Server recycling interval	50000
Logstream name	The name of the log stream you set up for capturing error information. See Table 18 on page 77. You may leave this blank, in which case the system uses the Daemon's log stream.
Control region proc name	BBOASR2 (default)
Enable Setting OS Thread Identity to RunAs	Select the check box
Allow non-authenticated clients	Select the check box
Userid password allowed	Select the check box
Userid passticket allowed	Clear the check box
DCE allowed	Clear the check box
DCE quality of protection	No protection
DCE keytab file	Leave blank
SSL Type 1 allowed	Clear the check box
SSL Client Certificates allowed	Clear the check box
Kerberos allowed	Clear the check box
Send Asserted Identities allowed	Clear the check box
Accept Asserted Idenitites allowed	Clear the check box
SSL Use Confidentiality only	Clear the check box
SSL RACF keyring	CBKeyring
SSL V2 timeout	100
SSL V3 timeout	600
Security preference list	Set Password to priority 1

Write Server Activity Records	If you want to gather server activity records, select the check box.
Write Container Activity SMF Records	If you want to gather container activity records, select the check box.
Write Server Interval SMF Records	If you want to gather server interval records, select the check box.
Write Container Interval SMF Records	If you want to gather container interval records, select the check box.
Interval length	0
Environment variable list	Check environment variables. See step 5.

_____

5. Set the environment variables for BBOASR2 as follows. For more information about how to add or modify environment variables, use the help system in the Administration application or see *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838.

The following are environment variables to set:

LIBPATH
   The LIBPATH variable specifies the DLL search paths for Java and JDBC in the hierarchical file system (HFS). Specify system, WebSphere for z/OS, Java, and JDBC DLLs.

   Example:
   ```
   LIBPATH=/db2_install_path/lib
   :/usr/lpp/java/J1.3/bin
   :/usr/lpp/java/J1.3/bin/classic
   :/usr/lpp/WebSphere/lib
   ```

   where *db2_install_path* is the HFS where you installed DB2.

CLASSPATH
   The CLASSPATH statement specifies Java class files for use by Java applications in server regions. Ensure the CLASSPATH has the following:
   ```
   CLASSPATH=db2_install_path/classes/db2j2classes.zip
   ```

   where *db2_install_path* is the HFS where you installed DB2.

   Note: After activation of this conversation, System Management automatically prepends ws390srt.jar, waswebc.jar, and xerces.jar to the application server CLASSPATH for you.

   For the Web client IVP, add the following to the CLASSPATH:

```
/usr/lpp/WebSphere/lib/waswebccp.jar
```

JVM_LOGFILE
Set it to a file in which you want the log.

Example:
```
/serverdir/jvm.log
```

where *serverdir* is a directory to which the BBOASR2 control and
server region identities have write access.

TRACEALL
Make sure the TRACEALL environment variable is set to 1, to improve
the performance of the server.

DB2SQLJPROPERTIES
Set it to point to the properties file for JDBC. For more information
about this environment variable, see *DB2 for OS/390 Application
Programming Guide and Reference for Java.*

_____

6. Click the save (diskette) icon. The words "Adding... J2EE servers" appear
in the tree.

_____

You know you are done when the following appears in the status bar:
```
BBON0515I J2EEServer BBOASR2 was added.
```

The screen looks like this:

WebSphere Application Server for z/OS and OS/390 Administration [_][□][X]

File Selected Build View Options Help

○—□ Conversations
  ○ BBOASR2 SERVER DEFINITION
    ○—□ Sysplexes
      ○ PLEX1
        ○—□ J2EEServers
          ○— BBOASR2
        ○—□ Servers
        ○—□ Systems
        ○—□ J2EE Resources
        ○—□ Logical Resource Mappings
  ○—⊠ Bootstrap Conversation

J2EEServers ▼

BBON0515I J2EEServer BBOASR2 was added.

BBOASR2 SERVER DEFINITION: Modifiable   CBADMIN
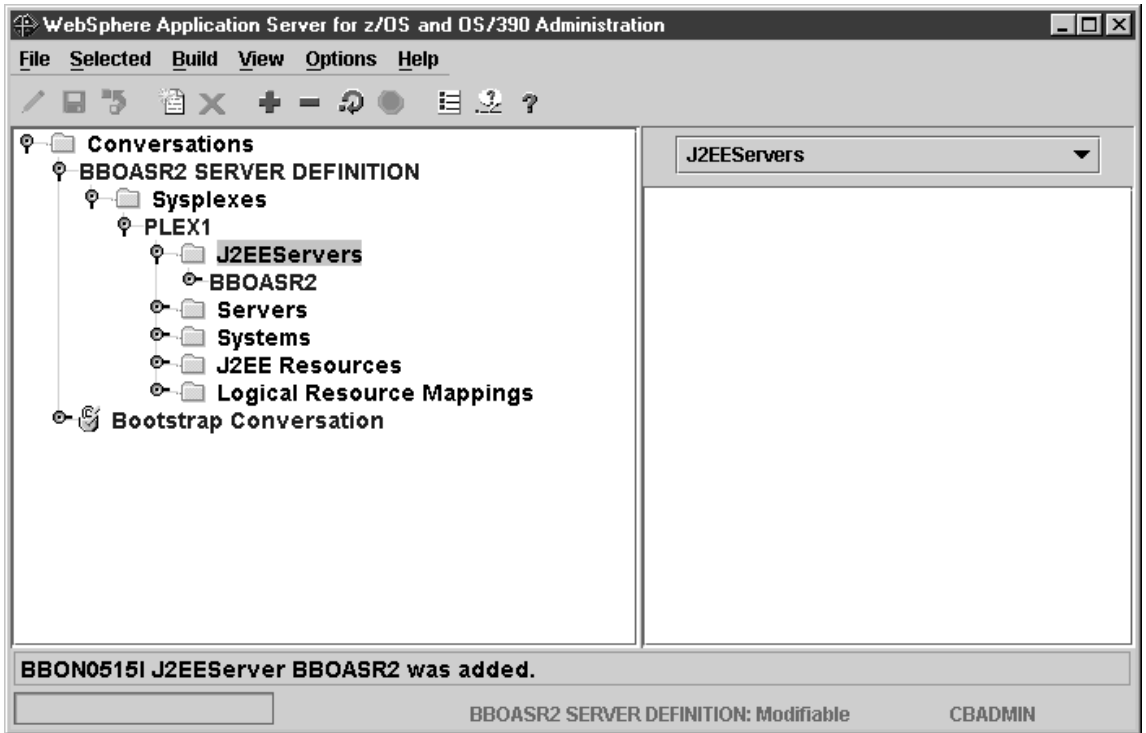
Steps for adding the BBOASR2A server instance
Before you begin: You must have the BBOASR2 server defined.

Perform these steps to add the server instance:

1. If necessary, expand the tree by clicking the node to the left of J2EEServers and BBOASR2.

   _____

2. Select Server Instances with the left mouse button. Then, using the right mouse button, click Server Instances, then select Add.

   _____

3. In the properties form, enter BBOASR2A as the server instance name.

   _____

4. Optional: enter a server instance description.

   _____

5. Optional: supply a log stream name. If you do not supply one, the default is the log stream name you chose for the BBOASR2 server.

   _____

6. Click the save (diskette) icon. The words "Adding... Server Instances" appear in the tree.

   _____

You know you are done when the following message appears in the status bar:

```
BBON0515I Server instance BBOASR2A was added.
```

The screen looks like this:
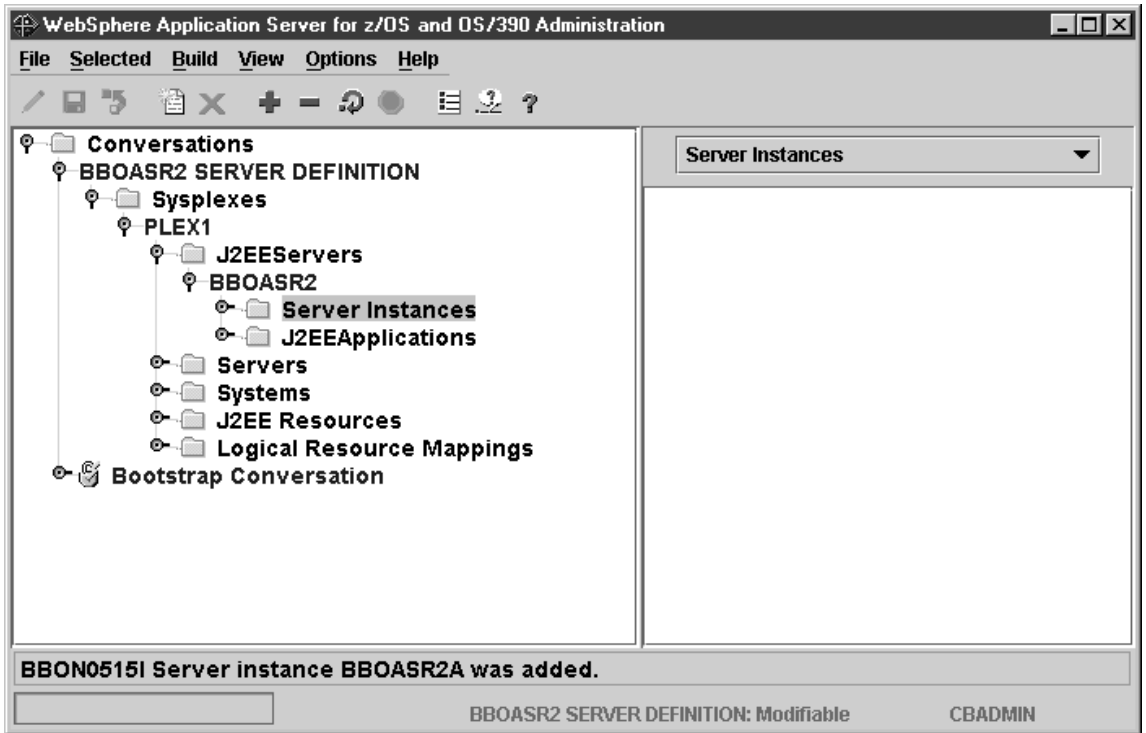
Steps for adding a J2EE resource
Before you begin: You must be working on the current conversation.

Perform these steps to add a J2EE resource:

1. Select J2EE resources with the left mouse button. Then, using the right
   mouse button, click J2EE resources, then select Add.

   _____

2. In the properties form, enter a name for the J2EE resource. For example,
   we used "BBOASR2_EJB_IVP_RESOURCE."

   _____

3. Optional: enter a description of the J2EE resource.

   _____

4. Find the property labelled J2EE resource type, and select DB2datasource.
   The Administration application fills in the fields above with the
   information that is appropriate for a DB2 data source.

   _____

5. Click the save (diskette) icon. The words "Adding... J2EE resources" appear
   in the tree.

   _____

You know you are done when the following message appears in the status
bar:

BBON0515I J2EE Resources *name* was added.

where *name* is the name you chose for the J2EE resource.

The screen looks like this:
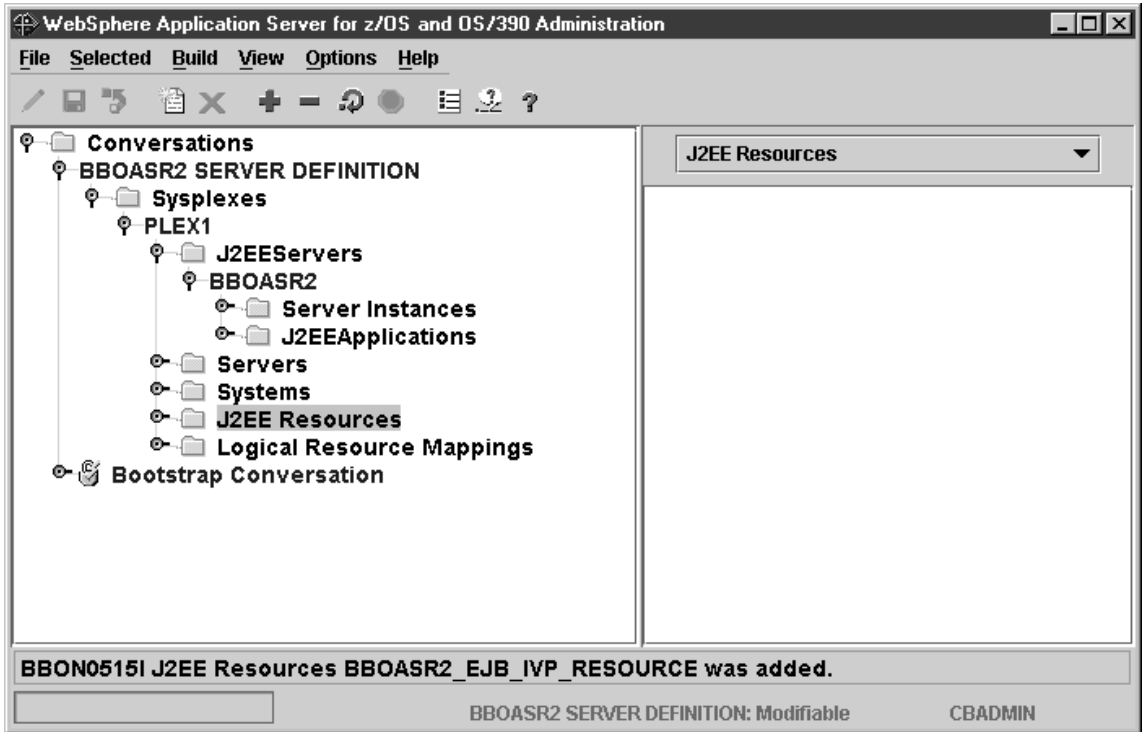
Steps for adding the J2EE resource instance
Before you begin: You must define a J2EE resource.

Perform these steps to add the J2EE resource instance:

1. If necessary, expand the tree for the newly created J2EE resource by clicking the node to the left of the J2EE resource name.

    _____

2. Select J2EE Resource Instances with the left mouse button. Then, using the right mouse button, Click J2EE Resource Instances, then select Add.

    _____

3. In the properties form, enter the appropriate values:
    *   J2EE resource instance name.
        Example: BBOASR2_EJB_IVP_RESOURCE_*system*, where *system* is your system name.
    *   J2EE resource instance description (optional).
    *   Location Name: supply the the DB2 location name.

    _____

4. Click the save (diskette) icon. The words "Adding... J2EE resource instances" appear in the tree.

    _____

You know you are done when the following message appears in the status bar:

```
BBON0515I J2EE resource Instance name was added.
```

where *name* is the name you chose for the J2EE resource instance.

The screen looks like this:
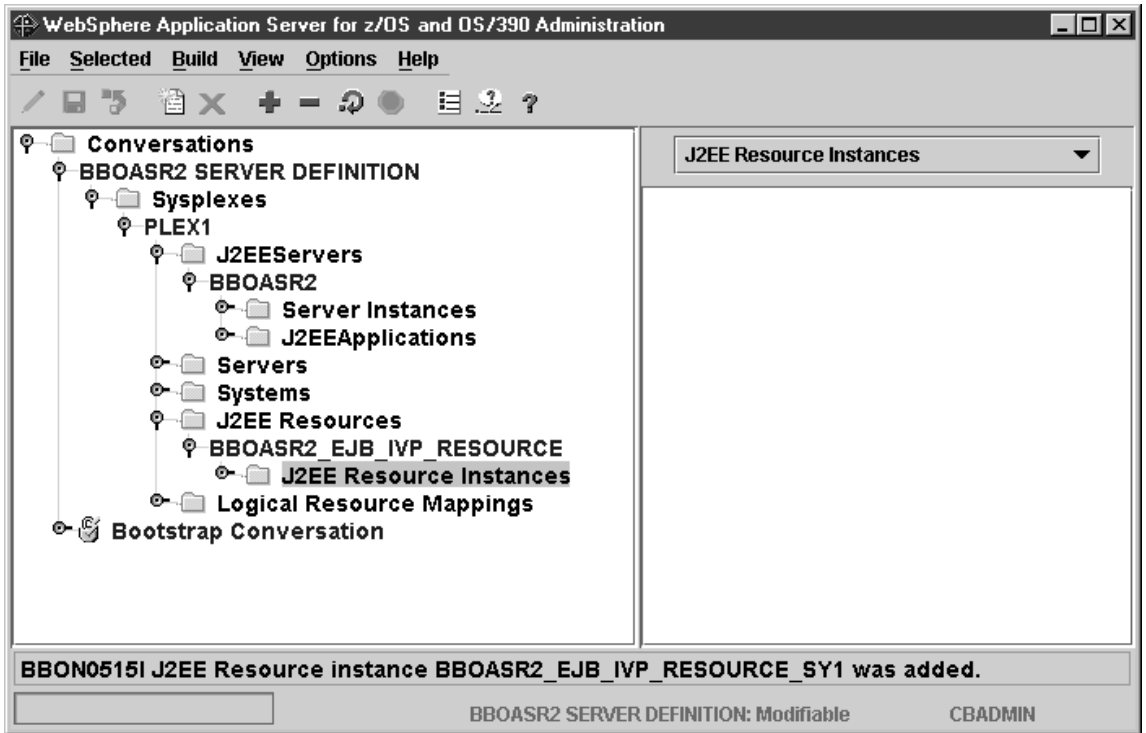
Steps for installing WebSphereSampleApp.ear into the Web container
Before you begin:

- Make sure that the FTP server on z/OS or OS/390 is running.
- Download in binary the WebSphereSampleApp.ear file from the WebSphere for z/OS samples directory. The default location for this file is:

  `/usr/lpp/WebSphere/samples`

Perform the following steps to installing WebSphereSampleApp.ear into the Web container:

1. In the tree, right-click the BBOASR2 server.

   _____

2. Choose Install J2EE Application... from the Selected menu bar. The Install J2EE Application dialog box appears.

   _____

3. In the dialog box, enter the following values:
   - The name of the EAR file that contains your J2EE application. Use the Browse button to navigate to the WebSphereSampleApp.ear file in your workstation file system.
   - The name of the FTP server for the sysplex in which you want to install your application. Usually, this is the IP name of the system you logged onto (it is displayed as the default).

   Example:

   Click OK.

   Result: A pop-up appears with the words "Loading ear file," then the Reference and Resource Resolution window appears and displays the application content in the ear file.

   _____

4. In the Reference and Resource Resolution window:
   a. Expand default_app_WebApp.jar by clicking the node to the left of the folder.

b. Click the default_app_WebApp bean, then click the button labelled "Set Default JNDI Path & Name."

   Result: You will know you have finished this process when the bean symbol to the left of the bean name has a checkmark over it.

5. Repeat these steps for examples_WebApp and the examples_WebApp bean.

   Result:

_____
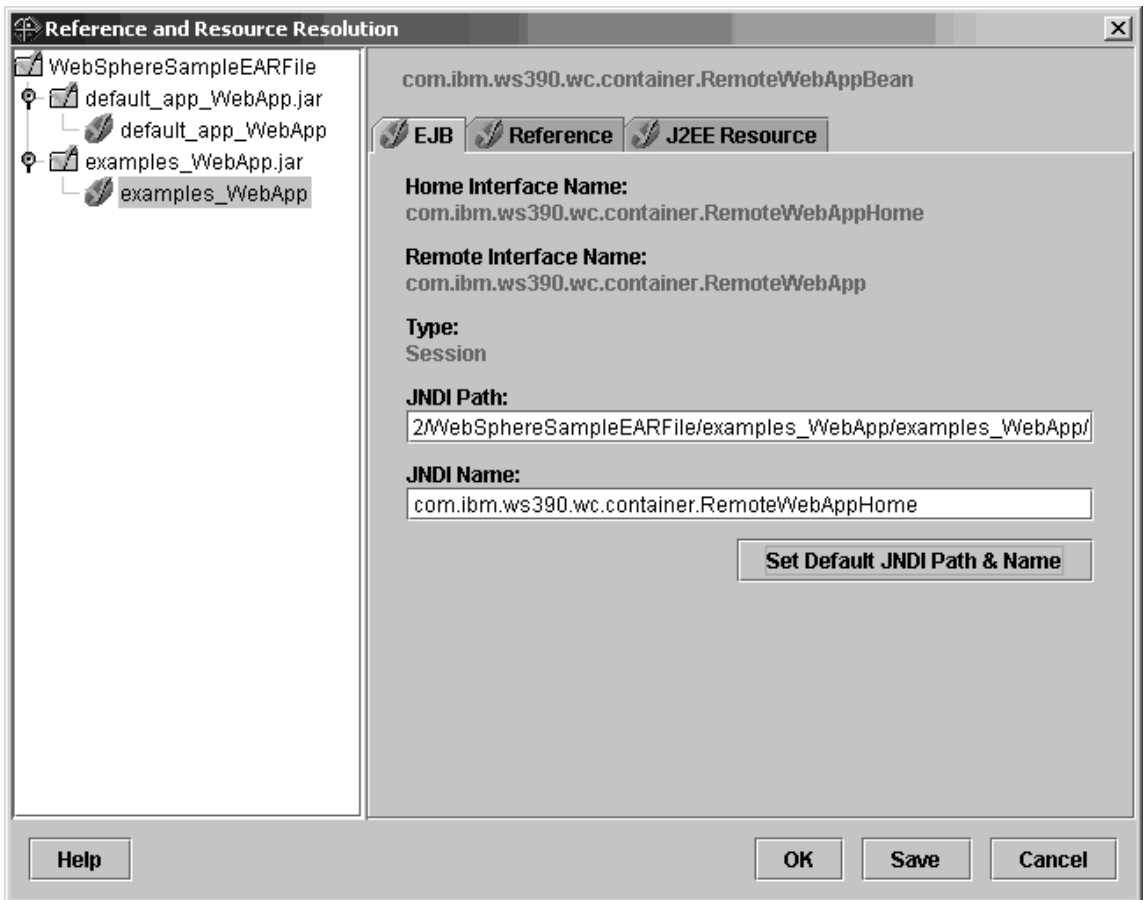
6. Click OK.

   Result: This action starts the automatic FTP transfer of your EAR file contents from your workstation to z/OS or OS/390. A pop-up appears with messages describing the stage in the FTP transfer.

   Example:

Processing Install EJBs ✕

Connecting to server...

[ ]

Cancel

Then the words Deploying... BBOASR2 appear in the tree.

The FTP transfer follows these stages:

Stage	Description
1	When the ear file is imported, the system FTPs it to
	targetdir/*sysplex*/temp/*administrator_ID*/WebSphereSampleApp_resolved.ear
	targetdir is the mount point, *sysplex* is the name of the sysplex, and *administrator_ID* is the user ID of the administrator (usually CBADMIN).
2	The ear file is processed. During ear file processing, the ear file is exploded into directory
	targetdir/apps/BBOASR2/L*n*/*app_name*/
	app_name is the name of the application (not necessarily equal to the ear file name).
3	A scaffolding directory
	targetdir/apps/BBOASR2/L*n*/A/
	is created under which all the deployment information is stored.

Note: Upon activation of the conversation, everything beneath
targetdir/apps/BBOASR2/L*n*/

is moved one level up to
targetdir/apps/BBOASR2/

_____

You know you are done when the following message appears in the status
bar:

```
BBON0470I EAR file WebSphereSampleApp_resolved.ear has been successfully installed on server BBOASR2.
```

Steps for installing PolicyIVP.ear in the EJB container
Before you begin:

- Make sure that the FTP server on z/OS or OS/390 is running.
- Download in binary the PolicyIVP.ear file from the WebSphere for z/OS samples directory to your workstation. The default location for the file on z/OS or OS/390 is:

  `/usr/lpp/WebSphere/samples/PolicyIVP/ejb`

Perform the following steps to install the EAR file for your application, using the WebSphere for z/OS Administration application:

1. In the tree, select the BBOASR2 server.

   _____

2. Choose Install J2EE Application... from the Selected menu bar. The Install J2EE Application dialog box appears.

   _____

3. In the dialog box, enter the following values:
   - The name of the EAR file that contains your J2EE application. Use the Browse button to navigate to the PolicyIVP.ear file in your workstation file system.
   - The name of the FTP server for the sysplex in which you want to install your application. Usually, this is the IP name of the system you logged onto (it is displayed as the default).
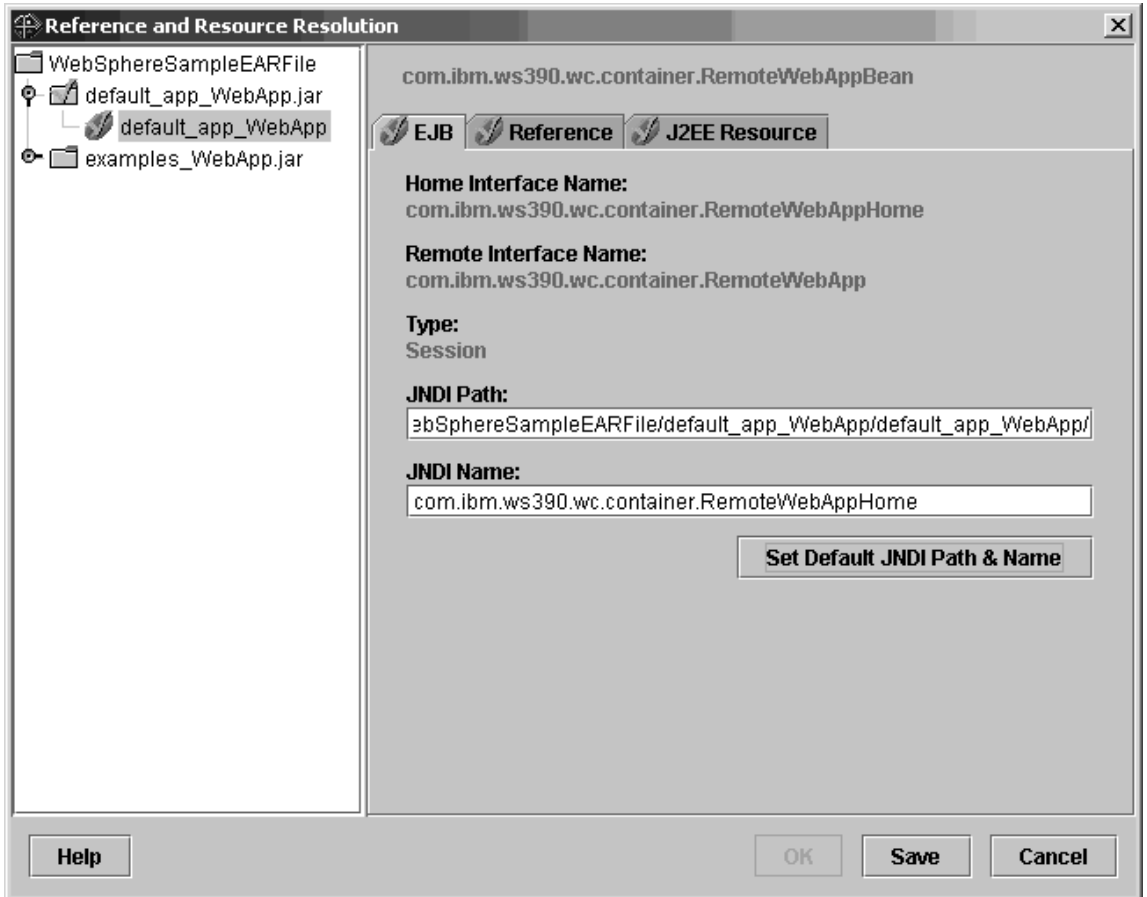
   Example:

   Click OK.

   Result: A pop-up appears with the words "Loading ear file," then the Reference and Resource Resolution window appears and displays the application content in the ear file.

   _____

4. Expand each folder listed in the Reference and Resource Resolution window by clicking the node to the left of the folder. Set the JNDI Path

and JNDI Name for each bean in turn by clicking the bean, then clicking the button labelled "Set Default JNDI Path & Name."

_____

5. Select each bean by clicking the bean symbol.
   a. If the "J2EE Resource" tab has a green check mark, select the next bean.
   b. If the "J2EE Resource" does not have a green check mark:
      - Click "J2EE Resource" tab.
      - Click on the blank space in the table in the "J2EE Resource" column, which brings up a list of J2EE resources.
      - Click on the name of the J2EE resource you created earlier in "Steps for adding a J2EE resource" on page 126.

   Example:
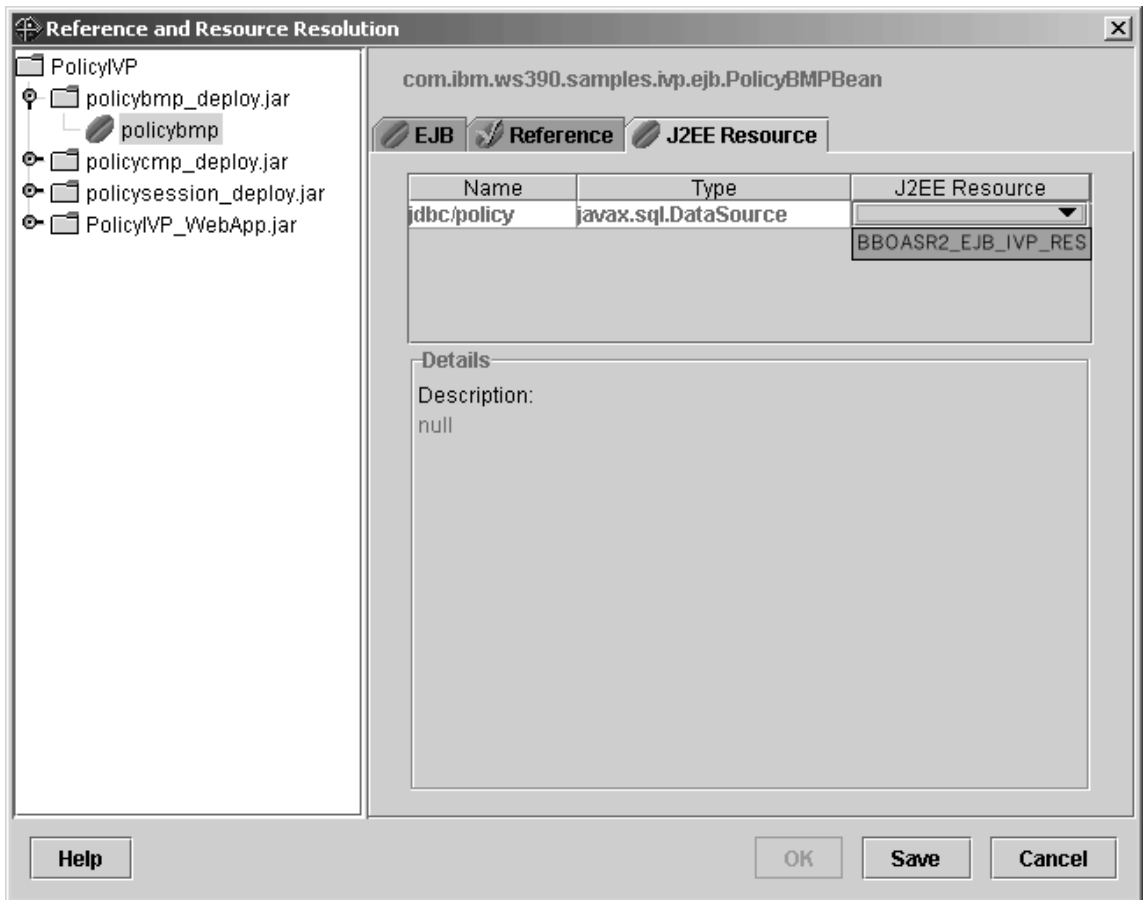
Result: A check-mark appears on the bean symbol.

6. When all beans have checkmarks to the left, click OK.
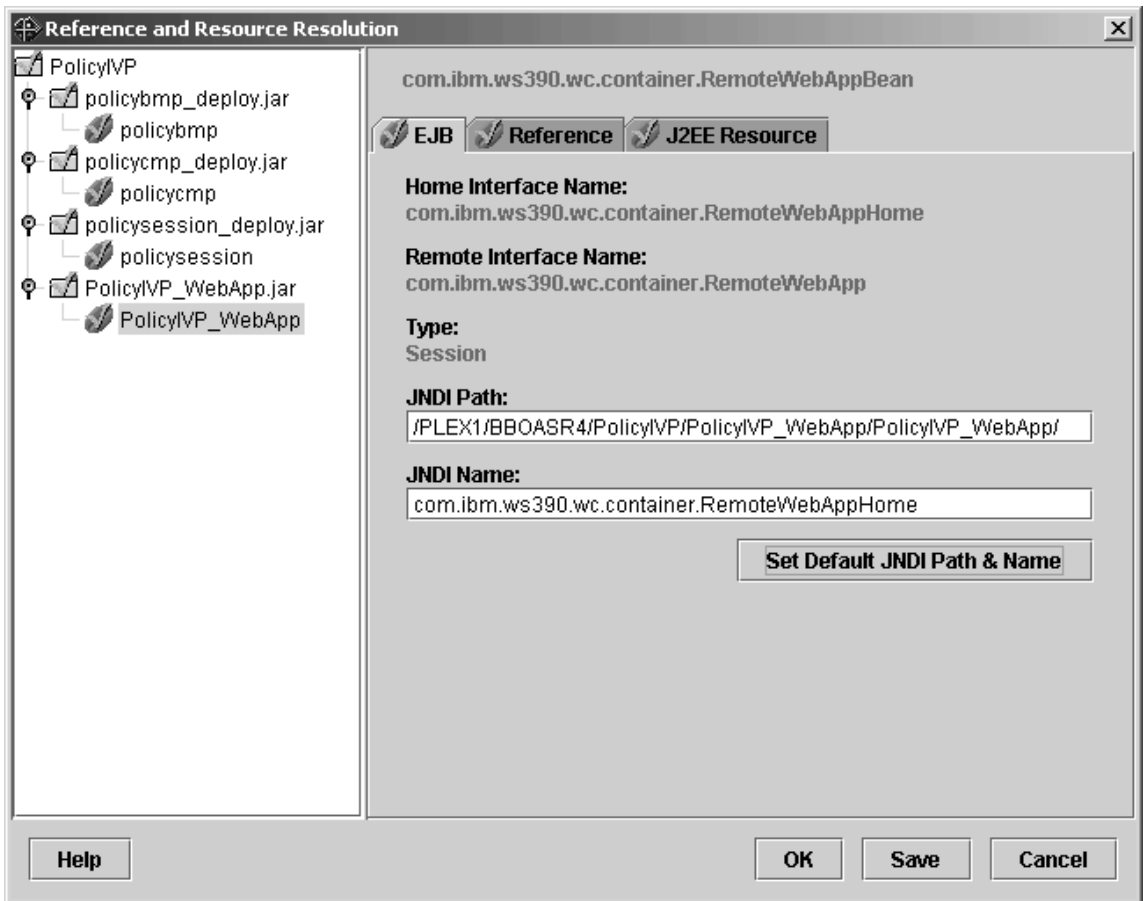
Result: This action starts the automatic FTP transfer of your EAR file contents from your workstation to z/OS or OS/390. A pop-up appears with messages describing the stage in the FTP transfer.

Processing Install EJBs ✕

Connecting to server...

Cancel

Then the words `Deploying... BBOASR2` appear in the tree.

Tip: Data from the Reference and Resource Resolution window is saved in a new copy of the ear file named *application_name_*resolved.ear before it is transferred to the server for deployment. If you reopen that copy of the file later, you do not have to re-enter the information a second time.

The FTP transfer follows these stages:

Stage	Description
1	When the ear file is imported, the system FTPs it to
	targetdir/*sysplex*/temp/*administrator_ID*/PolicyIVP_resolved.ear
	targetdir is the mount point, *sysplex* is the name of the sysplex, and *administrator_ID* is the user ID of the administrator (usually CBADMIN).
2	The ear file is processed. During ear file processing, the ear file is exploded into directory
	targetdir/apps/BBOASR2/L*n*/*app_name*/
	app_name is the name of the application (not necessarily equal to the ear file name).
3	A scaffolding directory
	targetdir/apps/BBOASR2/L*n*/A/
	is created under which all the deployment information is stored.

Stage	Description
	Note: Upon activation of the conversation, everything beneath *targetdir*/apps/BBOASR2/L*n*/ is moved one level up to *targetdir*/apps/BBOASR2/

You know you are done when the following message appears in the status bar:

```
BBON0470I EAR file PolicyIVP_resolved.ear has been successfully installed on server BBOASR2.
```

Here is what the screen looks like when you have successfully installed the IVP:
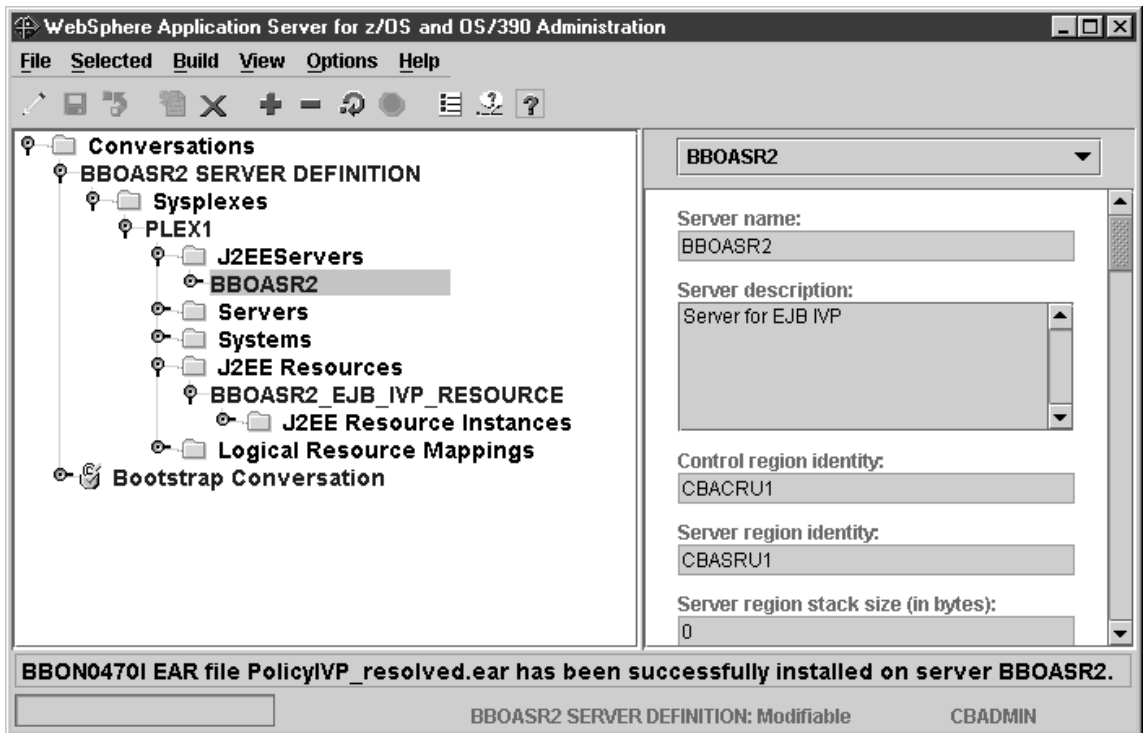
Steps for validating the conversation

Before you begin: You must complete all the previous steps in the current conversation.

Perform the following steps to validate the conversation:

1. If necessary, scroll up the tree to the BBOASR2 SERVER DEFINITION conversation name.

   _____

2. Select the conversation with the left mouse button. Then, using the right mouse button, click the conversation, then select Validate.

   _____

You know you are done when the following message appears in the status bar:

```
BBON0442I Conversation BBOASR2 SERVER DEFINITION is valid.
```

WebSphere Application Server for z/OS and OS/390 Administration

File   Selected   Build   View   Options   Help

- Conversations
- **BBOASR2 SERVER DEFINITION**
  - **Sysplexes**
    - **PLEX1**
      - **J2EEServers**
        - BBOASR2
      - **Servers**
      - **Systems**
      - **J2EE Resources**
        - BBOASR2_EJB_IVP_RESOURCE
          - **J2EE Resource Instances**
      - **Logical Resource Mappings**
  - **Bootstrap Conversation**

BBOASR2 SERVER DEFINITION ▼

Conversation name:
BBOASR2 SERVER DEFINITION

Conversation description:

BBON0442I Conversation BBOASR2 SERVER DEFINITION is valid.

BBOASR2 SERVER DEFINITION: Modifiable   CBADMIN

Step for committing the conversation

Before you begin: You must validate the current conversation.

⇔ Select the conversation with the left mouse button. Then, using the right mouse button, click the conversation, then select Commit. Answer Yes to the question:

```
BBON0534I You cannot undo Commit.  Do you still want to commit?
```
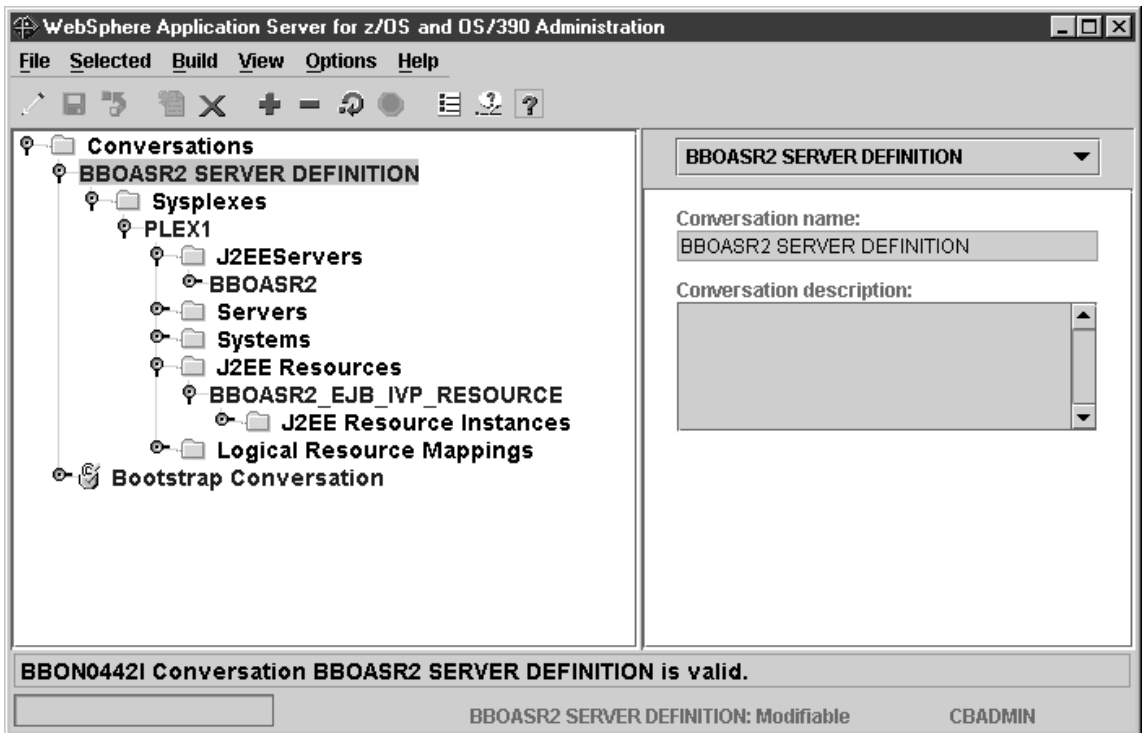
The words "Committing... BBOASR2 SERVER DEFINITION" appear in the tree.

You know you are done when the following message appears in the status bar:

```
BBON0444I Conversation BBOASR2 SERVER DEFINITION was committed.
```

The screen looks like this:
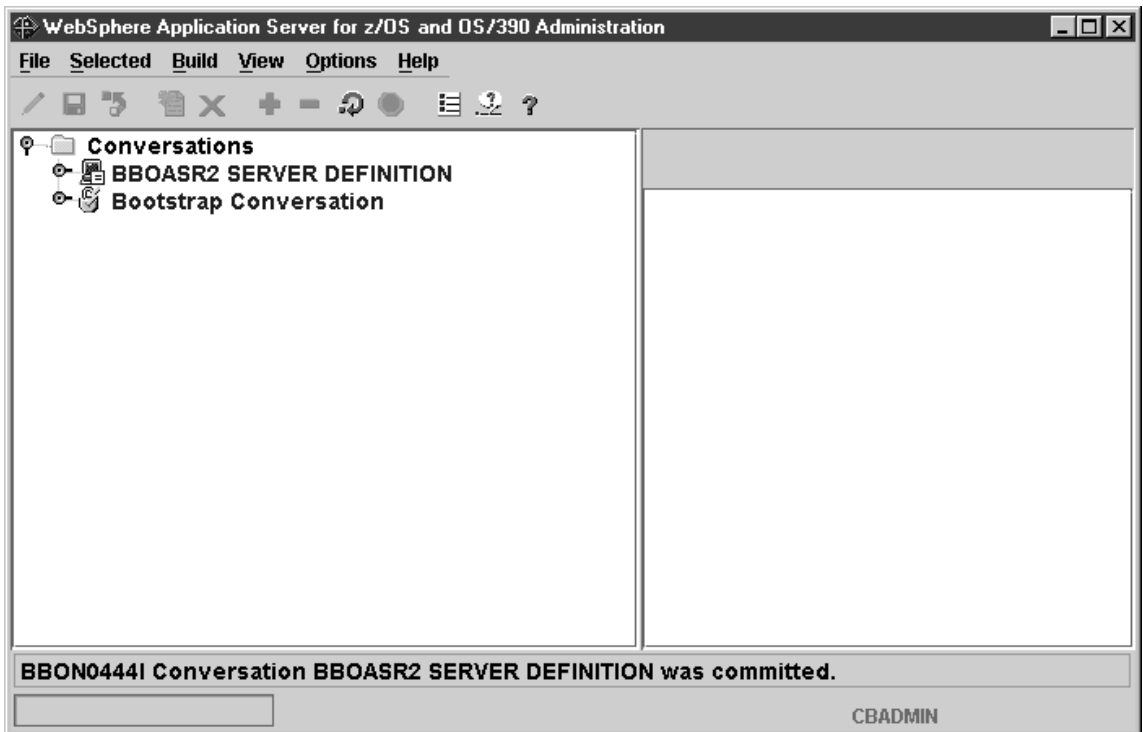
Steps for following the instructions for completing z/OS or OS/390 tasks
Before you begin: You must validate and commit the current conversation.

Perform these steps to follow the instructions for completing z/OS or OS/390 tasks:

1. Select the BBOASR2 SERVER DEFINITION conversation with the left mouse button. Then, using the right mouse button, click the conversation, then select Instructions.

   _____

2. Complete all instructions provided by the Administration application for completing z/OS or OS/390 tasks.

   _____

You know you are done when you have completed all the required z/OS or OS/390 tasks.

The screen looks like this:
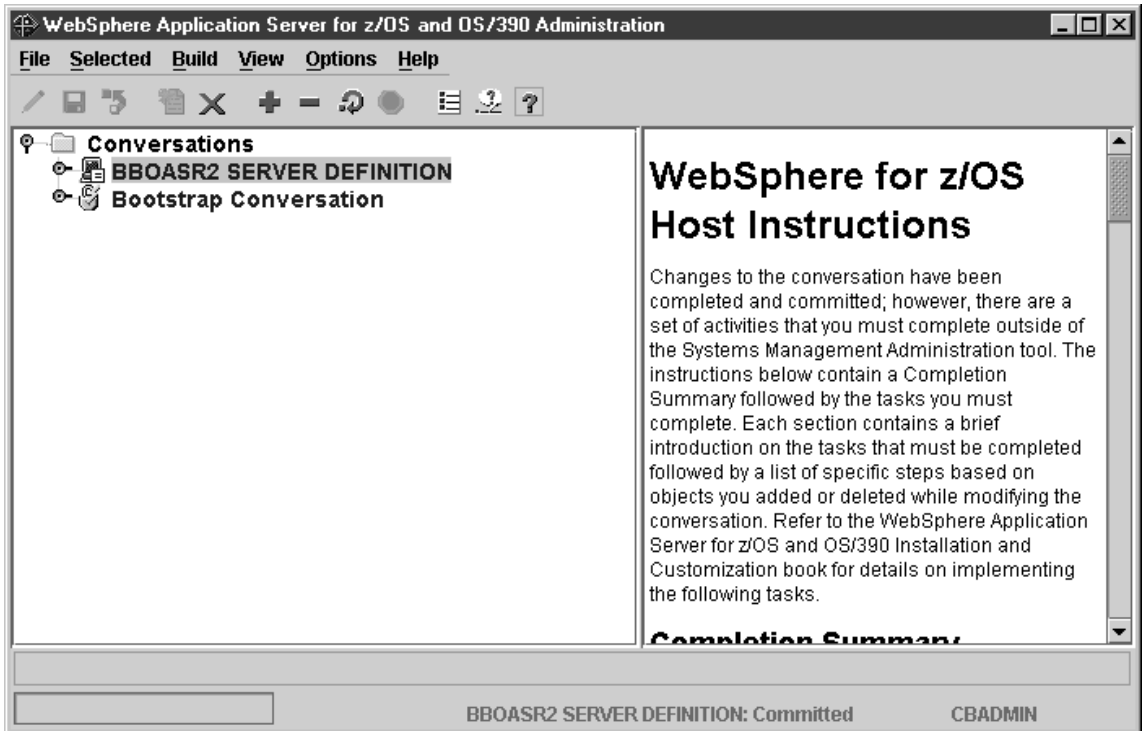
Steps for marking all tasks complete
Before you begin: You must complete all required z/OS or OS/390 tasks.

Perform these steps to mark all tasks complete:

1. Select the BBOASR2 SERVER DEFINITION conversation with the left mouse button. Then, with the right mouse button, click the conversation, select Complete, then All tasks.

   _____

2. Answer Yes to the question:

   ```
   BBON0550I  Are you sure that all tasks have been completed?
   ```

   _____

You know you are done when the following message appears in the status bar:

```
BBON0484I All tasks complete.
```

The screen looks like this:

Steps for activating your new conversation
Before you begin: You must complete all previous instructions in this section.

Perform these steps to activate your new conversation:

1. Select the BBOASR2 SERVER DEFINITION conversation with the left
   mouse button. Then, with the right mouse button, click the conversation,
   then select Activate.

   _____

2. Answer Yes to the question:

   ```
   BBON0539I  Activate cannot be undone.  Do you want to activate conversation
              BBOASR2 SERVER DEFINITION?
   ```

   Result: The words "Activating... BBOASR2 SERVER DEFINITION" appear
   in the tree.

   _____

You know you are done when the following message appears in the status
bar:

```
BBON0449I Conversation BBOASR2 SERVER DEFINITION was activated.
```

The screen looks like this:

Steps for printing the Administration Message Log

Before you begin: You must activate your conversation.

Follow these steps to print the Administration Message Log:

1. Click File, then Message log...

   Result: The screen looks like this:

---

2. From the Administration Message Log window, click File, then Print...

Result: You see the Windows print dialog. Select a printer and click ok. You see the folowing pop-up:
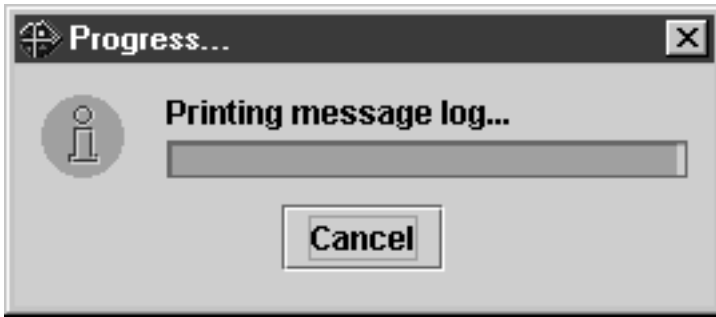
You know you are done when you get a printout of the Administration Message Log. You may exit the program.

You have finished defining the BBOASR2 server

If you want to run the MOFW IVP, continue with "Defining the BBOASR1 MOFW server" on page 149. Otherwise, go to "Steps for creating the database for the installation verification programs (IVPs)" on page 191.

Defining the BBOASR1 MOFW server

If you plan to use MOFW components, do the steps in this section to set up BBOASR1, the MOFW server that the IVP uses to test MOFW component support.

Steps for starting the Administration application

Before you begin: You must initialize the WebSphere for z/OS run-time server instances and have the Administration application installed.

Perform these steps to start the Administration application:

1. On your workstation, click Start, then Programs, then IBM WebSphere for z/OS Administration.

   _____

2. Fill in the dialog with the Bootstrap server IP name, port 900, the user ID cbadmin, and password (for the password, see our RACF sample BBOCBRAK). Click OK.

   Recommendations:

   a. We strongly recommend that you **not** use the same administrator ID to log on to multiple concurrent sessions of the application, from either a single workstation or from more than one workstation. For example, if you start the Administration application on your workstation using CBADMIN as the user ID, you should not start another session using CBADMIN from either your own or a different workstation.

   b. If you define several administrator user IDs, they all may be logged on simultaneously, but only **one** should update and activate a conversation at a time.

      If more than one administrator attempts to activate a conversation, unexpected results will occur. When an administrator starts a new conversation, a copy of the currently active conversation is used as the base level. If more than one administrator creates a new conversation based on the same currently active conversation, the first administrator to activate will be successful. All others who try to activate will fail, since their changes are not based on the currently active conversation (the currently active conversation has changed out from under them). The second and subsequent administrators will have to start over again using the new current conversation. Depending on the amount of change, this can be very disruptive. Thus, while one administrator is updating and activating a conversation, the others should use the administration application only for read or display functions.

_____

You know you are done when the main window appears showing the
bootstrap conversation. If you have trouble connecting, check the Help system
or *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management
User Interface*, SA22-7838, for more information.

Steps for starting a new conversation

Before you begin: You must start the Administration application by logging in.

Perform these steps to start a new conversation:

1. Select the Conversations folder with the left mouse button. Then, using the right mouse button, click the Conversations folder, then select Add.

   _____

2. In the properties form (right panel), name your new conversation. For example, we named the conversation "BBOASR1 Server Definition." Add a description (optional).

   _____

3. Click the save (diskette) icon. The words "Adding... Conversations" appear in the tree.

   _____

You know you are done when the following message appears in the status bar:

```
BBON0515I Conversation BBOASR1 Server Definition was added.
```
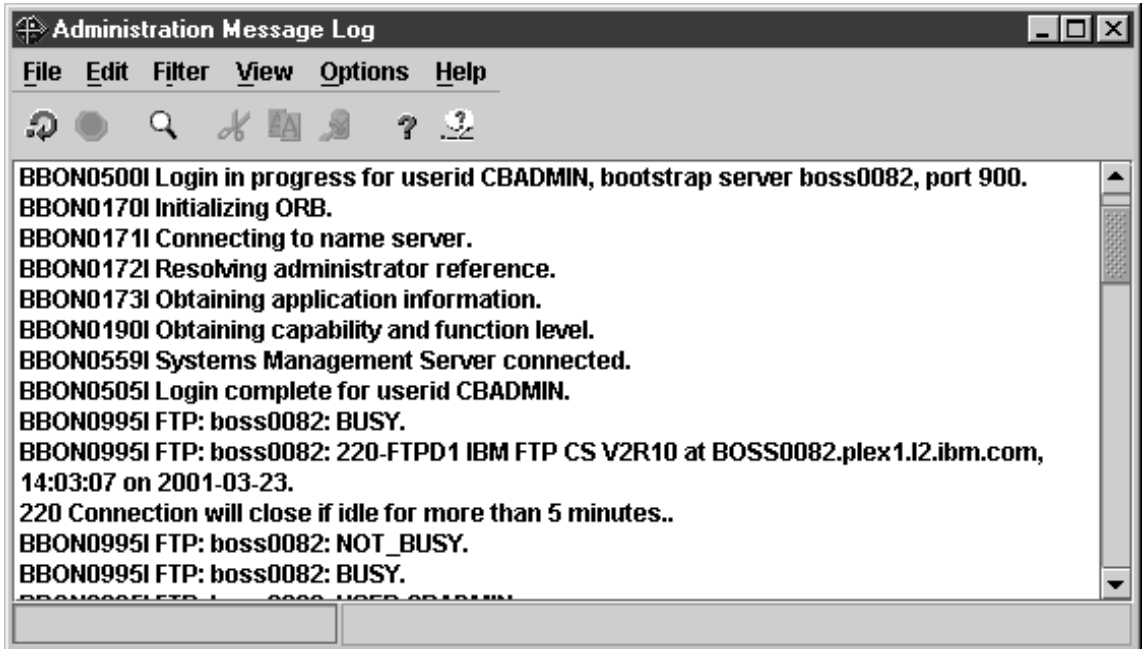
The screen looks like this:

Steps for adding the BBOASR1 MOFW server

Before you begin: You must be working on the current conversation.

Perform these steps to add the BBOASR1 server.

1. If necessary, expand your new conversation tree by clicking the node to the left of the conversation name.

   _____

2. Expand Sysplexes, then your sysplex.

   _____

3. Select the Servers folder with the left mouse. Then, using the right mouse button, click the Servers folder, then select Add.

   _____

4. In the properties form, enter values or make selections as follows.

Server name	BBOASR1
Server description	Optional server description
Control region identity	The user ID under which the control region runs. This must match an entry in the RACF STARTED class and have appropriate RACF authorizations for a control region. The default value in BBOCBRAK is CBACRU1.
Server region identity	The user ID under which the server region runs. This must match an entry in the RACF STARTED class and have appropriate RACF authorizations for a server region. The default value in BBOCBRAK is CBASRU1.
Server region stack size (in bytes)	0
Production server	Select the check box
Debugger allowed	Leave unchecked
Object Level Trace hostname	Leave blank
Object Level Trace port	Leave the default
Isolation policy	Multiple transactions per server region
Replication policy	One per server
Local identity	The user ID you chose as the local unauthenticated user ID on the IVP customization panel in the customization dialog. By default, the customization dialog uses CBIVP.
Remote identity	The user ID you chose as the remote unauthenticated user ID on the IVP customization panel in the customization dialog. By default, the customization dialog uses CBIVP.
Register transaction factory	Clear the check box[*]

* A server that registers as a transaction factory must be available at all times. Because BBOASR1 is available only during installation verification, this server should not register as a transaction factory.

The Naming Server is defined as a transaction factory. If you remove the Naming Server from the configuration, you need to make another server into a transaction factory. You can have more than one transaction factory, but remember that such servers must be available at all times.

Allow server region recycling	Select the check box
Server recycling collection interval	50000
Logstream name	The name of the log stream you set up for capturing error information. See Table 18 on page 77. You may leave this blank, in which case the system uses the Daemon's log stream.
Control region proc name	BBOASR1 (default)
Allow non-authenticated clients	Select the check box
Userid password allowed	Select the check box
Userid passticket allowed	Clear the check box
DCE allowed	Clear the check box
DCE quality of protection	No protection
DCE keytab file	Leave blank
SSL Type 1 (Basic Authentication) allowed	Clear the check box
SSL Client Certificates allowed	Clear the check box
Kerberos allowed	Clear the check box
Send Asserted Identities allowed	Clear the check box
Accept Asserted Identities allowed	Clear the check box
SSL Use Confidentiality Only	Clear the check box
SSL RACF Keyring	Leave blank
SSL V2 timeout	Leave blank
SSL V3 timeout	Leave blank
Security preference list	Set Password to priority 1

Write Server Activity SMF Records	If you want to gather server activity records, select the check box.
Write Container Activity SMF Records	If you want to gather container activity records, select the check box.
Write Server Interval SMF Records	If you want to gather server interval records, select the check box.
Write Container Interval SMF Records	If you want to gather container interval records, select the check box.
SMF Interval Length	Set the length of recording intervals for SMF recording. Valid when you specify Write Server Interval SMF Records or Write Container Interval SMF Records. The default interval is one hour. You can set the interval from 15 to 86400 seconds (24 hours). If you set this value to 0, the system uses the value from the INTERVAL statement in the SMFPRMxx parmlib member. If there is no INTERVAL statement in SMFPRMxx, the default interval is 30 minutes.
Environment variable list	Check environment variables. See step 5.

_____

5. Set the environment variables for BBOASR2 as follows. For more information about how to add or modify environment variables, use the help system in the Administration application or see *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838.

The following are environment variables to set:

LIBPATH
Include

/usr/lpp/java/IBM/J1.3/bin:/usr/lpp/java/IBM/J1.3/bin/classic:/usr/lpp/WebSphere/lib

CLASSPATH
Should include the following files:
- *path*/bboplsj.jar.
- *path*/bboplc.jar. The default path for these last two files is /usr/lpp/WebSphere/samples/PolicyIVP/PRODUCTION.

Note: After activation of this conversation, System Management automatically prepends ws390srt.jar, waswebc.jar, and xerces.jar to the application server CLASSPATH for you.

If you plan to use procedural application adapters, add the following to CLASSPATH:

```
/usr/lpp/WebSphere/lib/bboadptr.jar:
/usr/lpp/WebSphere/lib/bbokeart.jar:
/usr/lpp/WebSphere/lib/bbokpart.jar
```

JAVA_COMPILER

You do not have to specify JAVA_COMPILER at all, in which case the default is to set the JIT compiler on. If you use the environment variable, a null value (JAVA_COMPILER=) also turns the JIT compiler on. Any other value turns the JIT compiler off.

JVM_LOGFILE

Set it to a file in which you want the log.

Example:

/serverdir/jvm.log

where *serverdir* is a directory to which the BBOASR1 control and server region identities have write access.

PATH

Set it to include the `bin` directory for the SDK.

Some customers move the IVP executables (bbopls, bboplsj, and bboplc DLLs) into the HFS. If you do so, add the directory where they reside at the **beginning** of the PATH.

_____

6. Click the save (diskette) icon. The words "Adding... Servers" appear in the tree.

_____

You know you are done when the following message appears in the status bar:

```
BBON0515I Server BBOASR1 was added.
```

The screen looks like this:
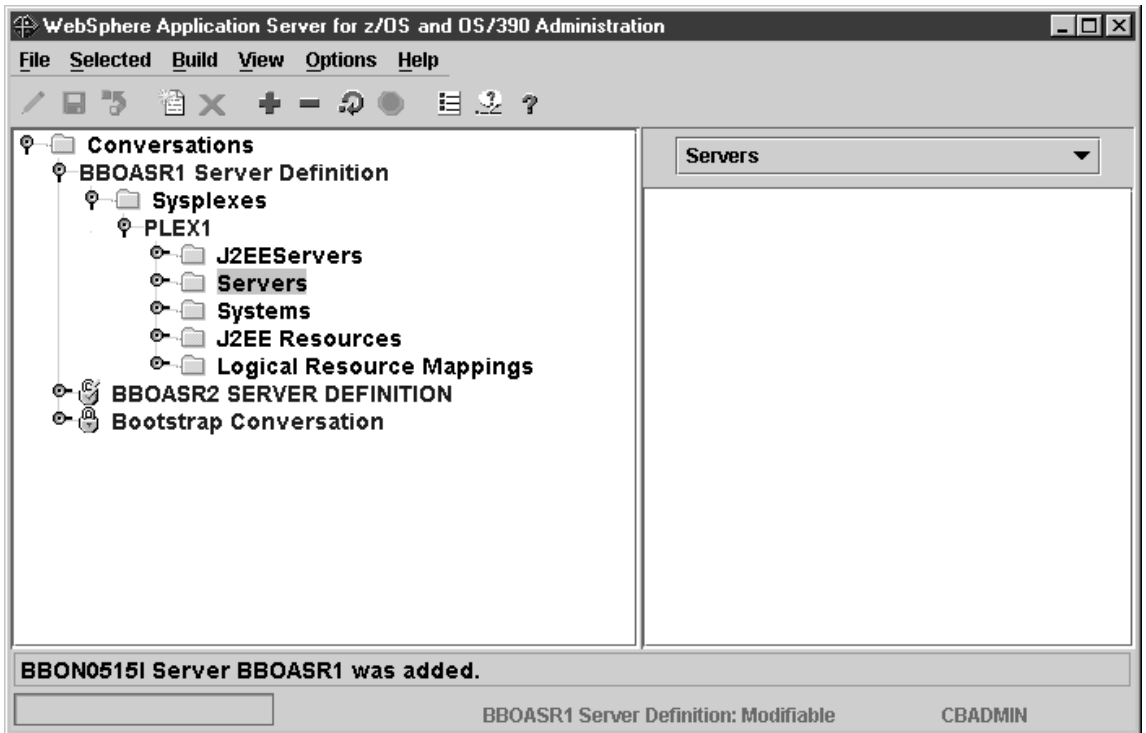
Steps for adding the BBOASR1A server instance
Before you begin: You must have the BBOASR1 server defined.

Perform these steps to add the BBOASR1A server instance:

1. Expand the Servers and BBOASR1 folders by clicking the node to the left of the folder icons.

   _____

2. Select Server Instances with the left mouse button. Then, using the right mouse button, click Server Instances, then select Add.

   _____

3. In the properties form, enter BBOASR1A as the server instance name.

   _____

4. Optional: enter a server instance description.

   _____

5. Optional: supply a log stream name and update the LOGSTREAMNAME environment variable. If you do not, the default is the log stream name you chose for the BBOASR1 server.

   _____

6. Click the save (diskette) icon. The words "Adding... Server Instance" appear in the tree.

   _____

You know you are done when the following message appears in the status bar:

```
BBON0515I Server instance BBOASR1A was added.
```

At the end of this procedure, this is how the screen appears after you expand Server Instances in the tree and select BBOASR1A:
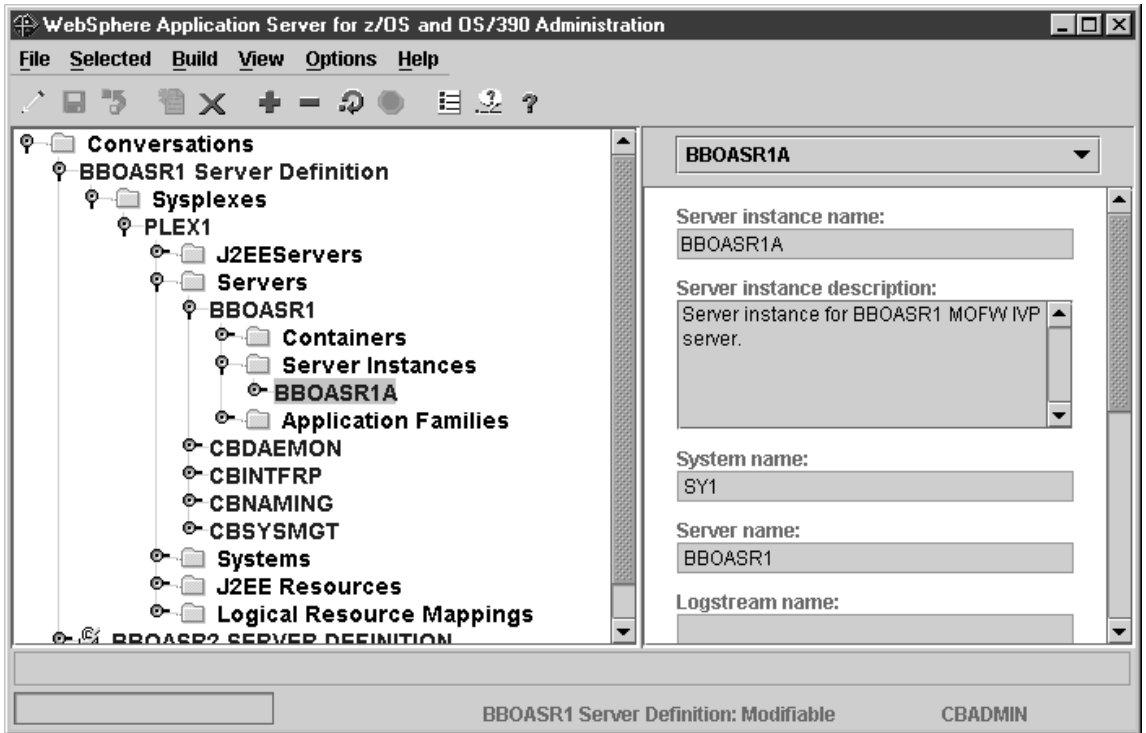
Steps for adding a logical resource mapping
Before you begin: You must be working on the current conversation.

Perform these steps to add a logical resource mapping.

1. Select Logical Resource Mappings with the left mouse button. Then, using the right mouse button, click Logical Resource Mappings, then select Add.

   _____

2. In the properties form, enter CB_OS/390_IVP_DB2 as the Logical Resource Mapping name.

   _____

3. Optional: enter a Logical Resource Mapping description.

   _____

4. Scroll the properties form to LRM subsystem type and select DB2.

   _____

5. Click the save (diskette) icon. The words "Adding... Logical Resource Mappings" appear in the tree.

   _____

You know you are done when the following message appears in the status bar:

BBON0515I Logical resource mapping CB_OS/390_IVP_DB2 was added.

The screen looks like this:
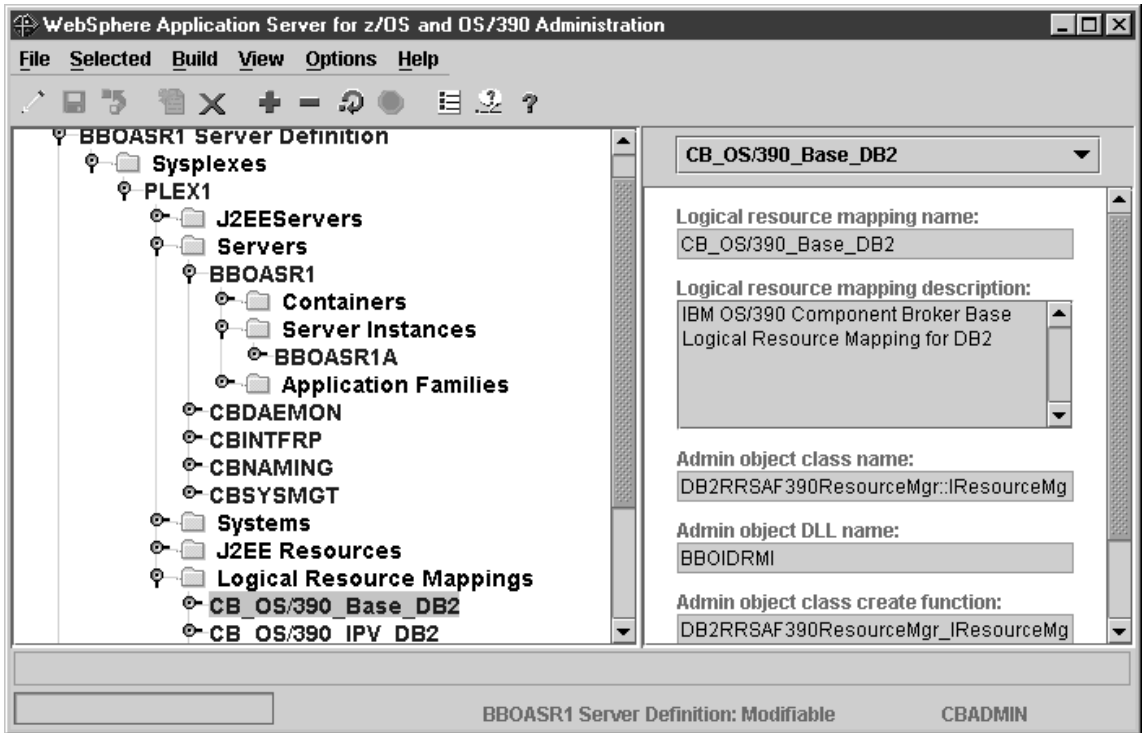
Steps for adding a logical resource mapping instance
Before you begin: You must define the CB_OS/390_IVP_DB2 logical resource mapping.

Perform these steps to add a logical resource mapping instance:

1. If necessary, expand the Logical Resource Mappings folder by clicking the node to the left of the folder icon.

   _____

2. Expand CB_OS/390_IVP_DB2 by clicking the node to the left of the folder icon.

   _____

3. Select LRM Instances with the left mouse button. Then, using the right mouse button, click LRM Instances, then select Add.

   _____

4. In the properties form, enter CB_OS/390_IVP_DB2_*system_name* as the LRM Instance name. The value you supply for *system_name* is, by convention, the system name of the system on which BBOASR1A runs.

   Example: If the system name is SY1, the LRM Instance name would be
   CB_OS/390_IVP_DB2_SY1.

   _____

5. Optional: enter a LRM Instance description.

   _____

6. Select the system this LRM Instance is for.

   _____

7. In the Connection data table, locate "DB2 Subsystem Name" in the Name column. Enter the DB2 subsystem name or group attachment name in the associated Value column. If "CollectionId" appears in the Name column, enter "CBIVP_PKG" in the associated Value column.

   _____

8. Click the save (diskette) icon. The words "Adding... LRM Instances" appear in the tree.

   _____

You know you are done when the following message appears in the status bar:

BBON0515I LRM instance CB_OS/390_IVP_DB2_*system_name* was added.

where *system_name* is the system name you chose.

At the end of this procedure, this is how the screen appears after you expand LRM Instances and select CB_OS/390_IVP_DB2_SY1:
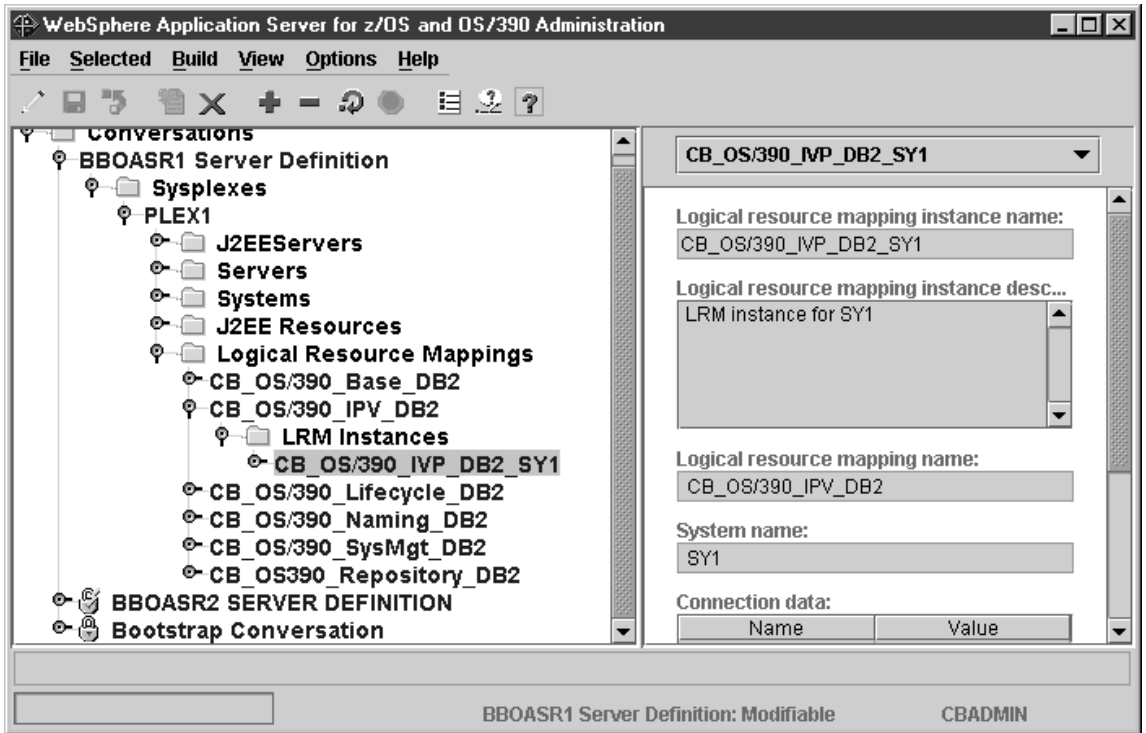
Steps for adding the PolicyHomeObjects container

Before you begin: You must be working on the current conversation.

Perform the following steps to add the PolicyHomeObjects container:

1. If necessary, expand the BBOASR1 folder by clicking the node to the left of the folder icon.

   _____

2. Select Containers with the left mouse button. Then, using the right mouse button, click Containers, then select Add.

   _____

3. In the properties form, enter the container name exactly as shown. The name is case sensitive:

   PolicyHomeObjects

   _____

4. Optional: enter a container description.

   _____

5. Click the save (diskette) icon. The words "Adding... Containers" appear in the tree.

   _____

You know you are done when following message appears in the status bar:

BBON0515I Container PolicyHomeObjects was added.

At the end of this procedure, this is how the screen appears after you expand Containers in the tree and select PolicyHomeObjects:
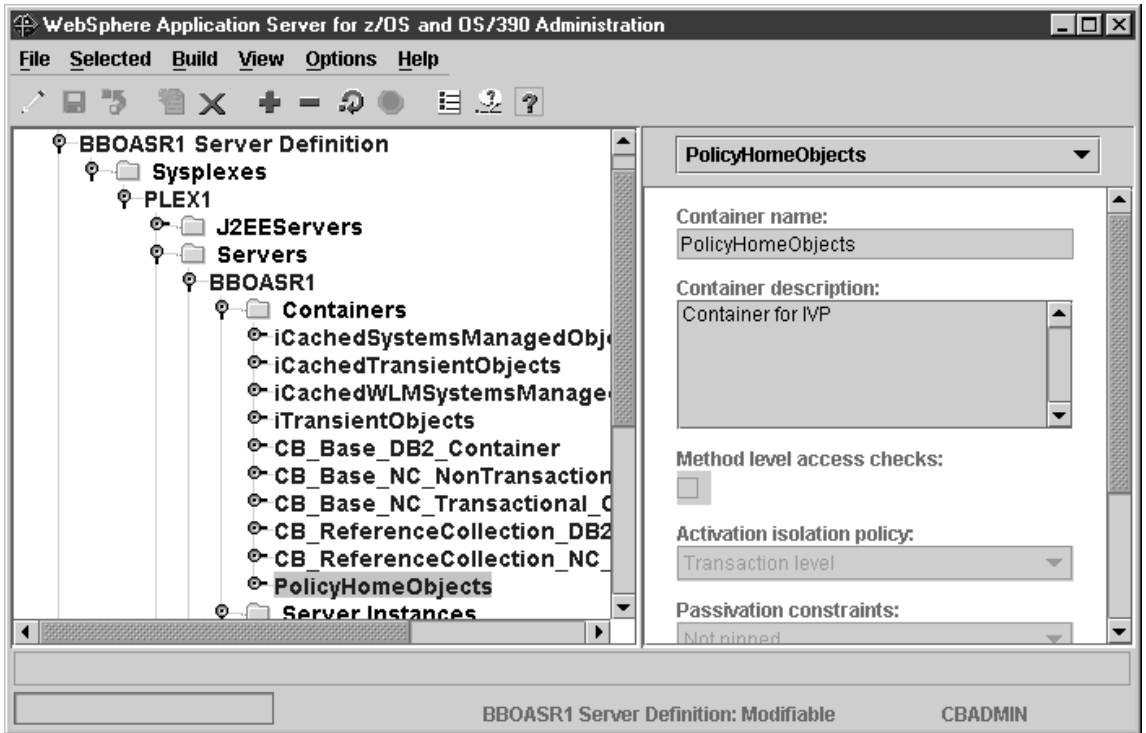
Steps for adding a logical resource manager (LRM) connection for the PolicyHomeObjects container

Before you begin: You must add the PolicyHomeObjects container.

Perform these steps to add a logical resource manager connection for the PolicyHomeObjects container.

1. If necessary, expand the Containers folder under the BBOASR1 server by clicking the node to the left of the folder icon.

   _____

2. Click the node to the left of PolicyHomeObjects.

   _____

3. Select LRM Connections with the left mouse button. Then, using the right mouse button, click LRM Connections, then select Add.

   _____

4. Choose the following as the logical resource mapping name:
   ```
   CB_OS/390_IVP_DB2
   ```

   _____

5. Click the save (diskette) icon. The words "Adding... LRM Connections" appear in the tree.

   _____

You know you are done when the following message appears in the status bar:
```
BBON0547I LRM connection CB_OS/390_IVP_DB2 was added.
```

At the end of this procedure, this is how the screen appears after you expand LRM Connections in the tree and select CB_OS/390_IVP_DB2:
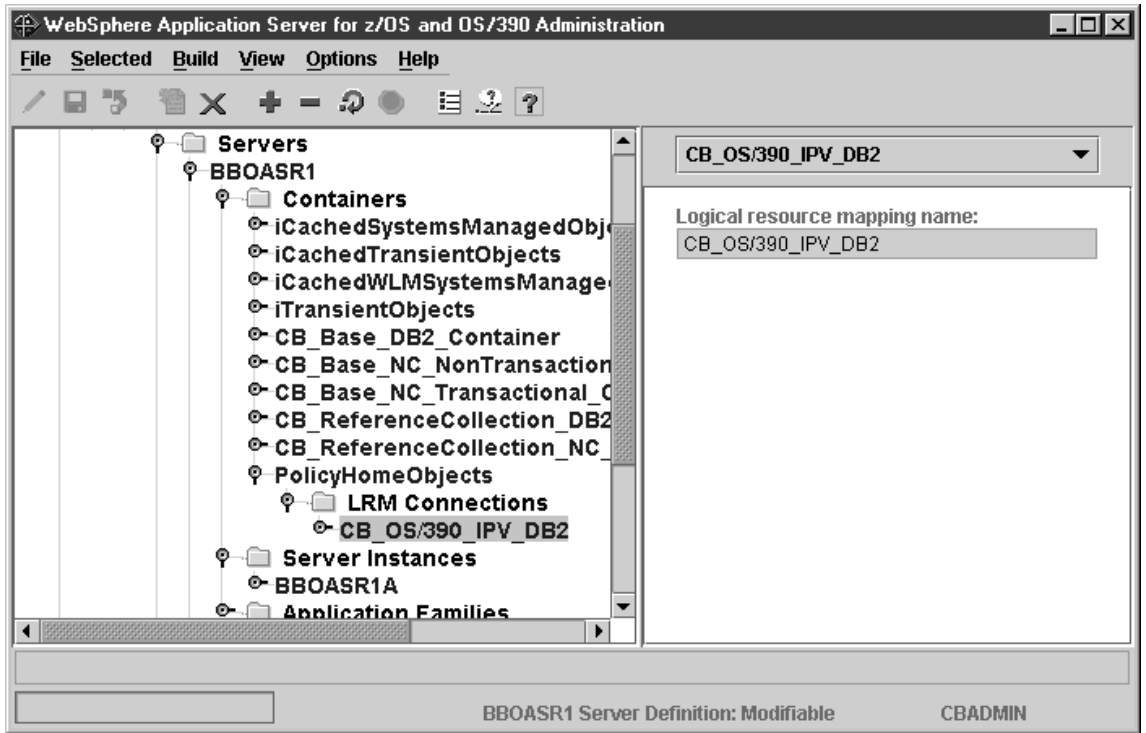
Steps for adding the PolicySQLObjects container

Before you begin: You must be working on the current conversation.

Perform the following steps to add the PolicySQLObjects container:

1. Select Containers with the left mouse button. Then, using the right mouse button, click Containers, then select Add.

   _____

2. In the properties form, enter the container name exactly as shown. The name is case sensitive:

   ```
   PolicySQLObjects
   ```

   _____

3. Optional: enter a container description.

   _____

4. Click the save (diskette) icon. The words "Adding... Containers" appear in the tree.

   _____

You know you are done when the following message appears in the status bar:

```
BBON0515I Container PolicySQLObjects was added.
```

At the end of this procedure, this is how the screen appears after you expand Containers in the tree and select PolicySQLObjects:
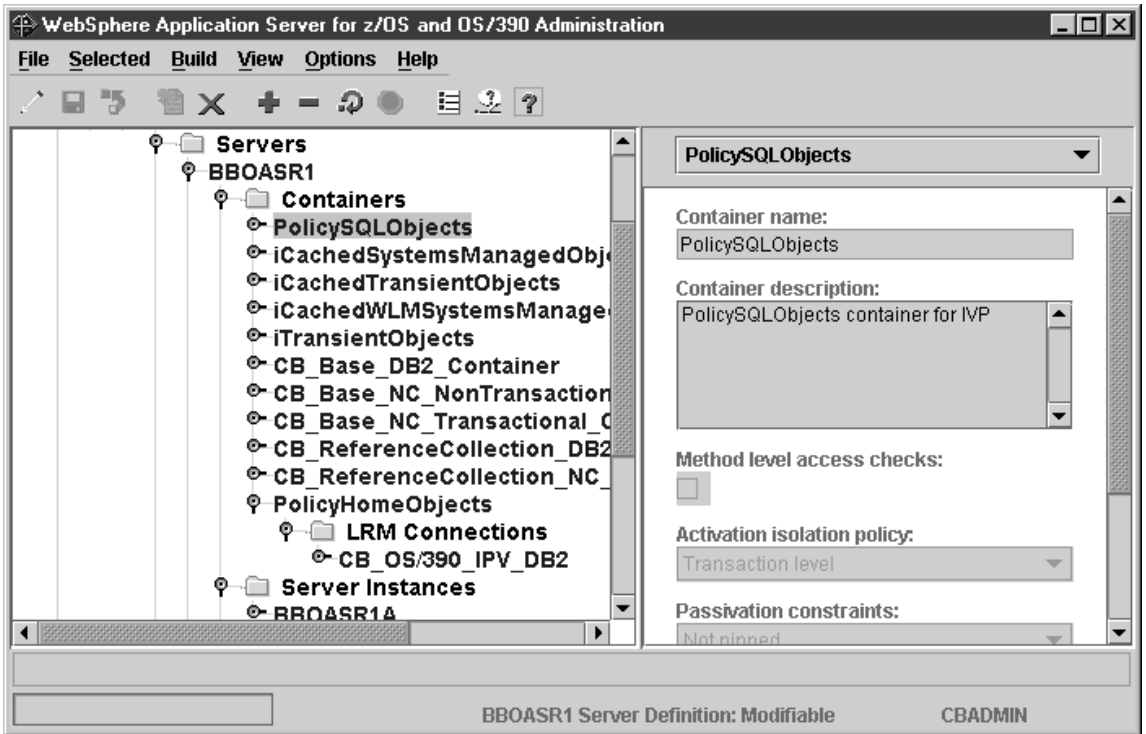
Steps for adding a logical resource manager (LRM) connection for the PolicySQLObjects container
Before you begin: You must add the PolicySQLObjects container.

Perform these steps to add a logical resource manager for the PolicySQLObjects container:

1. If necessary, expand the Containers folder under the BBOASR1 server by clicking the node to the left of the folder icon.

   _____

2. Expand PolicySQLObjects by clicking the node to the left of the folder icon.

   _____

3. Select LRM Connections with the left mouse button. Then, using the right mouse button, click LRM Connections, then select Add.

   _____

4. Choose the following as the logical resource mapping name:

   `CB_OS/390_IVP_DB2`

   _____

5. Click the save (diskette) icon. The words "Adding... LRM Connections" appear in the tree.

   _____

You know you are done when the following message appears in the status bar:

`BBON0547I LRM connection CB_OS/390_IVP_DB2 was added.`

At the end of this procedure, this is how the screen appears after you expand LRM Connections in the tree and select CB_OS/390_IVP_DB2:

Steps for adding the PolicyTransientObjects container
Before you begin: You must be working on the current conversation.

Perform the following steps to add the PolicyTransientObjects container:

1. Select Containers with the left mouse button. Then, using the right mouse button, click Containers, then select Add.

   _____

2. In the properties form, enter the container name exactly as shown. The name is case sensitive:

   PolicyTransientObjects

   _____

3. Optional: enter a container description.

   _____

4. In the properties form, for Activation isolation policy, select **Container level**.

   > **Important!**
   > Choose **Container level**. This is not the default.

   _____

5. Click the save (diskette) icon. The words "Adding... Containers" appear in the tree.

   _____

You know you are done when the following message appears in the status bar:

BBON0515I Container PolicyTransientObjects was added.

Note: An LRM Connection is not required for this container.

At the end of this procedure, this is how the screen appears after you expand Containers in the tree and select PolicyTransientObjects:
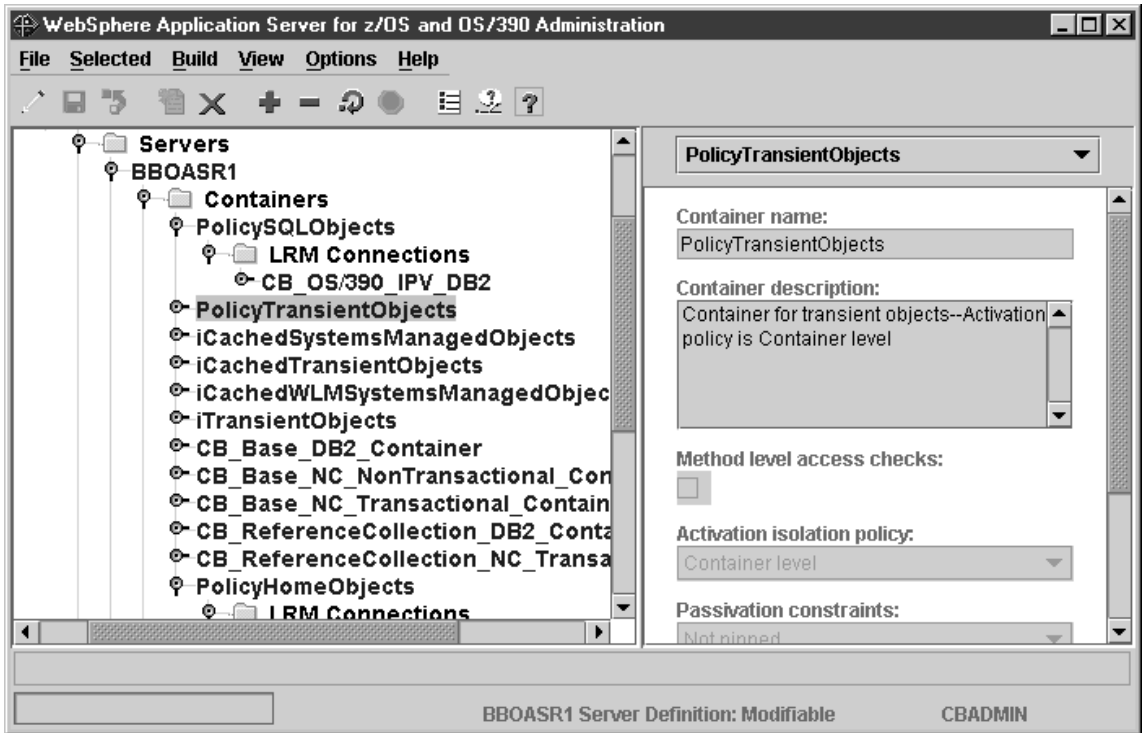
Steps for adding the PolicySQLLocalObjects container

Before you begin: You must be working on the current conversation.

Perform the following steps to add the PolicySQLLocalObjects container:

1. Select Containers with the left mouse button. Then, using the right mouse button, click Containers, then select Add.

   _____

2. In the properties form, enter the container name exactly as shown. The name is case sensitive:

   `PolicySQLLocalObjects`

   _____

3. Optional: enter a container description.

   _____

4. Under Transaction policy, choose **Supports Same-Server Hybrid Global**.

   ```
   ┌─ Important! ────────────────────────────────────────────────────┐
   │ Choose Supports Same-Server Hybrid Global.                      │
   │                                                                 │
   └─────────────────────────────────────────────────────────────────┘
   ```

   _____

5. Click the save (diskette) icon. The words "Adding... Containers" appear in the tree.

   _____

You know you are done when the following message appears in the status bar:

`BBON0515I Container PolicySQLLocalObjects was added.`

The screen looks like this:
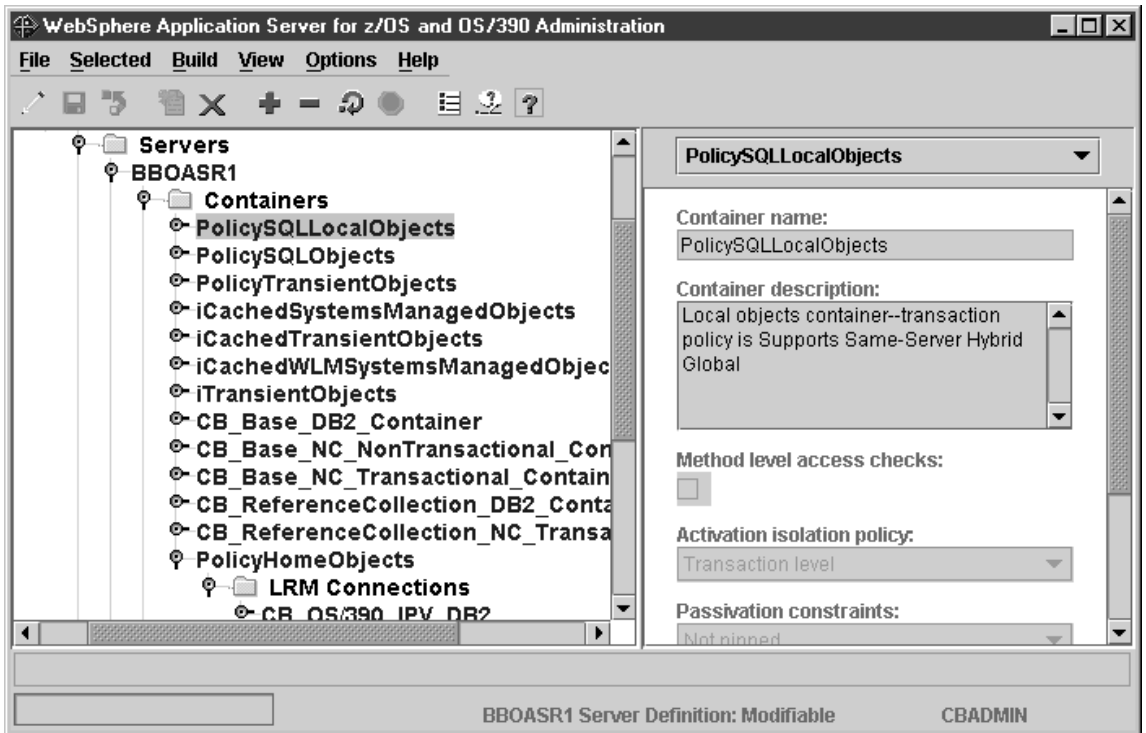
Steps for adding a logical resource manager (LRM) connection for the PolicySQLLocalObjects container
Before you begin: You must add the PolicySQLLocalObjects container.

Perform these steps to add a logical resource manager for the PolicySQLLocalObjects container:

1. If necessary, expand the Containers folder under the BBOASR1 server by clicking the node to the left of the folder icon.

   _____

2. Expand PolicySQLLocalObjects by clicking the node to the left of the folder icon.

   _____

3. Select LRM Connections with the left mouse button. Then, using the right mouse button, click LRM Connections, then select Add.

   _____

4. Choose the following as the logical resource mapping name:
   ```
   CB_OS/390_IVP_DB2
   ```

   _____

5. Click the save (diskette) icon. The words "Adding... LRM Connections" appear in the tree.

   _____

You know you are done when the following message appears in the status bar:
```
BBON0547I LRM connection CB_OS/390_IVP_DB2 was added.
```

At the end of this procedure, this is how the screen appears after you expand LRM Connections and select CB_OS/390_IVP_DB2:
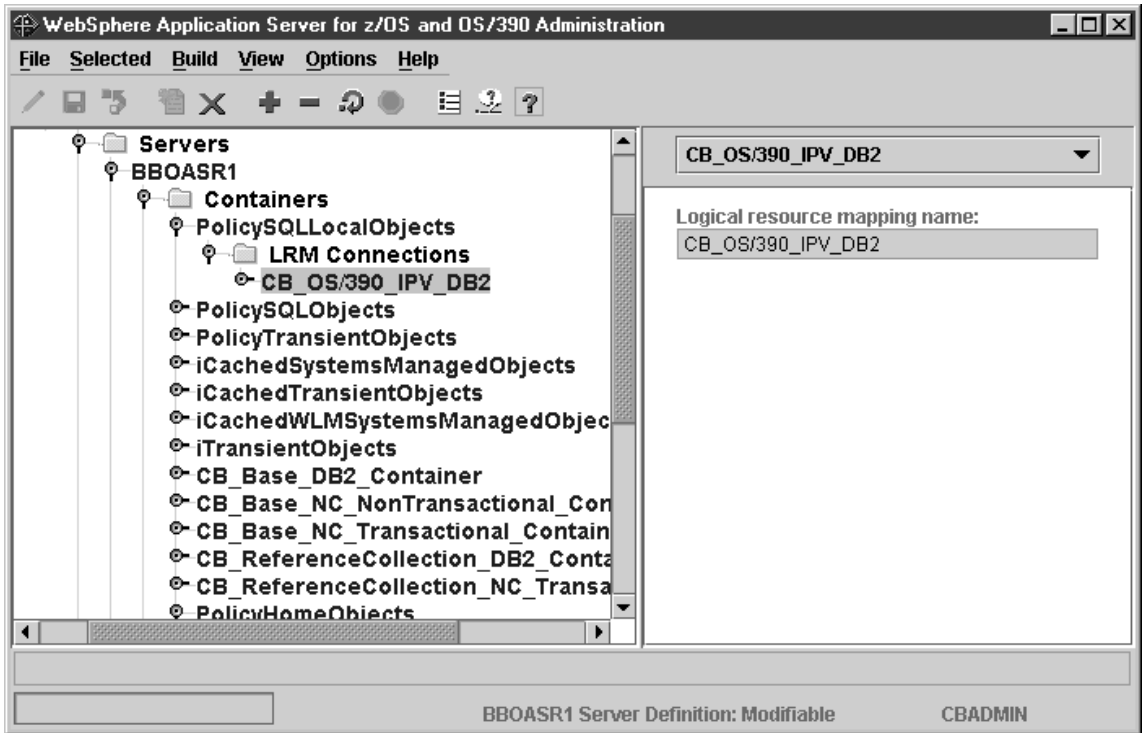
Steps for adding the PolicyTransientLocalObjects container

Before you begin: You must be working on the current conversation.

Perform the following steps to add the PolicyTransientLocalObjects container:

1. Select Containers with the left mouse button. Then, using the right mouse button, click Containers, then select Add.

   _____

2. In the properties form, enter the container name exactly as shown. The name is case sensitive:

   `PolicyTransientLocalObjects`

   _____

3. Optional: enter a container description.

   _____

4. In the properties form, for Activation isolation policy, select **Container level**.

   ┌─ **Important!** ───┐
   │ Choose **Container level**. This is not the default.   │
   └──┘

   _____

5. Under Transaction policy, choose **Supports Same-Server Hybrid Global**.

   ┌─ **Important!** ───┐
   │ Choose **Supports Same-Server Hybrid Global**.   │
   └──┘

   _____

6. Click the save (diskette) icon. The words "Adding... Containers" appear in the tree.

   _____

You know you are done when the following message appears in the status bar:

`BBON0515I Container PolicyTransientLocalObjects was added.`

Note: An LRM Connection is not required for this container.

At the end of this procedure, this is how the screen appears after you expand Containers and select PolicyTransientLocalObjects:
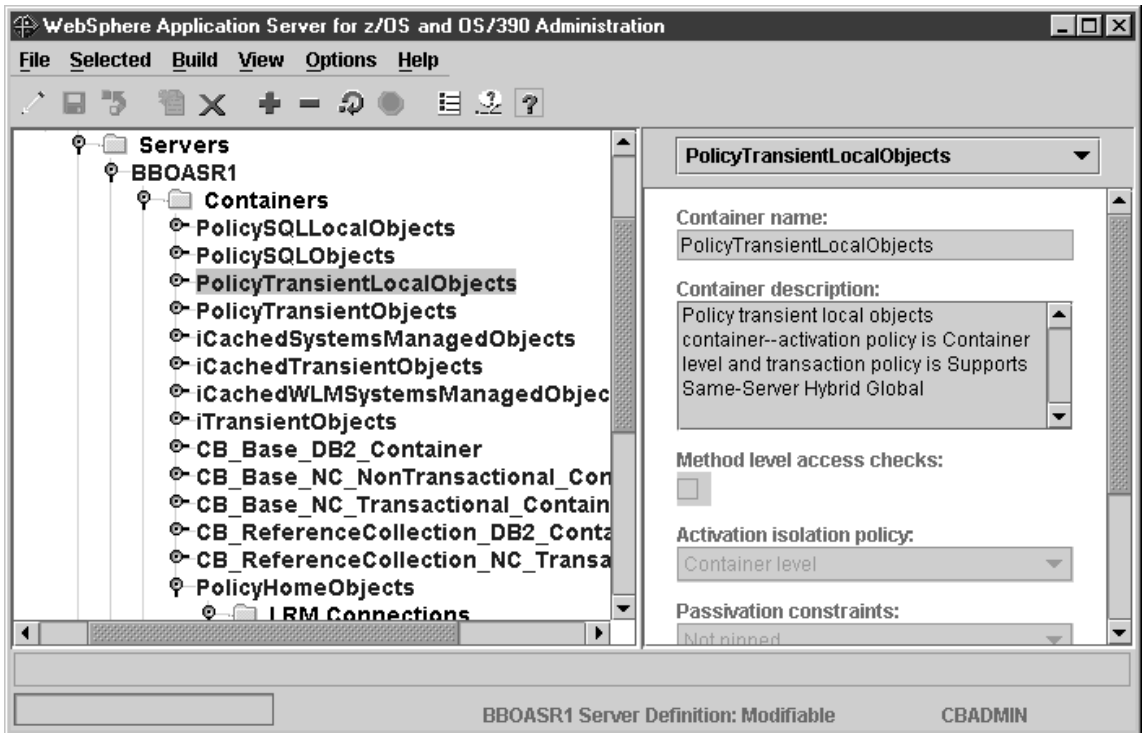
Steps for importing the PolicyFamily application
Before you begin: You must define the BBOASR1 server.

Perform these steps to import the PolicyFamily application:

1. On z/OS or OS/390, mount the WebSphere for z/OS HFS at mount point /usr/lpp/WebSphere.

   _____

2. If necessary, scroll up the conversation tree to the BBOASR1 server. Select BBOASR1 with the left mouse button. Then, using the right mouse button, click BBOASR1, then select Import application.

   _____

3. In the Import dialog, enter the input and output files for the PolicyFamily application. The input file is

   `/usr/lpp/WebSphere/samples/PolicyIVP/PolicyFamily.ddl`

   Rules:

   a. The import and output data sets are associated with the BBOSMSS address space user ID (CBSYMSR1 in our BBOCBRAK sample):
      - If you use data sets, this user ID must have read access to the input data set and alter access to the output data set.
      - If you use HFS files, this user ID must have the ability to search the directories to find the input file, the ability to read the input file, and the ability to write the output file.

   b. Another process cannot be using the import or output data sets used during the import process. For example, you cannot use ISPF to edit or browse the data set or data set member at the same time you start the import.

_____
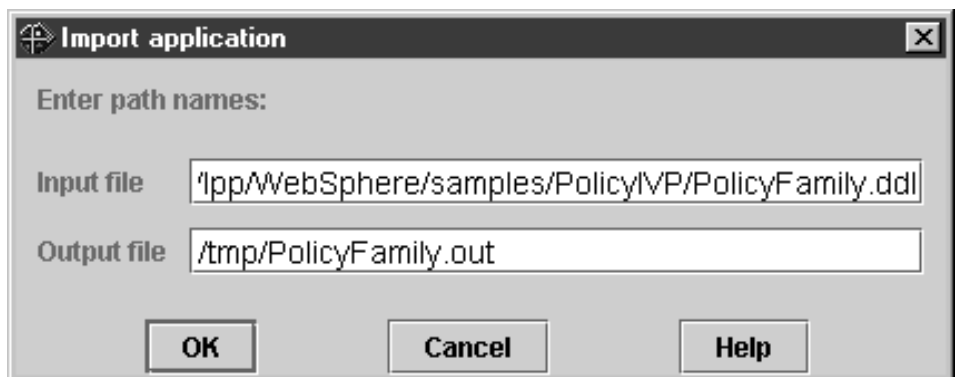
4. Click OK. The words "Importing... BBOASR1" appear in the tree. Wait for the following message:

```
BBON0467I Package file '/usr/lpp/WebSphere/samples/PolicyIVP/PolicyFamily.ddl'
          was imported.
```

_____

5. Click File, then Message log... Check the message log for more detailed error messages by searching for the word "Error" and reading the messages that follow it.

_____

You know you are done when the import succeeds with no errors.

Steps for validating the conversation
Before you begin: You must complete all the previous steps in the current conversation.

Perform the following steps to validate the conversation:

1. If necessary, scroll up the tree to the BBOASR1 Server Definition conversation name.

   _____

2. Select the conversation with the left mouse button. Then, using the right mouse button, click the conversation, then select Validate.

   Result: The words "Validating... BBOASR1 Server Definition" appear in the tree.

   _____

You know you are done when the following message appears in the status bar:

```
BBON0442I Conversation BBOASR1 Server Definition is valid.
```

Step for committing the conversation
Before you begin: You must validate the current conversation.

⇔ Select the conversation with the left mouse button. Then, using the right mouse button, click the conversation, then select Commit. Answer Yes to the question:

```
BBON0534I You cannot undo Commit.  Do you still want to commit?
```
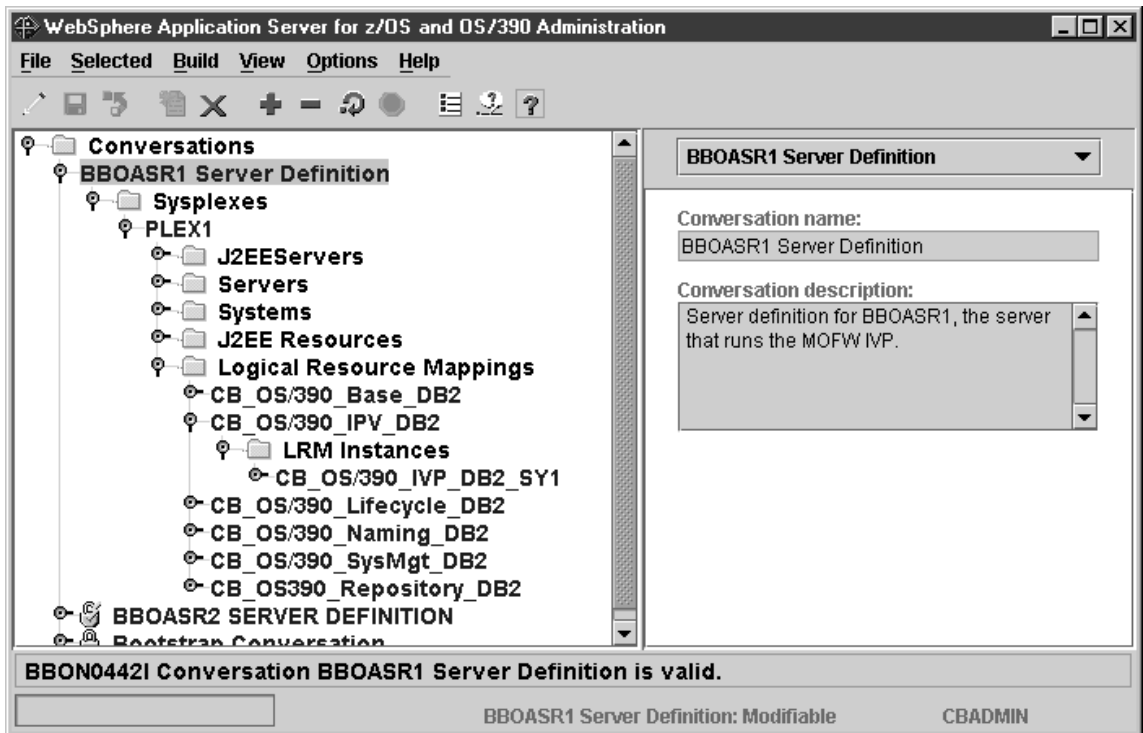
The words "Committing... BBOASR1 Server Definition" appear in the tree.

You know you are done when the following message appears in the status bar:

```
BBON0444I Conversation BBOASR1 Server Definition was committed.
```

Steps for following the instructions for completing z/OS or OS/390 tasks
Before you begin: You must validate and commit the current conversation.

Perform these steps to follow the instructions for completing z/OS or OS/390 tasks:

1. Select the BBOASR1 Server Definition conversation with the left mouse button. Then, using the right mouse button, click the conversation, then select Instructions.

   Result: The words "Getting instructions..." appear in the tree.

   _____

2. Complete all instructions provided by the Administration application for completing z/OS or OS/390 tasks.

   _____

You know you are done when you have completed all the required z/OS or OS/390 tasks.

Steps for marking all tasks complete
Before you begin: You must complete all required z/OS or OS/390 tasks.

Perform these steps to mark all tasks complete:

1. Select the BBOASR1 Server Definition conversation with the left mouse button. Then, with the right mouse button, click the conversation, select Complete, then All tasks.

   _____

2. Answer Yes to the question:

   ```
   BBON0550I  Are you sure that all tasks have been completed?
   ```

   Result: The words "Completing tasks... BBOASR1 Server Definition" appear in the tree.

   _____

You know you are done when the following message appears in the status bar:

```
BBON0484I All tasks complete.
```

The screen looks like this:

WebSphere Application Server for z/OS and OS/390 Administration

File   Selected   Build   View   Options   Help

Conversations
- BBOASR1 Server Definition
- BBOASR2 SERVER DEFINITION
- Bootstrap Conversation

WebSphere for z/OS Host Instructions

Changes to the conversation have been completed and committed; however, there are a set of activities that you must complete outside of the Systems Management Administration tool. The instructions below contain a Completion Summary followed by the tasks you must complete. Each section contains a brief introduction on the tasks that must be completed followed by a list of specific steps based on objects you added or deleted while modifying the conversation. Refer to the WebSphere Application Server for z/OS and OS/390 Installation and Customization book for details on implementing the following tasks.

Completion Summary:

BBON0484I All tasks complete.

BBOASR1 Server Definition: Ready for activate   CBADMIN

Steps for activating your new conversation
Before you begin: You must complete all previous instructions in this section.

Perform these steps to activate your new conversation:

1. Select the BBOASR1 Server Definition conversation with the left mouse button. Then, with the right mouse button, click the conversation, then select Activate.

   _____

2. Answer Yes to the question:

   ```
   BBON0539I  Activate cannot be undone.  Do you want to activate conversation
              BBOASR1 Server Definition?
   ```

   Result: The words "Activating... BBOASR1 Server Definition" appear in the tree.

   _____

You know you are done when the following message appears in the status bar:

```
BBON0449I Conversation BBOASR1 Server Definition was activated.
```

The screen looks like this:

Steps for printing the Administration Message Log

Before you begin: You must activate your conversation.

Follow these steps to print the Administration Message Log:

1. Click File, then Message log...

   Result: The screen looks like this:

```
Administration Message Log                                          _ □ ×
File  Edit  Filter  View  Options  Help

  🌀  ●   🔍   ✂ 🅰 🗟   ?  💻

BBON0507I Initialization complete for WebSphere for z/OS user interface on W ▲
BBON0500I Login in progress for userid CBADMIN, bootstrap server boss0082, port 900.
BBON0170I Initializing ORB.
BBON0171I Connecting to name server.
BBON0172I Resolving administrator reference.
BBON0173I Obtaining application information.
BBON0190I Obtaining capability and function level.
BBON0559I Systems Management Server connected.
BBON0505I Login complete for userid CBADMIN.
BBON0515I Conversation BBOASR1 Server Definition was added.
BBON0563I Default value has been set for Debugger allowed.
BBON0515I Server BBOASR1 was added.
BBON0515I Server instance BBOASR1A was added.
BBON0515I Logical resource mapping CB_OS/390_IPV_DB2 was added.       ▼
```

2. From the Administration Message Log window, click File, then Print...

   Result: You see the Windows print dialog. Select a printer and click ok. You see the following pop-up:

```
Progress...                                          ×

  ℹ   Printing message log...

      [                              ]

              Cancel
```

You know you are done when you get a printout of the Administration
Message Log. You may exit the program.

Steps for creating the database for the installation verification programs (IVPs)

Before you begin: You need your copies of BBOICD, BBOIBN, and BBOIGRT. After you run the customization dialog, these jobs are in *hlq*.CNTL, where *hlq* is the high-level qualifier you chose when you ran the dialog.

Perform the following steps to create the database for the IVP.

1. Submit your copy of BBOICD from a user ID with DB2 SYSADM authority.

   _____

2. Submit your copy of BBOIBN from a user ID with DB2 SYSADM authority.

   _____

3. Submit your copy of BBOIGRT from a user ID with DB2 SYSADM authority.

   _____

You know you are done when the jobs execute successfully.

Running the WebSphere for z/OS installation verification programs (IVPs)

WebSphere for z/OS provides the following installation verification programs (IVPs). Now that you have WebSphere for z/OS customized, you may run one or more, depending on which application servers (BBOASR2 or BBOASR1) you set up and which software components you plan to run on the system.

- The batch client IVP for J2EE (BBOIVPE) is a z/OS or OS/390 client that runs a shell script. The shell script interacts with an enterprise bean running in the BBOASR2 server. For setup and execution instructions, see "Steps for running the BBOIVPE (J2EE) installation verification program".
- Two Web application IVPs:
  - WebSphereSampleApp is a servlet application that you access through your browser. The IVP verifies that your Web container is functioning properly.
  - PolicyIVP has a servlet and a JSP that interact with an enterprise bean running in the BBOASR2 server.

  Both Web application IVPs have a common setup. See "Steps for setting up the Web application IVPs" on page 194. After you do the setup, follow the instructions in "Steps for running the Web application IVPs" on page 198.
- The batch client IVP for CORBA (BBOIVP) is a z/OS or OS/390 client that runs a shell script. The shell script interacts with a CORBA program running in BBOASR1. For setup and execution instructions, see "Steps for running the BBOIVP (MOFW) installation verification program (IVP)" on page 202.

The IVPs are pre-packaged applications. All the application development work has been done for you. If you want to see how to develop applications for WebSphere for z/OS, see *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836, and *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling CORBA Applications*, SA22-7848.

Steps for running the BBOIVPE (J2EE) installation verification program

These instructions explain how to run the BBOIVPE (J2EE) installation verification program. You can find the sample source used for the IVP in the `usr/lpp/WebSphere/samples/PolicyIVP/ejb` directory in the file system after SMP/E installation is complete. This sample runs an enterprise bean.

Before you begin: You need the BBOIVPE job customized by the customization dialog. The job is in *hlq*.CNTL, where *hlq* is the high-level qualifier you chose when you ran the customization dialog.

Perform the following steps to run the BBOIVPE IVP:

1. If you have not already started the LDAP server, do so now.

Example:

```
S BBOLDAP
```

Result: Wait for the following message:

```
GLD0122I Slapd is ready for requests
```

_____

2. If you have not already started WebSphere for z/OS, do so now.
   Example:

   ```
   S BBODMN.DAEMON01
   ```

_____

3. Start the BBOASR2A server instance:

   ```
   s bboasr2.bboasr2a
   ```

   Wait until BBOASR2A fully initializes and you see the following message on the console:

   ```
   BBOU0695I Naming registration completed for server BBOASR2
   ```

   Note: Any administration using the Administration application or System Management Scripting API during server startup and naming registration may cause problems.

_____

4. Edit BBOIVPE. On the JOB statement, change the user ID to correspond to the user ID you chose to run the J2EE IVP in the customization dialog (the default is CBIVP2).

_____

5. Submit BBOIVPE.

_____

You know you are done when BBOIVPE runs successfully.

Steps for setting up the Web application IVPs

Before you begin: You need to set up an HTTP server through which requests can be forwarded to the Web container in the J2EE IVP server, BBOASR2.

Recommendation: Set up an HTTP server specifically for the purpose of testing the IVP. For information, see the HTTP server documentation.

The HTTP server will need its own copies of the httpd.conf and httpd.envvars files. You can find samples of these files in the /usr/lpp/internet/samples directory. Customize the HTTP server cataloged procedure and configuration files to meet your installation requirements. You may also want to ensure the HTTP server starts correctly. Then shut the HTTP server down and continue with the Web application IVP below.

You need the webcontainer.conf file from the /usr/lpp/WebSphere/bin directory.

Perform the following steps to set up the Web application IVPs.

Note: We use the default WebSphere for z/OS installation and customization directory names (/usr/lpp/WebSphere and /WebSphere390/CB390). If you use different directory names, modify the example commands and statements as appropriate.

1. Customize the configuration files used by the Web container in the BBOASR2 server. The configuration files for the Web container can all go in the control file directory for the BBOASR2A server (/WebSphere390/CB390/controlinfo/envfile/*sysplex*/BBOASR2A, where *sysplex* is your sysplex name.)

   a. In the OMVS shell, switch to the control file directory:

      ```
      cd /WebSphere390/CB390/controlinfo/envfile/sysplex/BBOASR2A
      ```

   b. Copy the Web container configuration file, webcontainer.conf, from the WebSphere for z/OS bin directory into the control file directory for the BBOASR2A server:

      ```
      cp /usr/lpp/WebSphere/bin/webcontainer.conf  .
      ```

   c. Edit your copy of the webcontainer.conf. Change the host.default_host.alias statement as follows:

      ```
      host.default_host.alias=host_name:host_port, sysname:host_port
      ```

      where

      host_name
          Is the IP name (or address) of your base z/OS or OS/390 system

      host_port
          Is the port at which the HTTP server listens for the IVP

sysname

Is the z/OS or OS/390 system name of the system on which WebSphere for z/OS is running

host_port

Is the port at which the HTTP server listens for the IVP

d. Create a file called trace.dat in the control file directory. If you want to run with tracing on, insert the following statement into the file:

```
*=all=enabled
```

If you want to run with tracing off, insert the following statement into the file:

```
*=all=disabled
```

_____

e. Create a file called jvm.properties in the control file directory. This file points the BBOASR2 server to the two files you just created. Add the following two properties into the file:

```
com.ibm.ws390.wc.config.filename=/WebSphere390/CB390/controlinfo/envfile/sysplex/BBOASR2A/webcontainer.conf
com.ibm.ws390.trace.settings=/WebSphere390/CB390/controlinfo/envfile/sysplex/BBOASR2A/trace.dat
```

where *sysplex* is the name of your sysplex.

f. Ensure all three files have the same ownership and permissions as the current.env file in the control file directory.

_____

2. Edit the httpd.conf file for the HTTP server. Remove any ServerInit, ServerTerm and Service statements for previous WebSphere versions, and add the following statements.

Rule: Each statement (ServerInit, ServerTerm, Service) must go on a single line in the httpd.conf file.

a. Add these statements to the httpd.conf file:

```
ServerInit /usr/lpp/WebSphere/WebServerPlugIn/bin/was400plugin.so:init_exit /usr/lpp/WebSphere
ServerTerm /usr/lpp/WebSphere/WebServerPlugIn/bin/was400plugin.so:term_exit
Service    /PolicyIVP/*      /usr/lpp/WebSphere/WebServerPlugIn/bin/was400plugin.so:service_exit
Service    /webapp/examples/* /usr/lpp/WebSphere/WebServerPlugIn/bin/was400plugin.so:service_exit
```

b. Change the Port statement to listen on port *host_port*:

```
Port host_port
```

where *host_port* is the same port you specified on the host.default_host.alias statement in the webcontainer.conf file.

c. Change the default user ID used for HTTP clients to PUBLIC (or whatever unrestricted public user ID your installation uses):

```
Userid PUBLIC
```

3. Edit the httpd.envvars file for the HTTP server.

   a. Add or edit the JAVA_HOME environment variable to supply the value for your SDK home directory.

      Example:

      ```
      JAVA_HOME=/usr/lpp/java/IBM/J1.3
      ```

   b. Add the following directory to the NLSPATH environment variable:

      ```
      /usr/lpp/WebSphere/WebServerPlugIn/msg/%L/%N
      ```

   c. Add the following directory to the LIBPATH environment variable:

      ```
       /usr/lpp/WebSphere/wc/lib
      ```

   d. Add the following directory to the CLASSPATH environment variable:

      ```
      /usr/lpp/WebSphere/wc/lib
      ```

4. If you have not already started the LDAP server and WebSphere for z/OS, do so now.

   Example:

   ```
   S BBOLDAP
   S BBODMN.DAEMON01
   ```

5. If BBOASR2A is runnning, stop it:

   ```
   p bboasr2.bboasr2a
   ```

6. Start the BBOASR2A server instance:

   ```
   s bboasr2.bboasr2a
   ```

   Watch for the following message:

   ```
   +Server "BBOASR2A" open for business.
   ```

   Wait until BBOASR2A fully initializes. If you have not previously run BBOASR2A as part of the J2EE IVP, wait for the following message to appear on the MVS console:

   ```
   BBOU0695I Naming registration completed for server BBOASR2
   ```

7. Start the httpd daemon.

   Result: If the initialization process completed successfully, you should receive the following two messages in the HTTP server job output:

   ```
   ............IBM WebSphere Application Server native plugin initialization went OK :-)
   IMW0235I Server is ready.
   ```

> **Tip:** It is possible to get message IMW0235I without the preceding "smiley face" message if the WebSphere for z/OS plug-in did not successfully initialize. If you do not receive message IMW0235I, an error occurred during the Web server initialization process.

_____

You are done when the BBOASR2A server instance and httpd daemon start successfully.

Steps for running the Web application IVPs

Before you begin: You must set up the Web applications. See "Steps for setting up the Web application IVPs" on page 194.

Perform the following steps to run the Web application IVPs:

1. Enter the following URL from your browser:

   ```
   http://host_name:host_port/webapp/examples/index.html
   ```

   where

   host_name
   Is the host.default_host.alias from the webcontainer.conf file.

   host_port
   Is the port at which the HTTP server listens for the IVP

   Result: If the WebSphereSampleApp IVP is successfully installed, you should see the following Web page:
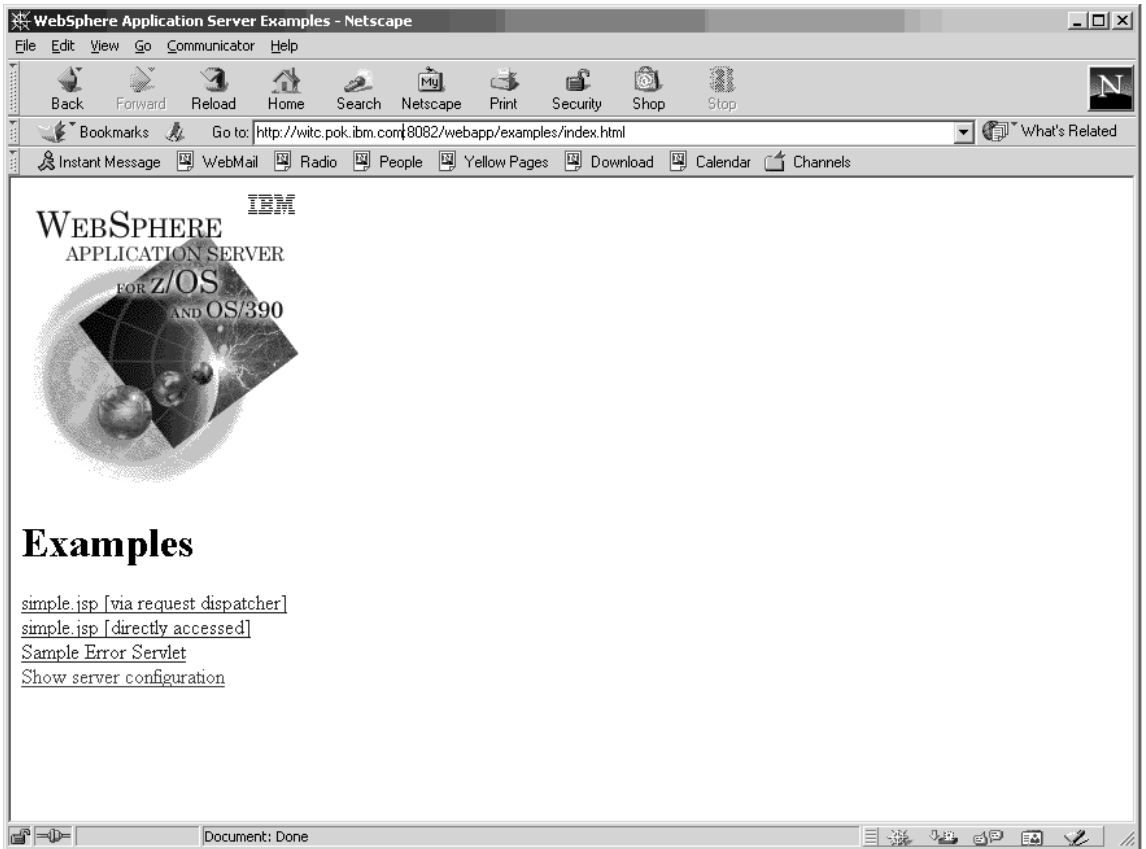
_____

2. From your workstation Web browser, enter the following URL:

   `http://host_name:host_port/PolicyIVP/cebit.html`

   where

   host_name
   > Is the host.default_host.alias from the webcontainer.conf file.

   host_port
   > Is the port at which the HTTP server listens for the IVP

   Result: You see a page like this:

_____
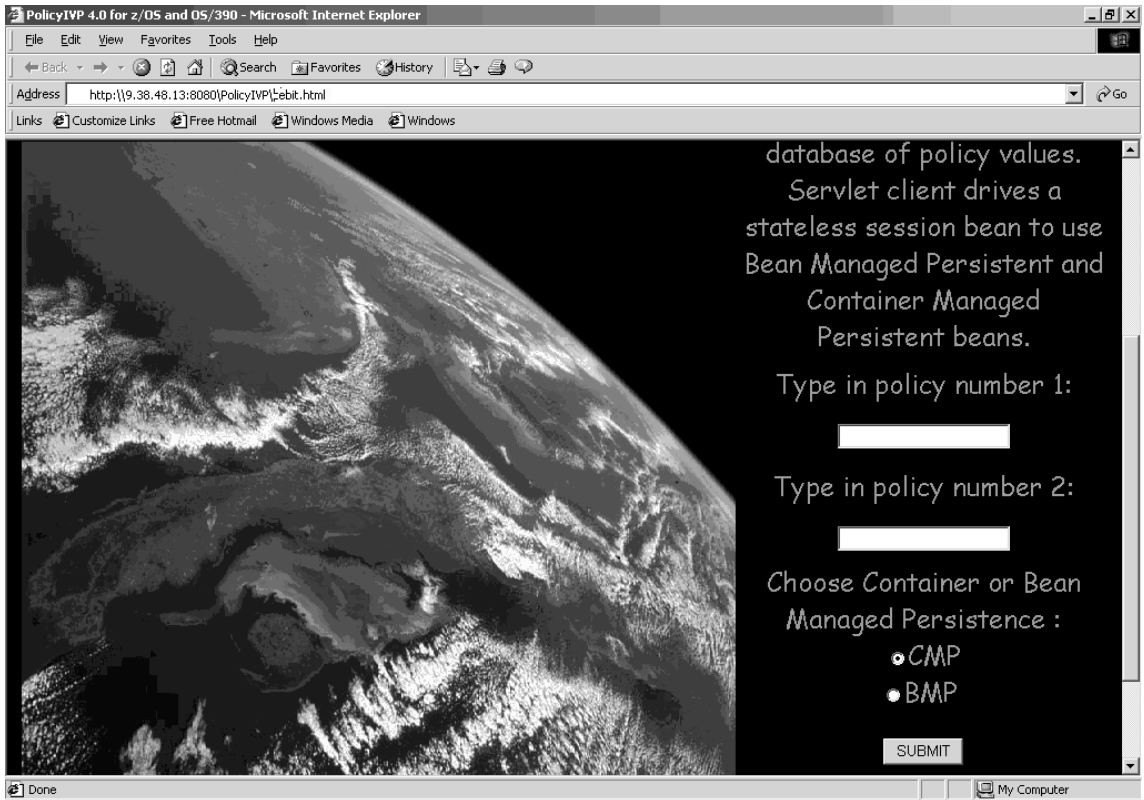
3. On the servlet page:

   a. Type 1111 in the field labelled `Type in policy number 1`.
   b. Type 7777 in the field labelled `Type in policy number 2`.
   c. Click BMP, then Submit.

   Result: You should see a page with the words "BMP IVP has completed successfully.".

_____

4. From your workstation Web browser, enter the following URL:

   `http://host_name:host_port/PolicyIVP/cebit.jsp`

   where

   host_name
   Is the host.default_host.alias from the webcontainer.conf file.

host_port

Is the port at which the HTTP server listens for the IVP

Result: You see a page like this:
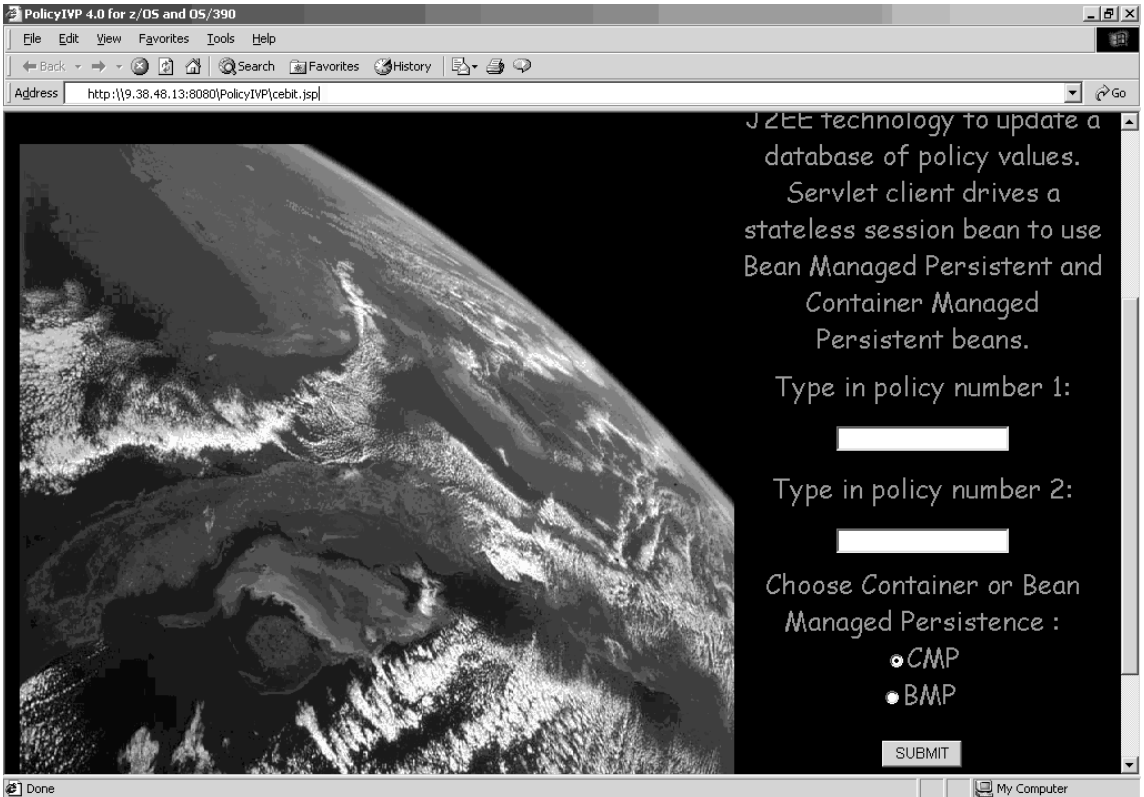
5. On the jsp page:
   a. Type 2222 in the field labelled `Type in policy number 1`.
   b. Type 8888 in the field labelled `Type in policy number 2`.
   c. Click CMP, then Submit.

   Result: You should see a page with the words "CMP IVP has completed successfully.".

6. As a clean-up step, edit the jvm.properties file you created in the control file directory and delete or comment out the com.ibm.ws390.trace.settings environment variable.

_____

You are done when you receive the success messages and do the clean-up step.

Steps for running the BBOIVP (MOFW) installation verification program (IVP)

These instructions explain how to run the BBOIVP (MOFW) installation verification program. You can find the sample source used for the IVP in the `usr/lpp/WebSphere/samples/PolicyIVP` directory in the file system after SMP/E installation is complete. This sample runs a C++ program, followed by a Java program.

Before you begin: You need the BBOIVP job customized by the customization dialog. The job is in *hlq*.CNTL, where *hlq* is the high-level qualifier you chose when you ran the customization dialog.

Perform these steps to set up the IVP and run it:

1. If you have not already started the LDAP server, do so now.

   Example:
   ```
   S BBOLDAP
   ```

   Result: Wait for the following message:
   ```
   GLD0122I Slapd is ready for requests
   ```

   _____

2. Start the BBOASR1A server instance:
   ```
   s bboasr1.bboasr1a
   ```

   Wait until BBOASR1A fully initializes and you see the following message on the console:
   ```
   BBOU0695I Naming registration completed for server BBOASR1
   ```

   Note: Any administration using the Administration application or System Management Scripting API during server startup and naming registration may cause problems.

   _____

3. Copy the environment file for the BBOASR1A server instance (*targetdir*/controlinfo/envfile/*SYSPLEX*/BBOASR1A/current.env) to the directory you specified in the customization dialog as the location of the IVP shell script (the default is /tmp). The environment file will become the environment file for the IVP client run by the BBOIVP job.

For details on the environment variables, see "Appendix A. Environment files" on page 303.

_____

4. Change the file permissions of the file you just copied so that everyone can read the environment file.

   Example:

   ```
   chmod 755 /tmp/current.env
   ```

_____

5. Edit your copy of BBOIVP. On the JOB statement, change the user ID to correspond to the user ID you chose to run the CORBA IVP in the customization dialog (the default is CBIVP).

_____

6. Run the client IVP program by submitting BBOIVP.

_____

You know you are done when you see the following messages in the SYSPRINT output files from the IVP client: the first for the C++ business objects, the second for the Java business objects, and the third for the Java client.

```
All tests completed successfully
All tests completed successfully
Java Client test complete and successful
```

Congratulations

The WebSphere for z/OS installation and customization is now complete. You may now wish to do certain post-installation tasks. See "Chapter 4. Performing post-installation tasks" on page 207.

Chapter supplement

This section provides a general reference for operations and jobs you might need during the installation.

Step for cold-starting RRS

Perform the following step to cold-start RRS:

⇔ Run the following job:

```
//ATRCOLD  JOB  MSGLEVEL=(1,1),REGION=4M
//*
//*01* FUNCTION: DELETES AND REDEFINES THE RRS RESOURCE MANAGER
//*               DATA LOGSTREAM FOR TESTING
//**********************************************************************
//STEP1    EXEC PGM=IXCMIAPU
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
DATA TYPE(LOGR) REPORT(YES)      /* DEFAULT TO SHOW OUTPUT OF JOB */
DELETE LOGSTREAM NAME(ATR.xxxx.RM.DATA)     xxxx = group name
DEFINE LOGSTREAM NAME(ATR.xxxx.RM.DATA)
```

Note: Use the same log stream attributes used to create this log stream.

You know you are done when the job completes successfully.

Steps for checking the contents of the name space

Before you begin: You must have an LDAP server installed.

Perform the following steps to check the contents of the name space.

1. Start the LDAP server.

   Example:

   ```
   S BBOLDAP
   ```

   The message "Starting slapd" will appear on the operator's console, and a message such as "Listening on 0" will appear in the SLAPDOUT data set defined in the job.

   _____

2. Check the job output for the words `done with initial namespace`.

   _____

3. Create a CLIST to search the contents of LDAP (for example, BOSS.SLAPD.CLIST(BBOLSRCH)). Put the following in the CLIST:

   ```
   /* REXX */
   queue('GLDSRCH  -h 127.0.0.1 -p 1389 -b "o=BOSS,c=US"  "objectclass=*"')
   queue('GLDSRCH  -h 127.0.0.1 -p 1389 -b "o=WASNaming,c=US"  "objectclass=*"')
   ```

   _____

4. Execute the CLIST.

   Example: Use ISPF Option 6 to view contents of LDAP by entering:

   ```
   ex 'boss.slapd.clist(bbolsrch)'
   ```

   There will be several screens of output.

   _____

   You know you are done when you see the screen output from the CLIST.

Steps for deleting LDAP entries

If the Naming client job fails during installation and customization, use this procedure to recover.

Due to the structure of the Interface Repository name space, you cannot use this procedure to delete Interface Repository entries.

Attention: After installation and customization is complete, do not use this procedure to recover LDAP tables for Naming or Interface Repository servers unless absolutely necessary. Using this procedure after installation and customization would require that you re-customize WebSphere for z/OS (that is, do a cold start). Use normal backup and data migration procedures for the LDAP tables.

Before you begin: You need the SDELETE module. You will find the SDELETE module in BBO.SBBOLOAD(BBOLSDEL). For more information about SDELETE, see *z/OS Security Server LDAP Server Administration and Use*, SC24-5923.

You must have an LDAP server installed.

To delete the entries:

1. Start the LDAP server.

   Example:

   ```
   S BBOLDAP
   ```

   _____

2. Create a CLIST to delete the LDAP entries (for example, BOSS.SLAPD.CLIST(BBOLSDEL)). Put the following in the CLIST:

   ```
   /* REXX */
   queue('sdelete -h 127.0.0.1 -p 1389 -D "cn=admin,cn=localhost" -w secret
   "TypelessRDN=/,o=BOSS,c=US"')
   ```

   _____

3. Execute the CLIST.

Example: Use ISPF Option 6 to run the CLIST by entering:

```
ex 'boss.slapd.clist(bbolsdel)'
```

---

4. If running BBOLSDEL is unsuccessful:
   a. Drop the LDAP table using BBOLDTBD.
   b. Recreate the LDAP table using BBOLDTBC.
   c. Rerun the bind jobs, BBO1JCL and BBO2JCL.
   d. Rerun the GRANT jobs, BBOCBGRT and BBOLDGRT.
   e. Rerun the LDAP bulk loader (sample BBOLD2DB).

You are done when BBOLD2DB runs successfully.

Handling workload management and server failures

During operations, if your application fails repeatedly, causing the application server regions 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=*
```

---

2. To start the application environment, issue:

```
v wlm,applenv=environment_name,resume
```

where **environment_name** is the application environment name.

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
- Updating the LDAP access control list
- 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. Follow your own backup procedures to back up the LDAP database that contains naming and interface repository data and the JNDI name space.

   If you restore LDAP data, be sure to coordinate the restoration with other WebSphere systems in the federated naming space. Otherwise, your naming space will not be consistent.
4. 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 value of the CBCONFIG environment variable; the default is `/WebSphere390/CB390`).
5. Back up your own application executables, databases, and bindings.
6. When you activate a conversation, System Management automatically backs up the current environment files for each server instance in `/path/envfile/sysplex/server_instance/backup/`, where

   path
   > Is the value of the CBCONFIG environment variable (default is `/WebSphere390/CB390`).

   sysplex
   > Is the name of your sysplex.

server_instance
> Is the name of the server instance.

The backup files have a time stamp in their names. You may wish to erase the older backup files as the backup directory fills up.

7. If you wish to back up a single server instance, you can use the export/import function in the Administration application. For details on how to do this, see *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836.

8. Follow your own backup procedures to back up the WebSphere for z/OS database that contains system management, reference collection, and J2EE data. Back up all table spaces created by jobs BBOMCRDB and BBOICD. Coordinate your backup of the WebSphere for z/OS database with the backup of the LDAP database, because it is very important that both be synchronized. If you restore the database, be sure to coordinate the restoration with that of the LDAP database.

Adding a new administrator for the Administration application

The default administrator for the Administration application is CBADMIN. If you want to add an administrator, you must perform the following tasks:

Subtask	Associated procedure (See . . .)
Creating an MVS user ID or using a current one **Note:** Give the new administrator user ID the same RACF authorizations as CBADMIN.	*z/OS TSO/E Administration*, SA22-7780, or *z/OS SecureWay Security Server RACF Security Administrator's Guide*, SA22-7683
Updating the access control list for LDAP	"Steps for updating the access control list for LDAP"
Defining the new administrator to the Administration application	*WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838
Granting the administrator user ID System Management database authority	"Step for granting the new administrator database authorities" on page 210

Steps for updating the access control list for LDAP

If you add an administrator for the Administration application, you must add that administrator to the access control list in LDAP.

Before you begin: You need to set up the LDAP server. We assume you have already set up an exclusive LDAP server for WebSphere for z/OS administrative purposes. For more information about setting up the LDAP server, see "DB2 database and LDAP" on page 42.

You also need the bboslapd.conf file currently in use by the LDAP server.

Perform the following steps to change the access control list for LDAP:

1. View the bboslapd.conf file and note the following:
   a. Administrator distinguished name.

      Example:
      ```
      adminDN          "cn=CBAdmin"
      ```
   b. Administrator password.

      Example:
      ```
      adminPW          mypass
      ```
   c. Root naming context (RDN) for the WebSphere for z/OS name space structure.

      Example:
      ```
      suffix           "o=BOSS,c=US"
      ```

   _____

2. Start the LDAP server:
   ```
   S BBOLDAP
   ```

   _____

3. Extract the current access control list with the `ldapcp` command.

   Example:
   ```
   /u/myself-> ldapcp -p 1389 -h 127.0.0.1 -d "cn=CBAdmin" -w *****
   GLD6019I Communicating with server on port 1389.
   ldapcp> acl q ob "o=boss,c=us"
    object = o=boss,c=us
    aclSource = O=BOSS,C=US
    aclPropagate = TRUE

    acl = access-id:CBADMIN:object:ad:normal:rwsc

    acl = access-id:CBSYMCR1:object:ad:normal:rwsc

    acl = group:CN=ANYBODY:normal:rsc

    acl = access-id:CN=BOSSAdmin,O=BOSS,C=US:object:ad:normal:rwsc

   ldapcp>quit
   ```

   _____

4. Create a new file in your home directory (for example, acl_update.txt). Add these lines to the file:
   ```
   dn: o=boss, c=us
   changetype:modify
   replace:x
   ```

5. Following the first three lines you added to the file, add aclentry statements for each of the acl lines you extracted in step 3 on page 209. Add a new aclentry statement for USER1.

   Notes:
   a. It is important to add the dash ('-') at the end.
   b. The output format of the ldapcp command is not the same as the input aclentry lines ("acl=" must change to "aclentry:", for example).
   c. The aclentry for USER1 in the example gives USER1 the same authority as CBADMIN.

   Example:
   ```
   aclentry: access-id:cn=BOSSAdmin, o=boss, c=us:normal:rwsc:object:ad
   aclentry: access-id:USER1:normal:rwsc:object:ad
   aclentry: access-id:CBADMIN:normal:rwsc:object:ad
   aclentry: access-id:CBSYMCR1:normal:rwsc:object:ad
   aclentry: group:CN=ANYBODY:normal:rsc
   -
   ```

6. Save the update file and issue the following ldapmodify command:
   ```
   u/myself-> ldapmodify -v -p 1389 -D "cn=CBAdmin" -w mypass -f acl_update.txt
   ```

   Result: ldapmodify responds with:
   ```
   modifying entry o=BOSS, c=US
   ```

7. Repeat step 3 on page 209 to verify that you have added a new user to the access control list.

You know you are done when you see the new user in the access control list.

Step for granting the new administrator database authorities

Your new administrator requires execute authority for CBSYSMGT_PKG and select, update, insert, and delete authority for the tables required for an administrator to deploy a J2EE application in the system management database.

Before you begin: You need to have a user ID with DB2 for OS/390 SYSADM authority.

Perform the following step to grant the new administrator database authorities.

⇔ Issue the following DB2 commands:

```
GRANT EXECUTE ON PACKAGE CBSYSMGT_PKG.*    TO user_ID

GRANT SELECT,UPDATE,INSERT,DELETE ON TABLE
BBO.BBOMT80_J2EEAPP TO user_ID;

GRANT SELECT,UPDATE,INSERT,DELETE ON TABLE
BBO.BBOMT81_MODULE TO user_ID;

GRANT SELECT,UPDATE,INSERT,DELETE ON TABLE
BBO.BBOMT82_COMPONENT TO user_ID;

GRANT SELECT,UPDATE,INSERT,DELETE ON TABLE
BBO.BBOMT83_METHOD TO user_ID;

GRANT SELECT,UPDATE,INSERT,DELETE ON TABLE
BBO.BBOMT86_DATASI TO user_ID;

GRANT SELECT,UPDATE,INSERT,DELETE ON TABLE
BBO.BBOMT87_COMP_DS TO user_ID;
```

where *user_ID* is the administrator user ID you defined.

You know you are done when the GRANT commands succeed.

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 V4.0.1 for z/OS and OS/390: Program Directory*, GI10-0680. 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 Daemon.

Stage	Description
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.
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 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 control regions and server regions need access to the *db2_ssn*.RRSAF profiles, where *db2_ssn* is your DB2 subsystem name. If a control region or server region 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.

We provide a commented section in sample BBOCBRAC that uses the required RACF commands to protect DB2 resources used by WebSphere for z/OS. You can use the sample RACF commands to authorize the WebSphere for z/OS run time or model authorization for your application servers. The sample:

- Defines a DSNR class profile *db2_ssn*.RRSAF, where *db2_ssn* is your DB2 subsystem name.

  Note: For a sysplex, you must define *db2_ssn*.RRSAF class profiles for each DB2 subsystem in the sysplex using their unique subsystem names.

- Gives READ authority to the *db2_ssn*.RRSAF class profile to the following:
  - The Daemon control region
  - The System Management Server control region
  - Every server region

The following table shows the subtasks and associated procedures for setting up RACF protection for DB2 as required by WebSphere for z/OS.

Subtask	Associated procedure (See . . .)
Adding entries to the RACF router table	*DB2 Administration Guide*, SC26-9931
Installing identification and signon exits (DSN3@ATH and DSN3@SGN)	*DB2 Administration Guide*, SC26-9931
Defining RACF user IDs for DB2 started tasks	*DB2 Administration Guide*, SC26-9931
Defining DB2 resources and authorizations required by WebSphere for z/OS in RACF	"Steps for defining DB2 authorizations in RACF"

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 copies of the BBOCBRAJ and BBOCBRAC samples provided with WebSphere for z/OS.

Perform the following steps to define DB2 resources and authorizations in RACF:

1. Edit the BBOCBRAC sample, copy the section labeled "DSNR PROFILES," then paste the section into a new file.

   _____

2. Remove the comment marks that surround the REXX and RACF commands. As shipped, the DSNR profile section is commented out.

   _____

3. Copy the BBOCBRAJ job to a new file.

   _____

4. Change the BBOCBRAC member name in BBOCBRAJ to your new
   member name that has the DSNR profile commands.

   _____

5. 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 and steps for setting
up automatic restart management.

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. LDAP (optional)
6. DCE (if used)
7. The Daemon Server, which automatically starts the System Management
   Server, Naming Server, and Interface Repository Server
8. Your business application servers

For more information about automating WebSphere for z/OS servers, see
*WebSphere Application Server V4.0.1 for z/OS and OS/390: Operations and
Administration*, SA22-7835.

Setting up automatic restart management

If you have an application that is critical for your business, you need facilities
to manage failures. z/OS or OS/390 provides rich automation interfaces that
you can use to detect and recover from failures, but there are some recovery
situations that are too complex to handle with automation. For such
situations, z/OS or OS/390 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 instance (including server instances 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 instance.

Because server instances automatically register with automatic restart management, you must activate the function 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 server instances. 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. We provide information about WebSphere for z/OS's requirements for automatic restart management policies in "Guidelines and restrictions for changing automatic restart management policies for WebSphere for z/OS" on page 216. 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. We do define WebSphere for z/OS's requirements for automatic restart management in "Guidelines and restrictions for changing automatic restart management policies for WebSphere for z/OS" on page 216, but, 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, first read WebSphere for z/OS's requirements for automatic restart management policies in "Guidelines and restrictions for changing automatic restart management policies for WebSphere for z/OS", then 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.

Guidelines and restrictions for changing automatic restart management policies for WebSphere for z/OS

"Setting up automatic restart management" on page 214 led you through the steps to set up automatic restart management for WebSphere for z/OS, but did not discuss changing automatic restart management policies. You are not required to change the automatic restart management policy, but you might want to modify the policy to create custom restart groups. Because server instances register with the default restart group, a system failure means automatic restart management attempts to restart the entire default group on another system in the sysplex and you might want a restart group other than the default.

This section describes guidelines and restrictions for WebSphere for z/OS's use of automatic restart management policies. It is beyond the scope of this

manual to describe how to change the policies. For more information about changing automatic restart management policies, see, *z/OS MVS Setting Up a Sysplex*, SA22-7625.

Follow these guidelines and restrictions:

Restrictions for Version 4.0.1

- Do **not** enable cross-system restart for WebSphere for z/OS server instances. The workload can move from the failing system to a running system, but you cannot bring the workload back to the original system once it is restored. Use the Administrative Data Utility (IXCMIAPU) to change the default ARM policy set by the WebSphere for z/OS servers. See *z/OS MVS Setting Up a Sysplex*, SA22-7625, for details.

End of Restrictions for Version 4.0.1

- If a failure occurs, automatic restart management can restart WebSphere for z/OS and related server instances on the same system.
- To change the policy, you need to know the existing element names for WebSphere for z/OS run-time server instances and how to name new elements for additional run-time server instances.

  The element names for the WebSphere for z/OS run-time server instances are shown in Table 33:

Table 33. Automatic Restart Management element names for WebSphere for z/OS run-time server instances

Server instance	Element name[*]
Daemon	CBDMNDAEMON01
System Management	CBSRVSYSMGT01
Naming	CBSRVNAMING01
Interface Repository	CBSRVINTFRP01

[*] The first server instance has the suffix 01. Each subsequent server instance replica increments the suffix by 1.

  As Table 33 shows, WebSphere for z/OS creates element names for server instances by prefixing the server instance name with CBSRV. The Daemon server instance is an exception: its server instance name is prefixed with CBDMN. For example, the element name for a system management server instance called SYSMGT01 is CBSRVSYSMGT01, but the element name for a Daemon server instance called DAEMON01 is CBDMNDAEMON01.
- Prefix the names of your application server instances with CBSRV. For instance, if your server instance is called MYSERVER, the element name would be CBSRVMYSERVER.

- Do not enable ARM for non-data sharing WebSphere for z/OS configurations in a sysplex (that is, multiple discreet WebSphere for z/OS systems running in a sysplex, but not doing data sharing).
- If you create a restart group, keep the following in the same restart group and set the restart order for the elements as indicated:
  1. RRS
  2. DB2 with IRLM
  3. IMS, CICS, and other transaction or resource managers, if used by your application servers in the restart group
  4. WebSphere for z/OS Daemon server instance
  5. WebSphere for z/OS System Management, Naming, and Interface Repository server instances

     Note: Though the Daemon server instance usually starts the System Management, Naming, and Interface Repository server instances, it does not do so during a restart. Automatic restart management restarts these server instances, so be sure to include them in your restart policy, should you change it.
  6. Your application server instances

Chapter 5. Performing advanced tasks

This section covers advanced tasks, such as sysplex setup, advanced TCP/IP setup, and procedural application adapter setup.

Enabling WebSphere for z/OS on 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 sysplex	"Steps for planning WebSphere for z/OS and sysplex" on page 224
Preparing your security system	"Steps for preparing your security system" on page 226
Setting up data sharing	*DB2 Data Sharing: Planning and Administration*, SC26-9935 "Steps for setting up data sharing" on page 227
Customizing base z/OS or OS/390 functions on the other systems in the sysplex	"Steps for customizing base z/OS or OS/390 functions on the other systems in the sysplex" on page 227

Subtask	Associated procedure (See . . .)
Making changes to TCP/IP	"Steps for making changes to TCP/IP" on page 230
Setting up LDAP files for other systems in the sysplex	"Steps for setting up LDAP files for other systems in the sysplex" on page 231
Defining new WebSphere for z/OS clustered host instances in the sysplex	"Defining new WebSphere for z/OS clustered host instances in the sysplex" on page 233
Refreshing the WebSphere for z/OS systems	"Steps for restarting WebSphere for z/OS on the second system" on page 237
Checking your configuration with the installation verification program	"Steps for running the installation verification program" on page 237

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:

- The WebSphere for z/OS host cluster
- Setting up your sysplex for a rolling upgrade

Overview of the host cluster

To systems and application programs outside of the sysplex, the WebSphere for z/OS sysplex configuration appears to be a single system, even though there may be two or more physical systems within the sysplex. We call such a configuration a *host cluster*, and a single set of WebSphere for z/OS server instances within the host cluster we call a *clustered host instance*.

Example: Figure 6 on page 221 shows an example host cluster, in which each of the three z/OS or OS/390 systems in the sysplex support a WebSphere for z/OS clustered host instance. The triangle in the diagram represents the coupling facility linking the three z/OS or OS/390 systems together.

Host cluster

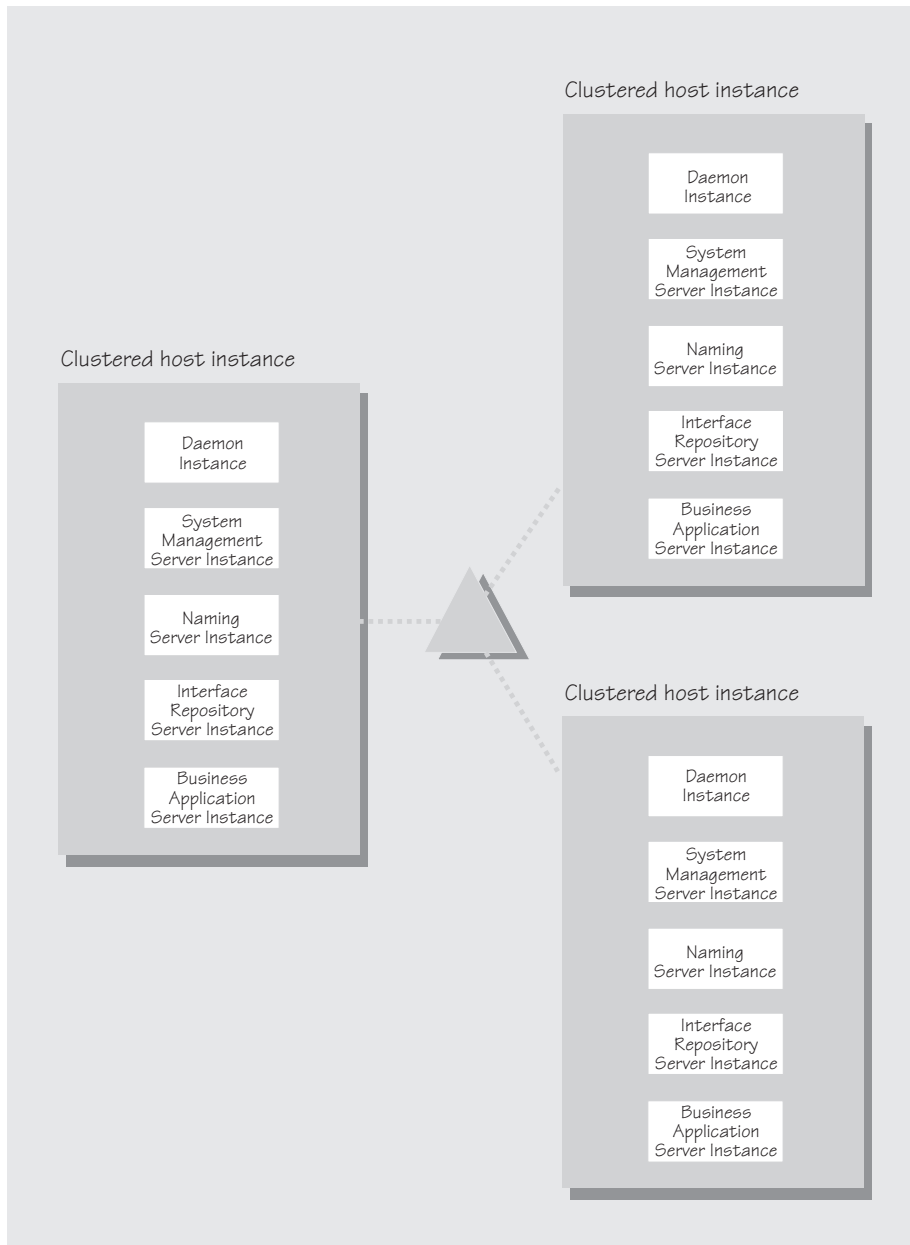

Figure 6. A host cluster

A host cluster is configured into the WebSphere for z/OS name space as a host and is represented by a single Daemon IP Name. Because there is a single Daemon IP Name, systems and applications outside the sysplex treat

the sysplex as a single host. Functions in WebSphere for z/OS, in cooperation with subsystems in z/OS or OS/390, such as TCP/IP, the domain name server (DNS), and workload management, route work through the sysplex according to availability of server instances and workload balancing rules.

Overview of setting up your sysplex for a rolling upgrade
Setting up WebSphere for z/OS in a sysplex allows you to upgrade the system with release and service upgrades by following a method called the *rolling upgrade*. Through the rolling upgrade method, you 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.

A rolling upgrade requires a special Hierarchical File System (HFS) structure. To understand this HFS structure, let us compare a conventional sysplex HFS structure to one using the rolling upgrade.

The conventional way to set up a sysplex is to create the following types of Hierarchical File Systems (HFSes) mounted at the sysplex root.
1. System-specific HFSes that contain directories and files unique to each system in the sysplex. Each system in the sysplex has its own HFS that contains directories such as /dev, /tmp, /var, and /etc.
2. Version-specific HFSes that contain system code and binaries shared by all systems in the sysplex. Typically, there is only one version-specific HFS per sysplex, which contains directories such as /bin, /usr, /lib, /opt, and /samples.

   Note: Because some of WebSphere for z/OS's code is in an HFS mounted off the /usr directory, and the /usr directory is in the version-specific HFS, all systems in the sysplex share the WebSphere for z/OS code.
3. Shared files systems, some of which are in read/write mode, such as /WebSphere390.

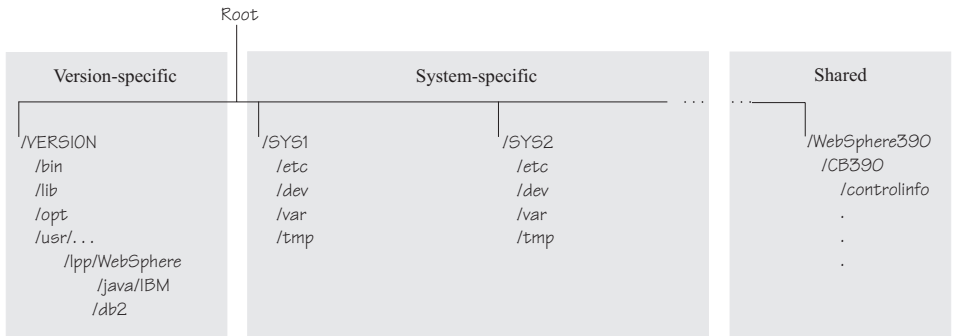Figure 7 on page 223 shows the structure of a conventional sysplex HFS.

```
                                 Root
                                  |
  +----------------------+   +--------------------------------------+   +------------------+
  |  Version-specific    |   |          System-specific             |   |     Shared       |
  +----------------------+   +--------------------------------------+   +------------------+
  |                      |   |                                      |   |                  |
  | /VERSION             |   | /SYS1              /SYS2             |   | /WebSphere390    |
  |   /bin               |   |   /etc               /etc           |   |   /CB390         |
  |   /lib               |   |   /dev               /dev           |   |     /controlinfo |
  |   /opt               |   |   /var               /var           |   |       .          |
  |   /usr/...           |   |   /tmp               /tmp           |   |       .          |
  |     /lpp/WebSphere   |   |                                      |   |       .          |
  |        /java/IBM     |   |                                      |   |                  |
  |        /db2          |   |                                      |   |                  |
  +----------------------+   +--------------------------------------+   +------------------+
```

Figure 7. Conventional sysplex HFS structure

There are other important things to know about the sysplex environment:

1. Each system uses a special symbol, $VERSION, which is equal to a value chosen by the installation. This value is set either in the BBXPRMxx member of PARMLIB or through the SETOMVS command.

2. There are many symbolic links established in the sysplex environment. An important one for versioning is the symbolic link for /usr:

   ```
   /usr --> $VERSION/usr
   ```

   Through this symbolic link, a reference to a file in /usr resolves to a file in $VERSION/usr.

In the conventional sysplex environment, you would only have one version-specific HFS and all systems in the sysplex would share it. By contrast, the rolling upgrade method requires two mount points for version-specific HFSes. To take advantage of the rolling upgrade method, you must set up an additional version-specific HFS when you initially set up your sysplex. Then, when you need to upgrade WebSphere for z/OS, you have the necessary system infrastructure in place to do the upgrades.

Recommendation for the HFS structure: IBM recommends that you set up the sysplex root, WebSphere for z/OS mount points, and z/OS or OS/390 system mount points like Figure 8 on page 224.

Note: For information about an alternate HFS structure you can use, see "Appendix C. Using an alternate HFS structure for product upgrades" on page 345.

```
                                        Root
                                         |

┌─────────────────┐ ┌─────────────────┐ ┌──────────────────────────────────┐  ┌──────────────┐
│ Version-specific │ │ Version-specific │ │         System-specific          │  │   Shared     │
└─────────────────┘ └─────────────────┘ └──────────────────────────────────┘  └──────────────┘

/VersionA           /VersionB            /SYS1              /SYS2                /WebSphere390
 /bin                /bin                 /etc               /etc                 /CB390
 /lib                /lib                 /dev               /dev                  /controlinfo
 /opt                /opt                 /var               /var                  .
 /usr/...            /usr/...             /tmp               /tmp                  .
   /lpp/WebSphere      /lpp/WebSphere                                             .
   /java/IBM           /java/IBM
   /db2                /db2
```
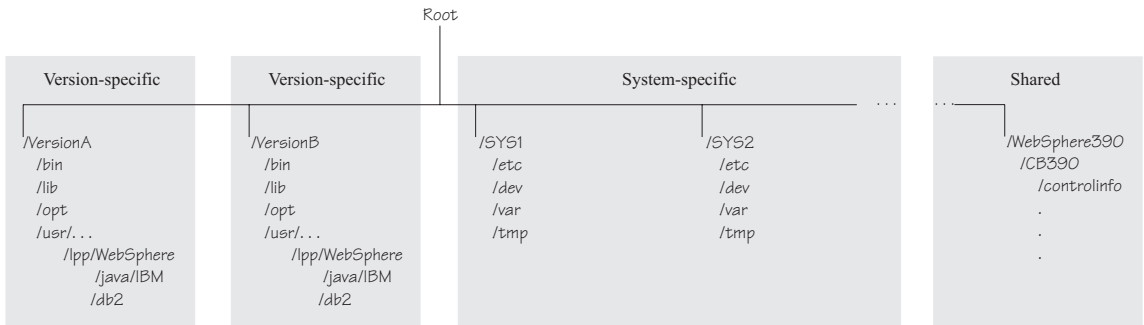
Figure 8. Mount points for rolling upgrade

In Figure 8, there are **two** mount points for version-specific HFSes. Each mount point corresponds to a specific code level of WebSphere for z/OS and associated Java and JDBC HFSes.

Figure 8 also shows system-specific directories (/SYS1 and /SYS2) for each z/OS or OS/390 system in the sysplex and a shared HFS (/WebSphere390).

For now, we will discuss only the first version-specific mount point. The second version-specific mount point becomes important when you need to make a code change for a release or maintenance level (more on that in "Chapter 6. Installing new releases and maintenance levels of WebSphere for z/OS" on page 285). When setting up WebSphere for z/OS in a sysplex initially, you would mount the current code level at the first version-specific mount point.

Example: Assume you have a version-specific HFS for one service level (PTF 10) mounted at /VersionA.

```
mount omvs.ptf10.was.hfs  at /VersionA/usr/lpp/WebSphere
mount omvs.ptf10.java.hfs at /VersionA/usr/lpp/java/IBM
mount omvs.ptf10.jdbc.hfs at /VersionA/usr/lpp/db2
```

The host cluster is running on code from the /VersionA mount point, the version-specific HFS for PTF10. That is, $VERSION on all systems in the sysplex is set to VersionA, so all references to /usr are actually resolved to /VersionA/usr through the symbolic link.

Steps for planning WebSphere for z/OS and sysplex

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 sysplex. This topic covers planning steps for your sysplex deployment.

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 or OS/390 sysplex. For more information on sysplex, see *z/OS MVS Setting Up a Sysplex*, SA22-7625.

Follow these steps to plan WebSphere for z/OS and sysplex:

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. You must establish some means of sharing an HFS in read/write mode across the sysplex. WebSphere for z/OS uses this HFS for writing environment files used by the server start procedures. (For more information, see "Appendix A. Environment files" on page 303.) For z/OS or OS/390 Version 2 Release 8, you must use the Network File System. For z/OS or OS/390 Version 2 Release 9, you can choose either the Network File System or the shared HFS function.

   _____

3. Following z/OS or OS/390 recommendations, set up two version-specific mount points for maintenance, as exemplified in "Overview of setting up your sysplex for a rolling upgrade" on page 222. For more information about sysplex recommendations, see *z/OS Parallel Sysplex Test Report*, SA22-7663.

   _____

4. Decide how you will share application executables in the sysplex. For tips and recommendations, see *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836.

   _____

5. 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).

   _____

6. Decide whether you will run all the WebSphere for z/OS run-time servers on every system in the sysplex.

   Note: The Administration application automatically defines all WebSphere for z/OS run-time servers. Also, the Daemon starts all run-time servers when it initializes. You are allowed, following the recommendations and requirements in the following table, to cancel certain run-time servers after they initialize.

   Recommendations: The following table provides recommendations and requirements for running server instances in a sysplex.

Table 34. Running server instances in a sysplex

Server	Recommendations and requirements for running server instances in a sysplex
Daemon Server and System Management Server	• You must run both these server instances on each system in the sysplex in which you wish WebSphere for z/OS work to run. Thus, you may have some systems in your sysplex 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 Daemon and System Management server instances. • If a server indicates that PassTickets are desirable for interaction with a client, you must run the Daemon and System Management server instances on the system where the z/OS or OS/390 client resides.
Naming Server	• You must have at least one Naming server instance in the sysplex and it must be on the system where you do the WebSphere for z/OS bootstrap. • IBM strongly recommends you run the Naming server instance on each system in the sysplex. If you do not run the Naming server instance on all systems in the sysplex, we recommend you run it on at least one other system for availability. • If you use the rolling upgrade method to change code and functional levels of WebSphere for z/OS, you must run a Naming server on each system in the sysplex. That ensures a Naming server will always be available when you shut down a clustered host instance.
Interface Repository Server	• You must have at least one Interface Repository server instance and it must be on the system where you do the WebSphere for z/OS bootstrap. • You can run this server instance on other systems in the sysplex for availability. • If you have applications that do predicate evaluation queries, IBM recommends you run this server instance on every system in the sysplex.

Steps for preparing your security system

Before you begin: Read the background information about security in "Setting up security" on page 19.

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 sysplex.

   _____

2. Each replicated control region and server region must have the same authorizations throughout the sysplex. You can accomplish this by using generic RACF profiles in the STARTED class and authorizing common user IDs to those profiles. For example, a BBOASR1* profile would cover all start procedures for the BBOASR1 server instances.

   _____

You are done with setting up security on the sysplex.

Steps for setting up data sharing

Before you begin: You must have a coupling facility.

Perform the following steps to set up data sharing:

1. Set up DB2 data sharing. For details, see *DB2 Data Sharing: Planning and Administration*, SC26-9935.

   _____

2. You must have BP32K buffer pools in the coupling facility. Review the number of BP32K buffer pools you have and the size of your DSNDB07 database.

   _____

You know you are done when data sharing is functioning.

Steps for customizing base z/OS or OS/390 functions on the other systems in the sysplex

Repeat the same customizations to base z/OS or OS/390 functions that you did for your initial installation and customization of WebSphere for z/OS. The steps are repeated here for convenience.

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 in BBO.SBBOJCL.

   _____

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 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 35. 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.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.

Table 35. Placing modules in LPA or link list (continued)

Modules	Notes

Notes:

a. You must load members dynamically into LPA because they reside in PDSEs, and OS/390 cannot load members of a PDSE at system initialization time.

   Example: Issue:

   ```
   SETPROG LPA,ADD,MASK=*,DSNAME=hlq.SBBOLOAD
   SETPROG LPA,ADD,MASK=*,DSNAME=hlq.SBBOLPA
   ```

   where *hlq* is the high-level qualifier for your WebSphere for z/OS data sets.

   Attention: Be sure that the size of your LPA can hold the WebSphere for z/OS modules. See "Recommendations for using memory" on page 49.

b. Be sure to purge modules with the same name as those from BBO.SBBOLPA, BBO.SBBOLOAD, or BBO.SBBOLD2 that are already in the LPA.

c. We recommend that you update automation to load WebSphere for z/OS modules into LPA after an IPL. COMMNDxx is not appropriate for this task because the commands execute prior to DFSMS services being made available.

---

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 and all LDAP 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 in BBO.SBBOJCL. 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 Administration application, 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 V4.0.1 for z/OS and OS/390: Operations and Administration*, SA22-7835.

_____

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 replicated server instances, you must adjust the IP names in your DNS. Keep the generic IP name of the Daemon, but add a new IP name for the second and subsequent Daemon server instances. This is important not only for workload balancing, but in the event of a server instance failure: the DNS can direct work to other server instances.

   For more information, see "Connection optimization" on page 238 and "IBM Network Dispatcher" on page 239.

   _____

2. In the TCP/IP profile for each additional system in the sysplex, add port 900 for the resolve IP port and associate it with a new System Management server instance name. By default, WebSphere for z/OS named the first System Management server instance SYSMGT01, and increments the suffix on that name for each new System Management server instance (SYSMGT02, SYSMGT03, and so forth). Thus, on your second system in the sysplex, add port 900 and associate it with SYSMGT02.

   Example:

```
900    TCP    SYSMGT02
```

Follow the same pattern for your third and subsequent systems in the sysplex.

_____

3. In the TCP/IP profile for each additional system in the sysplex, add a port for the Daemon and associate it with a new Daemon server instance name. By default, WebSphere for z/OS uses port 5555 for the Daemon. Also, WebSphere for z/OS names the first Daemon server instance DAEMON01 and increments the suffix on that name for each new Daemon server instance (DAEMON02, DAEMON03, and so forth). Thus, on your second system in the sysplex, add a port and associate it with DAEMON02.

   Example:
   ```
   5555    TCP    DAEMON02
   ```

   Follow the same pattern for your third and subsequent systems in the sysplex.

_____

4. Update the workstation hosts file on the workstation where the Administration application runs to include the IP names of the sysplex and systems running in the sysplex.

   Example: The sysplex name is WSCCB and there are two systems, WSCCB1 and WSCCB2, in the sysplex. The entries in the workstation hosts file would be:
   ```
   #
   9.82.93.1 wsccb1.washington.ibm.com  wsccb1 #CB Daemon_IPname and alias for wsccb1
   #
   9.82.93.2 wsccb2.washington.ibm.com  wsccb2 #CB Daemon_IPname and alias for wsccb2
   #
   9.82.93.1 wsccb.washington.ibm.com   wsccb #CB Daemon_IPname and alias for wsccb
   #
   ```

_____

You should now have completed your TCP/IP updates.

Steps for setting up LDAP files for other systems in the sysplex

You do not need to create a new LDAP database as you did during your initial installation and customization. You need to create unique bboslapd.conf, bboldif.cb, and dsnaoini files for each new system on which Naming and Interface Repository server instances run. This is due to the fact that each dsnaoini file is system-specific and refers to a unique DB2 subsystem. When multiple server instances exist in a multi-system configuration, each Naming and Interface Repository server region must refer to a system-specific dsnaoini file.

We follow the naming convention established for these files during initial installation and customization of WebSphere for z/OS. That is, we use the system name in the filename and data set name for these files and data sets. The steps below tell you how.

Before you begin: You must have LDAP configured for WebSphere for z/OS.

Attention: If you have already set up LDAP as you should have during initial installation and customization, do **not** rerun the table creation, bind, or bulk loader jobs for LDAP. Those jobs will destroy your existing name space.

This procedure assumes you have already created a shared HFS directory for LDAP files during initial installation and customization. The directory is created by the BBOMCFG job and the default directory is /WebSphere390/CB390/*sysplex_name*/etc/ldap, where *sysplex_name* is the name of your sysplex.

Perform the following steps to set up LDAP files for other systems in the sysplex.

1. In /WebSphere390/CB390/*sysplex_name*/etc/ldap, create new bboslapd.conf, bboldif.cb, and dsnaoini files. We suggest the following naming convention:
   - *system*.bboslapd.conf
   - *system*.bboldif.cb
   - *system*.dsnaoini

   where *system* is the name of the second system in the sysplex. Repeat this step for the third and subsequent systems in the sysplex on which you want to deploy WebSphere for z/OS.

   _____

2. Modify each new dsnaoini file to refer to the subsystem name for DB2 on that system. You cannot use the DB2 group attachment name.

   _____

3. If you plan to do rolling upgrades, create a start procedure for an LDAP server on each system in the sysplex.

   Recommendation: Configure LDAP within the sysplex—this assures that naming services are fully transactional. It is possible to configure LDAP as a server outside the sysplex, in which case you would not specify the LDAPCONF environment variable.

   Guideline: You can have one LDAP server in the sysplex, but, if you are concerned about availability of naming services, you should have more than one.

   _____

You should now have the LDAP files you need.

Defining new WebSphere for z/OS clustered host instances in the sysplex

Use the Administration application to define additional systems in the sysplex with their server instances. We assume you have already created the first WebSphere for z/OS system with an application server called BBOASR2, BBOASR1, or both (the application servers used for the installation verification programs).

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 Administration application 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, then DB2. Then start WebSphere for z/OS:

```
S BBODMN.DAEMON01
```

Follow these steps to define the second WebSphere for z/OS system:

1. Log onto the Administration application.

   _____

2. Add a conversation.

   _____

3. Define a second system in the sysplex. The run-time server instances are defined automatically for you.

   _____

4. Check the environment variables for each run-time sever instance on the second system. The environment variables are defined hierarchically in the following order: sysplex, server, then server instance. An environment variable lower in the hierarchy overrides a matching one higher in the hierarchy. Check the environment variables for the following server instances. Some environment variables are common for all systems in the sysplex, while others are unique for each system.

   You must override the following environment variables at the server instance level. Go to the properties form for each run-time server instance and code environment variable values as specified in Table 36.

Table 36. Server instance environment variables in a sysplex

Server	Server instance	Environment variable to change	Value
Daemon	DAEMON02	DM_SPECIFIC_SERVER_NAME	DAEMON02

Table 36. Server instance environment variables in a sysplex (continued)

Server	Server instance	Environment variable to change	Value
		SM_SPECIFIC_SERVER_NAME	SYSMGT02
		NM_SPECIFIC_SERVER_NAME	NAMING02
		IR_SPECIFIC_SERVER_NAME	INTFRP02
		SYS_DB2_SUB_SYSTEM_NAME	Your DB2 subsystem name
		LDAPCONF	The appropriate bboslapd.conf file
		LDAPIRCONF	The appropriate bboslapd.conf file
System Management	SYSMGT02	DM_SPECIFIC_SERVER_NAME	DAEMON02
		SM_SPECIFIC_SERVER_NAME	SYSMGT02
		NM_SPECIFIC_SERVER_NAME	NAMING02
		IR_SPECIFIC_SERVER_NAME	INTFRP02
		SYS_DB2_SUB_SYSTEM_NAME	Your DB2 subsystem name
		LDAPCONF	The appropriate bboslapd.conf file
		LDAPIRCONF	The appropriate bboslapd.conf file
Naming	NAMING02	DM_SPECIFIC_SERVER_NAME	DAEMON02
		SM_SPECIFIC_SERVER_NAME	SYSMGT02
		NM_SPECIFIC_SERVER_NAME	NAMING02
		IR_SPECIFIC_SERVER_NAME	INTFRP02
		SYS_DB2_SUB_SYSTEM_NAME	Your DB2 subsystem name
		com.ibm.was.naming.ldap.masterurl	ldap://*IP_name*:*port*
		LDAPCONF	The appropriate bboslapd.conf file
		LDAPIRCONF	The appropriate bboslapd.conf file
Interface Repository	INTFRP02	DM_SPECIFIC_SERVER_NAME	DAEMON02
		SM_SPECIFIC_SERVER_NAME	SYSMGT02
		NM_SPECIFIC_SERVER_NAME	NAMING02
		IR_SPECIFIC_SERVER_NAME	INTFRP02

Table 36. Server instance environment variables in a sysplex (continued)

Server	Server instance	Environment variable to change	Value
		SYS_DB2_SUB_SYSTEM_NAME	Your DB2 subsystem name
		LDAPCONF	The appropriate bboslapd.conf file
		LDAPIRCONF	The appropriate bboslapd.conf file

5. Specify start procedures to be used by the Daemon Server to start the System Management, Naming, and Interface Repository server instances (control regions) on the second system. After you start the Daemon, it starts these server instance control regions automatically. Specify these start procedures on the SMPROC, NMPROC, and IRPROC environment variables.

   If you do not want additional Naming and Interface Repository server instances in the sysplex, set the NMPROC and IRPROC environment variables to nulls. For guidelines on replicating Naming and Interface Repository server instances, see Table 34 on page 226.

6. For each default LRM (CB_OS/390_Base_DB2, CB_OS/390_Lifecycle_DB2, CB_OS/390_Naming_DB2, CB_OS/390_SysMgt_DB2, and CB_OS390_Repository_DB2) open the LRM instance associated with the second system and add the connection data for the DB2 subsystem on the second system.

You have defined the new WebSphere for z/OS run time. Continue with "Steps for defining new application server instances and activating the conversation".

Steps for defining new application server instances and activating the conversation

This procedure explains how to create new server instances and activate your new conversation.

Before you begin: You must define a second WebSphere for z/OS run-time system.

Follow these steps to define new server instances and activate the conversation:

1. Define new server instances on the second system (for example, a server instance called BBOASR2B or BBOASR1B).

   _____

2. On the properties form for each new server instance, override the environment variable settings according to Table 37.

Table 37. Server instance environment variables in a sysplex

Server	Server instance	Environment variable to change	Value
Business server (BBOASR1, BBOASR2, and so on)	Business server instance (BBOASR1B, BBOASR2B and so on)	DM_SPECIFIC_SERVER_NAME	DAEMON02
		SM_SPECIFIC_SERVER_NAME	SYSMGT02
		NM_SPECIFIC_SERVER_NAME	NAMING02
		IR_SPECIFIC_SERVER_NAME	INTFRP02
		SYS_DB2_SUB_SYSTEM_NAME	Your DB2 subsystem name
		LDAPCONF	The appropriate bboslapd.conf file
		LDAPIRCONF	The appropriate bboslapd.conf file

   _____

3. Create a new LRM instance for CB_OS/390_IVP_DB2 for the second system. Add the connection data for the DB2 subsystem for the second system.

   _____

4. Create a new resource instance associated with BBOASR2_EJB_IVP_RESOURCE for the second system. Add the connection data for the DB2 subsystem for the second system.

   _____

5. Validate the new conversation.

   _____

6. Commit the new conversation.

   _____

7. Complete all tasks.

   _____

8. Mark all tasks complete.

   _____

9. Activate the new conversation.

_____

You are done when the conversation activates successfully.

Steps for restarting WebSphere for z/OS on the second system

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 instance on the second system.

_____

You are done when all server instances initialize on the second system.

Steps for running the installation verification program

Before you begin: You must complete all procedures in this section.

You must have your copy of the BBOIVPE client job, the BBOIVP client job, or both, depending on whether you want to test the J2EE server or the CORBA (MOFW) server.

Follow these steps to run the installation verification program:

1. Run BBOIVPE or BBOIVP (or both) on the new system you have defined.

_____

2. Cancel the local BBOASR2 J2EE server instance or BBOASR1 CORBA (MOFW) server instance and run the corresponding client job locally, forcing the work to move to a server instance on another system in the sysplex.

    Example: Cancel the BBOASR2B server instance on the second system. Leave BBOASR2A running on the first system. Use the Administration application or the CANCEL command:

    ```
    c BBOASR2.BBOASR2B
    ```

    Submit BBOIVPE on the second system.

_____

You are done when the installation verification programs run successfully.

Implementing an advanced TCP/IP network

This topic describes advanced TCP/IP configurations, including:

- The use of multiple TCP/IP stacks on z/OS or OS/390
- Connection optimization, a z/OS or OS/390 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

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* and *z/OS Communications Server: IP Configuration Reference*, SC31-8776, for details.

Connection optimization

Figure 9 on page 239 shows a configuration in which the Domain Name Server cooperates with workload management (WLM) to route client requests throughout a sysplex. 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 sysplex.
- The client needs to know the Daemon IP Name in order to connect to WebSphere for z/OS.
- Each system in the sysplex has the same 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 sysplex as a single system, though its requests may be balanced across systems in the sysplex.
- As part of workload balancing and maximizing performance goals, workload management also routes work requests to systems in the sysplex. This function is possible because WebSphere for z/OS cooperates with workload management (see "Workload management and WebSphere for z/OS" on page 264 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 sysplex.
- 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 sysplex.
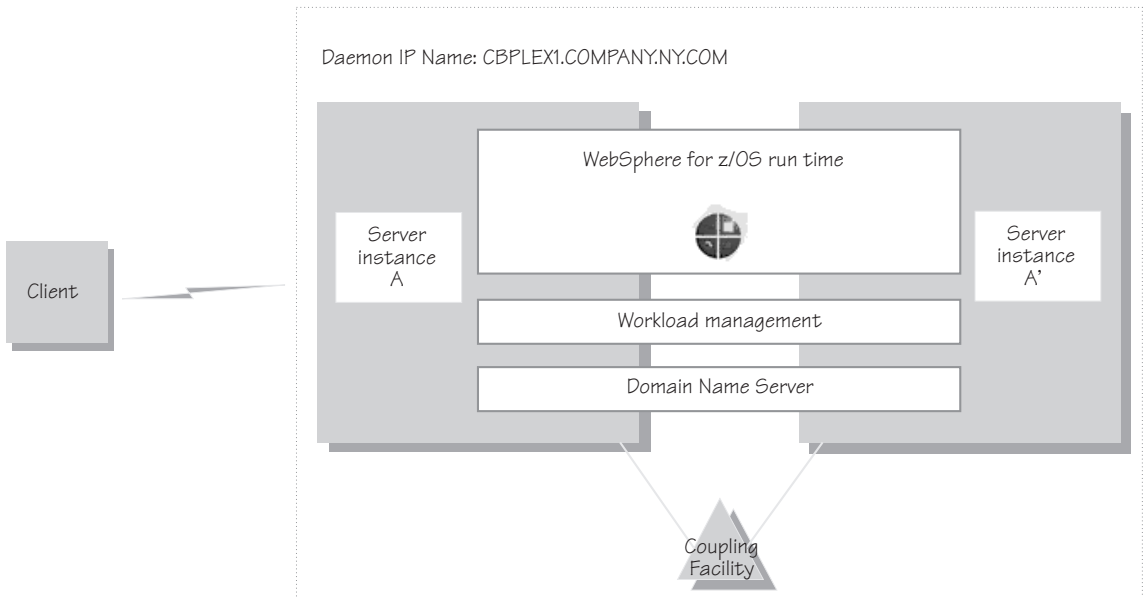

Figure 9. Connection optimization configuration

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 240) is a router that handles network requests for the sysplex. Characteristics of such a configuration are:

- The 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 sysplex. 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.
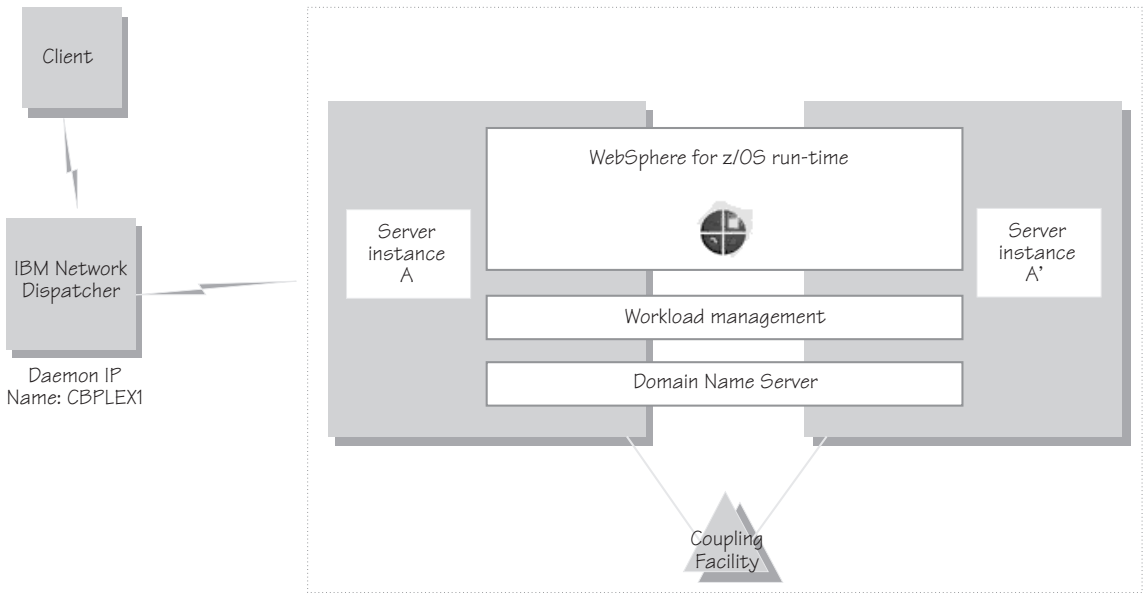
Figure 10. IBM Network Dispatcher configuration

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 or OS/390 system without requiring the client code to configure unique ports. In other words, this support allows use of port 900 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 environment 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 900 servers could share the same TCP/IP stack with other products and applications requiring port 900, because you made their IP addresses unique with SRVIPADDR.

Alternatively, you can, without the use of bind-specific support, define alternate ports for port 900 and the 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 bootstrap port to something other than 900. Configure the ports for the daemon and system management server by specifying port numbers on the DAEMON_PORT and RESOLVE_PORT environment variables.

For details on environment variables, see "Appendix A. Environment files" on page 303.

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 SC31-8776.

Implementing advanced security

This topic covers advanced security issues:
- How clients and servers negotiate security protocols
- Setting up SSL security
- Setting up the asserted identity function
- Setting up Kerberos security

How clients and servers negotiate security protocols

Because there are several security protocols supported by clients and servers, there are many possible ways a client and server can secure their communications. A server may support many security mechanisms simultaneously. At run time, a client and server 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 server they interact with may support SSL, DCE, and user ID/password security. Each client and server 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 server 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 server 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. Currently, the order of security preferences for servers specified through the Administration application is ignored by clients.
2. It is possible that the negotiation between client and server 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 server. Figure 11 shows the types of interactions between clients and servers. The number labels on the diagram are explained in Table 38.
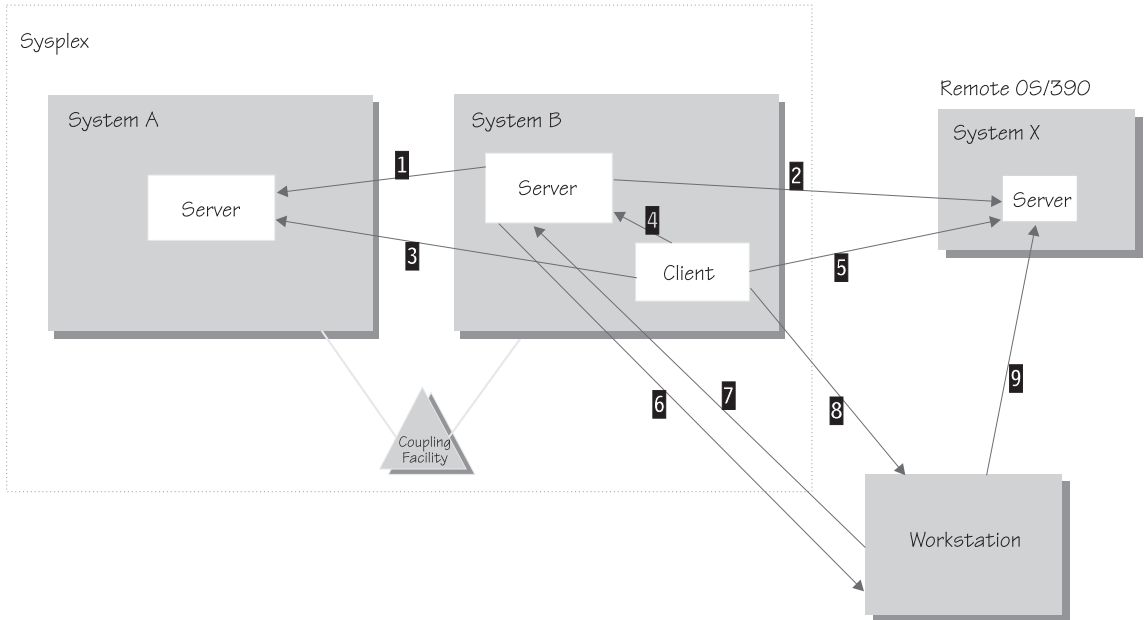

Figure 11. Interactions between clients and servers

Table 38. Ordered list of choices based on interaction

Item	Type of interaction	Ordered list used for this interaction
1	Server to server within the sysplex	1. Kerberos over SSL 2. Asserted identity 3. User ID/PassTicket 4. DCE 5. SSL client certificates 6. User ID/password 7. No security
2	Server to a remote z/OS or OS/390 server	1. Kerberos over SSL 2. Asserted identity 3. DCE 4. SSL client certificates 5. User ID/password 6. No security

Table 38. Ordered list of choices based on interaction (continued)

Item	Type of interaction	Ordered list used for this interaction
3	Client to server within a sysplex	1. SSL client certificates 2. Kerberos over SSL 3. SSL basic authentication 4. User ID/PassTicket 5. DCE 6. User ID/password 7. No security
4	Client to server within a z/OS or OS/390 system	User ID (RACO) always used
5	Client to a remote z/OS or OS/390 server	1. SSL client certificates 2. Kerberos over SSL 3. SSL basic authentication 4. DCE 5. User ID/password 6. No security
6[1]	Server to workstation	1. DCE 2. SSL client certificates 3. No security
7[1] and 9[1]	Workstation to z/OS or OS/390 server	Determined by the workstation client configuration
8[1]	Client to workstation	1. SSL with DCE principal/password authentication 2. DCE 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 DCE and 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.

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.

    - A z/OS or OS/390 client can communicate securely with a WebSphere Application Server Enterprise Edition server on a distributed platform by using a DCE principal and password.

    - 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.

– Kerberos security, in which a server proves its identity by passing a digital certificate to the client. A client proves its identity to the server using Kerberos authentication.

– 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 support is optional: running WebSphere for z/OS without using SSL affects only the SSL functions that protect communication and authenticate clients and servers.

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.

Stage	Description
First client request	The client identity is established during the first request. The determination of client identity depends upon the client authentication mechanism chosen, which is one of the following: • Basic Authentication • SSL client certificates • Kerberos • Asserted identities

Rules:

- Only server control regions and z/OS or OS/390 clients require access to Cryptographic Services System SSL. Your control regions 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.

- 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 Administration application, which, through the "Use SSL Confidentiality Only" attribute, can force the use of the confidentiality level of cipher suites, else the SSL handshake fails.

  (For more information, see *z/OS System Secure Sockets Layer Programming*, SC24-5901.)

- 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.

- The Daemon server does not use SSL.

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, and define SSL security properties for your server through the Administration application.

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 248 shows the certificate arrangement involved in SSL basic authentication.

- **For the client to authenticate the server**, the server (actually, the control region 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.
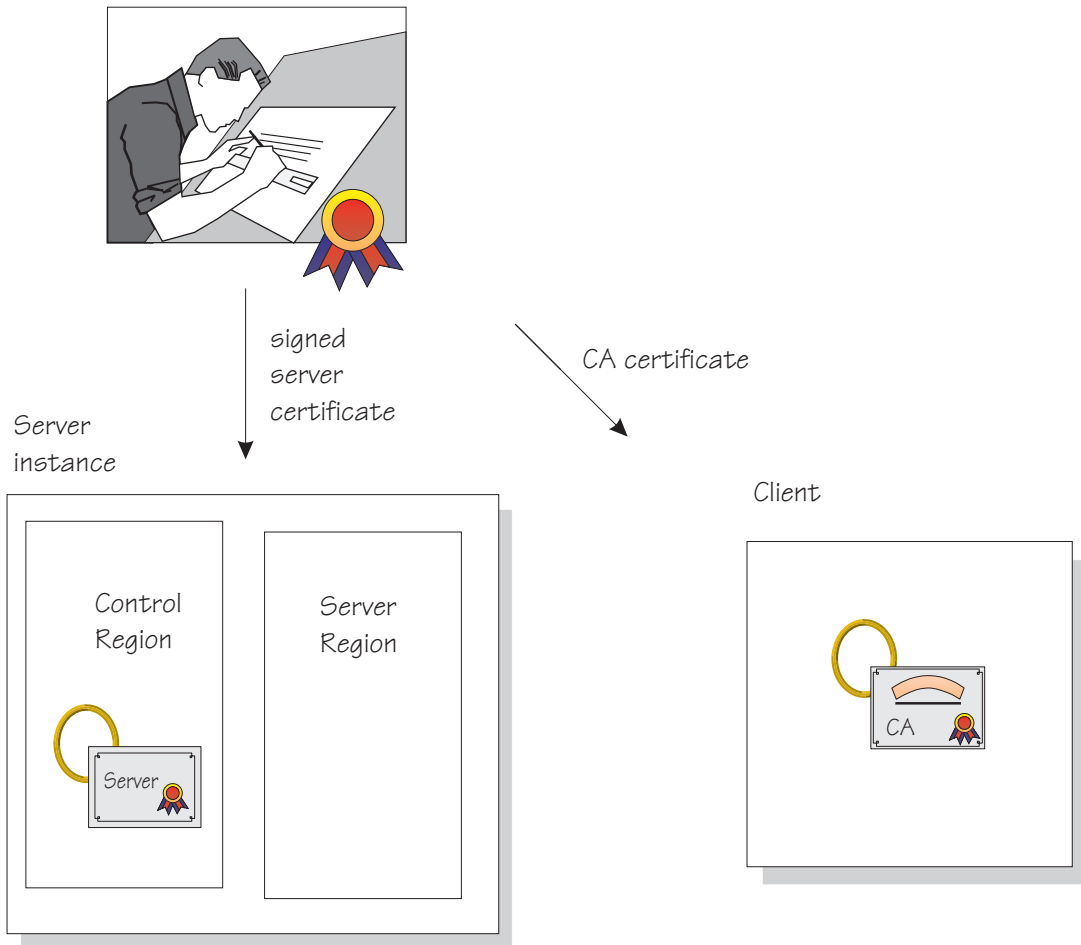
Certificate Authority (CA)

signed server certificate

CA certificate

Server instance

Client

Control Region

Server Region

Server

CA

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) environment 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 environment variables.
- A z/OS or OS/390 client must also identify its key ring through the SSL_KEYRING environment 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
Setting up SSL basic authentication security for servers	"Steps for using RACF to authorize the server to use digital certificates" on page 252
	"Steps for defining server security properties for SSL-based security" on page 254
Setting up SSL basic authentication security for clients	"Steps for setting up SSL security for clients" on page 255

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 Administration application.

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 251 shows the certificate arrangement involved in SSL client certificate authentication.

- **For the client to authenticate the server**, the server (actually, the control region 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 251 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.
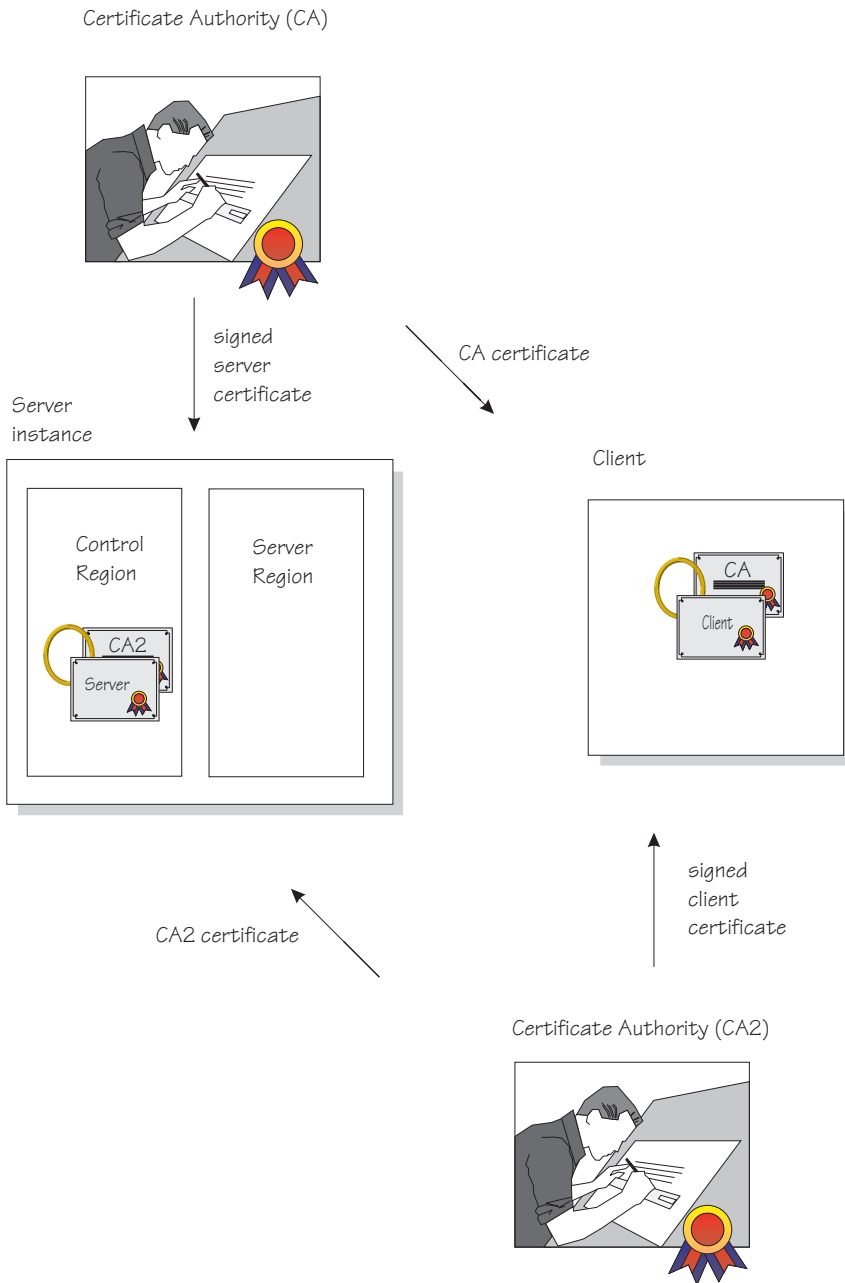
Figure 13. Certificate arrangement for SSL client certificate security

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"
	"Steps for defining server security properties for SSL-based security" on page 254
Setting up SSL client certificate security for clients	"Steps for setting up SSL security for clients" on page 255
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 256

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 control regions 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 SecureWay Security Server RACF Command Language Reference*, SA22-7687, and *z/OS SecureWay 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 control region user ID.

   Example: Your control region is associated with the user ID called CBACRU1. Issue:

   ```
   RACDCERT ADDRING(ACRRING) ID(CBACRU1)
   ```

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 CBACRU1.CA. Issue:

   ```
   RACDCERT ID (CBACRU1) ADD('CBACRU1.CA') WITHLABEL('ACRCERT') PASSWORD('password')
   ```

3. Connect the signed certificate to the control region 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 CBACRU1. Issue:

   ```
   RACDCERT ID(CBACRU1) CONNECT (ID(CBACRU1) 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 control region user ID's key ring.

     Example: Connect the CLIENT1 CA certificate to the ring ACRRING owned by CBACRU1.

     ```
     RACDCERT ID(CBACRU1) 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 control region user ID.

   Example: Your control region user ID is CBACRU1. Issue:

   ```
   PERMIT IRR.DIGTCERT.LIST     CLASS(FACILITY) ID(CBACRU1) ACC(READ)
   PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(CBACRU1) ACC(READ)
   ```

You are done with the RACF phase when the RACF commands succeed. Continue on to "Steps for defining server security properties for SSL-based security" on page 254.

Steps for defining server security properties for SSL-based security: This procedure tells you how to specify that a server use SSL client certificate security through the Administration application.

Before you begin: You need to start the Administration application, log on, and create a new conversation. For more information, see *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838.

Perform the following steps to define security characteristics for the server:

1. Expand Servers in the Conversations tree.

   _____

2. Create a new server, or click the name of your existing server.

   _____

3. In the properties form:
   - If you are implementing SSL basic authentication, click the SSL Type 1 (basic authentication) check box.
   - If you are implementing SSL client certificates, click the SSL Client Certificates check box.
   - If you are implementing Kerberos, click the Kerberos check box.
   - If you are implementing asserted identities, click the Asserted identity check box. Be sure to also click the SSL client certificates check box.
   - If you require the confidentiality of cipher suites, click the SSL Confidentiality Only check 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 252.

   Note: If you specify the wrong RACF key ring, the server gets an error message at run time.

   _____

5. Specify the SSL V2 timeout value, which is the length of time, in seconds, that the system holds session keys. The range is 0-100 seconds. The default is 100 seconds.

   _____

6. 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.

   _____

7. Order the security preference list. For more information about the security preference list, see "How clients and servers negotiate security protocols" on page 241.

_____

8. Complete all other specifications for the server, then validate, commit, complete all tasks, and activate the conversation.

_____

You know you are done when the system tells you the conversation is activated.

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 SecureWay Security Server RACF Command Language Reference*, SA22-7687, and *z/OS SecureWay 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.CBSERVER.CA. Issue this command:

   ```
   RACDCERT ADD('USER.CBSERVER.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 SSL_KEYRING environment variable to correspond to the client's key ring.

   For more information, see "Appendix A. Environment files" on page 303.

   _____

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 sysplex, 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 SecureWay 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.

Setting up the asserted identity function

SSL client certificate support provides a function called asserted identity, in which an intermediate server can send the identities of its clients to a target server in a secure yet efficient manner. This function requires client certificate support to establish the intermediate server as the owner of the SSL session. Through RACF, the system can check that the intermediate server can be trusted (special RACF permission is given to the address spaces, such as control regions, that run secure system code). 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.

Steps for setting up the asserted identity function

Before you begin: The target server must be set up for SSL client certificate support, but the certificates it receives are those from the intermediate servers. The intermediate servers must run on a system that has SSL configured, but do not have to have SSL client certificate support enabled. See "Overview of SSL client certificate security for your application server and clients" on page 250.

Perform the following steps to set up the asserted identity function:

1. Open the Administration application and log on. Start a new conversation. If necessary, define new servers.

   _____

2. For the server that will receive an asserted identity (the target server), add these properties in the properties form:
   - Accept asserted identity allowed
   - SSL client certificates allowed

   _____

3. For the server that will send asserted identities (the intermediate server), specify "Send asserted identities allowed" on its properties form. Fill in the values for the SSL-related elements (SSL RACF keyring, SSL V2 timeout, and SSL V3 timeout).

   _____

4. Validate, commit, and activate the conversation.

   _____

5. On z/OS or OS/390, give CONTROL authority for CB.BIND.*servername* to the user ID (*controlRegionUserid*) of the intermediate server's control region, where *servername* is the **target** server's name.

   ```
   PERMIT CB.BIND.servername CLASS(CBIND) ID(controlRegionUserid) ACCESS(CONTROL)
   ```

Attention: Scrutinize the user IDs that receive CONTROL authority. Such authority should be granted to system code (control regions) only. If you use RACF certificate name filtering to map digital certificates to user IDs, you may inadvertently grant CONTROL authority to users that should not have this powerful authority.

_____

6. Activate the CBIND class.

_____

You are done when you have finished the RACF commands.

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.

Stage	Description
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 SecureWay 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 252
Associating the server identity with a Kerberos principal	"Step associating a server identity with a Kerberos principal"
Defining server attributes for Kerberos	"Steps for defining server security properties for SSL-based security" on page 254
Setting up a client to use Kerberos	"Steps for setting up a client to use Kerberos" on page 262

Step associating a server identity with a Kerberos principal
Before you begin: You need to have a RACF user ID established for the server's control region.

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 SecureWay 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 control region 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 SecureWay 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(CBIVP) 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.

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 or OS/390 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 sysplex by using the domain name server (DNS). Figure 14 on page 265 shows how work gets routed in the sysplex. 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 sysplex 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.
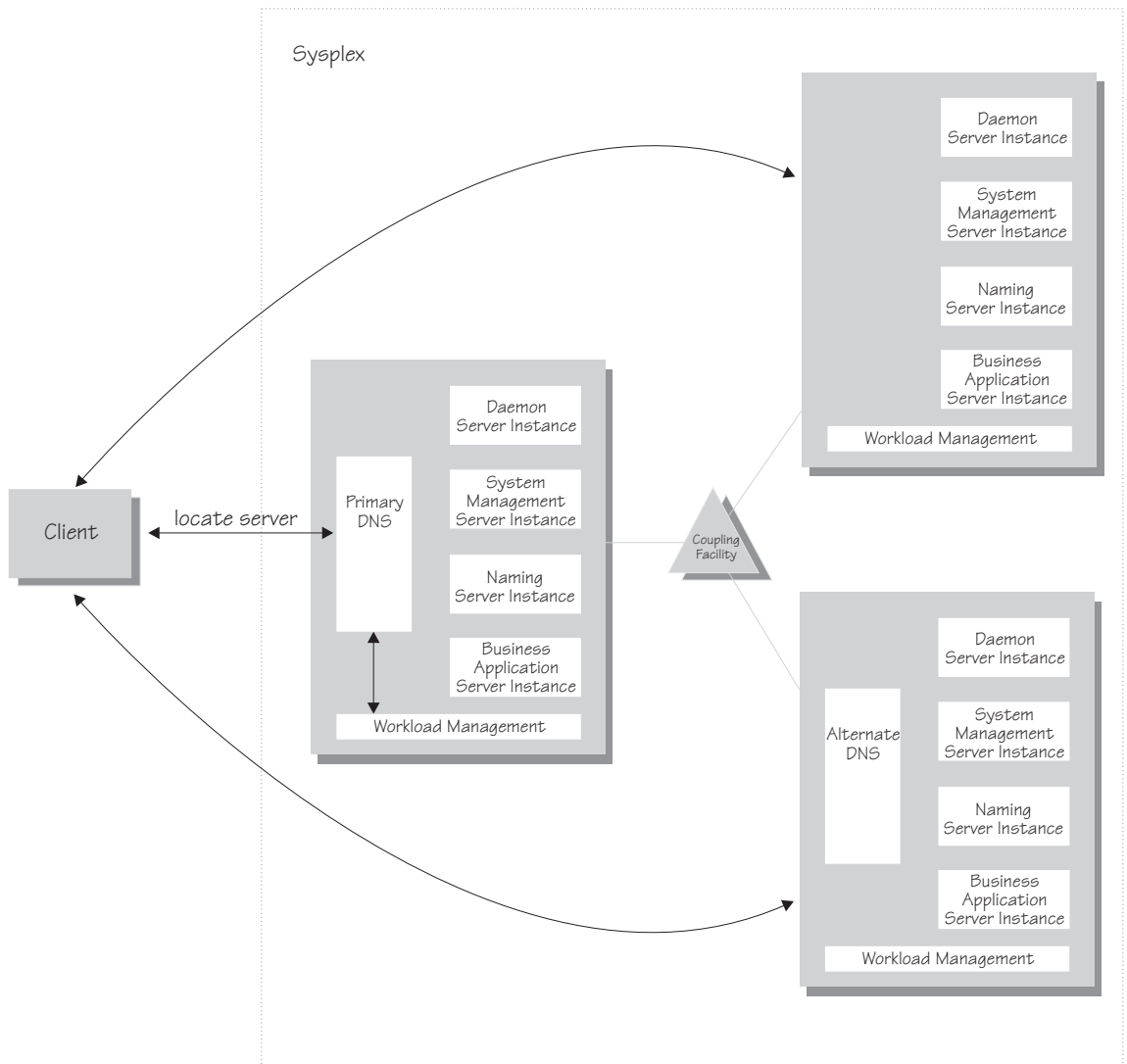
Figure 14. WebSphere for z/OS, the domain name server (DNS), and workload management

In Figure 14, each system in the sysplex has the WebSphere for z/OS run time (the Daemon, System Management, and Naming Servers), plus business application servers. The client uses the CORBA General Inter-ORB Protocol (GIOP) to make requests of WebSphere for z/OS. The Daemon acts as a location service agent. It accepts locate requests with object keys in the requests. The 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 instance in the sysplex to handle the request. The Daemon merges specific IOR

information related to the chosen server instance 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 instance 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 or OS/390 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 15 on page 267, the control region of a server instance, 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 server region in the server instance. If the work has a low priority, workload management can direct the work to a low-priority server region. The effect is to partition the work according to priority within the same server instance.
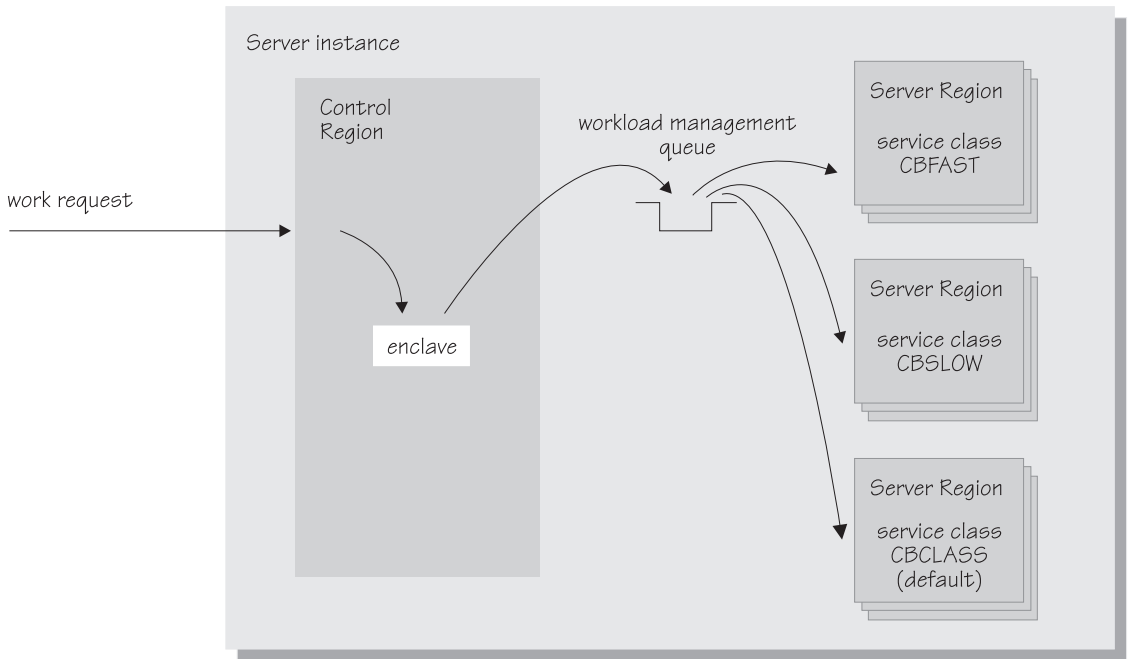
Figure 15. Use of enclaves for managing the priority of work

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 39. WLM work qualifiers and corresponding WebSphere for z/OS entities

Work qualifier abbreviation	Work qualifier	Corresponding WebSphere for z/OS entity
CN	Collection name	Server 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 control regions act as work routers, so they must have high priority. Because workload management starts and stops server regions dynamically, server regions also need high priority in order to be initialized quickly. Once initialized, however, server regions run work according to the priority of the client enclave, so the server region priority you assign has no significance after initialization.

In summary, use the following table to set the performance goals for each class:

Table 40. Workload management rules

If you are classifying...	... assign it to:	Reason
The 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 **control** region	SYSSTC	A control region must route work quickly.
An WebSphere for z/OS run-time server **server** region	SYSSTC	A server region must initialize quickly, but, once initialized, it runs work according to the priority of the client enclave.
Your business application **control** region	A class having at least as much importance as that of the work that flows through it.	A control region must route work quickly, but you must balance the priority of your business application server with other work in the system.
Your business application **server** region	SYSSTC	A server region 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

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 BBOASR1 that requires fast response times. Also, you want to give work that runs under your boss' user ID (DBOOZ) slower response times. Finally, all remaining work requests should run under the default service class, CBCLASS.

Table 41. Classification rules example

Type column	Name column	Service column	Goal
CN	BBOASR1	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  . . . . . . . . . CB Series 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       BBOASR1  ___                 CBFAST    _____
  ____  1  UI       DBOOZ    ___                 CBSLOW    _____
  ***************************** BOTTOM OF DATA *****************************
```

In this example, all work for BBOASR1, 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
sysplex and including the work for WebSphere for z/OS run-time servers,
gets classified as CBCLASS.

Setting up the CICS-EXCI Procedural Application Adapter for CORBA applications

The CICS Procedural Application Adapter uses the CICS-EXCI interface. This
section covers steps you should take to set up the CICS-EXCI interface for
WebSphere for z/OS.

Steps for setting up the CICS-EXCI Procedural Application Adapter

Before you begin: You must have Java for z/OS or OS/390, WebSphere for z/OS, and the CICS subsystem to which it connects on the same z/OS or OS/390 image. This limitation exists because the EXCI interface, which allows RRS to coordinate the transaction, requires that the client and CICS server reside on the same system.

Follow these steps to set up the CICS-EXCI Procedural Application Adapter:

1. Specify RRMS=YES in the CICS *hlq*.SYSIN(*member*) data set to make CICS participate in the RRS context.

   _____

2. Include CICS in the same restart group as WebSphere for z/OS and DB2. See "Setting up automatic restart management" on page 214.

   _____

3. Set up the CICS region for your application. We provide a sample job, BBOADEFS, that sets up the CICS region for an application (in our case, the BCASHAC program).

   _____

4. Follow these requirements and guidelines for business application servers that use the CICS Procedural Application Adapter:

   - You must define a *specific* type connection in the CICS resource definition with a NETNAME that is the same as the WebSphere for z/OS server name.
   - When defining the logical resource manager for a server, you must choose CICS_EXCI_PAA as the logical resource manager subsystem type and identify the following for the logical resource manager instance connection data:

   CICS applid
   > The CICS application ID.

   Details about coding WebSphere for z/OS applications that use CICS, including the setup of the server, are in *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*

   _____

You are done with configuration steps. You will need to test the CICS-EXCI set up with your buiness application testing.

Guidelines for the IMS-OTMA Procedural Application Adapter (CORBA applications)

The IMS-OTMA Procedural Application Adapter uses the Open Transaction Manager Access (OTMA) protocol for IMS. As such, there are guidelines and requirements you must follow:

- IMS, Java for z/OS or OS/390, and WebSphere for z/OS must be on the same system in the sysplex. This limitation exists because the OTMA interface, which allows RRS to coordinate transactions, requires that the client (WebSphere for z/OS) and IMS server reside on the same system.

- Include IMS in the same restart group as WebSphere for z/OS and DB2. See "Setting up automatic restart management" on page 214.

- A WebSphere for z/OS application server instance acts as an IMS-OTMA client, which means it must be in the same XCF group to communicate with the IMS-OTMA.

  The IMS-OTMA XCF group name is one of the parameters required when you define an IMS-OTMA PAA Logical Resource Mapping (LRM) through the Administration application. The other is the XCF partner name that identifies the specific IMS with which the server communicates. The XCF partner name is the name specified by the OTMANM parameter in the IMS DFSPBxxx proclib member used for initialization. If no OTMANM parameter is defined, then the name specified by the APPLID1 parameter in the IMS DFSPBxxx member will be used as the default XCF partner name.

- You must give the control region user ID in the application server instance READ authority to the IMSXCF.OTMACI resource in the RACF FACILITY class. For details, see *IMS/ESA Open Transaction Manager Access Guide*, SC26-8743.

- Set the IMS parallel scheduling limit to 0 (any number of transactions can be scheduled).

- A transaction in a WebSphere for z/OS application may result in several transactions in IMS. For instance, within a transactional scope in a WebSphere for z/OS application, a program may perform a findByPrimaryKey, three setters, and three getters, resulting in three separate IMS transactions. This multiplying effect on transactions affects the number of message processing regions IMS must have. You must specify the number of message processing regions in the DFSMPR job to equal the number of transactions that could result from a WebSphere for z/OS transaction.

  Example: If you have a WebSphere for z/OS transaction that could generate 5 IMS transactions, set the number of message processing regions to 5.

  If additional WebSphere for z/OS applications generate additional IMS transactions on the same database, set the number of message processing regions according to the maximum number of transactions that could be generated from all applications.

- You may use only SendReceive requests when communicating with a target transaction program in IMS. Requests to do Send-only or Receive-only processing with an IMS transaction program are not supported.
- For more information about OTMA, see *IMS/ESA Open Transaction Manager Access Guide*, SC26-8743.
- The following are planning requirements and guidelines for business application servers that use the IMS-OTMA Procedural Application Adapter. Details about coding WebSphere for z/OS applications that use IMS, including the setup of the server, are in *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836:
  - When defining the logical resource manager for a server, you must choose IMS_OTMA_PAA as the logical resource manager subsystem type and identify the following for the logical resource manager instance connection data:

    XCF group name
    Fill in the name specified on the GRNAME parameter in the DFSPBxxx proclib member used for IMS initialization.

    XCF partner name
    Fill in the name specified on the OTMANM parameter in the DFSPBxxx proclib member used for IMS initialization. Otherwise, use the name specified by the APPLID1 parameter in the DFSPBxxx member, which is the default XCF partner name if no OTMANM parameter is defined.

    number of sessions
    Specify 1.

    TPIPE prefix
    Specify a prefix, which must be four characters or less, for the system to use for all transaction pipes required for this LRM. When creating a transaction pipe for this LRM, the system generates a unique transaction pipe name by using this prefix and appending four characters of session-related information.

    Rule: You cannot have more than one logical resource manager instance with the same XCF group name configured to a given server instance.

    For a given server instance, WebSphere for z/OS connects once, and only once, to a single IMS member within an IMS XCF group specified by a logical resource manager instance. If you have configured the server instance with another logical resource manager instance that has the same XCF group name, but a different IMS member name, TPIPE name, or number of sessions, initialization of that logical resource manager instance will fail when it attempts to connect to the same IMS XCF

group. This is because the server instance will already be a member of the IMS group as a result of the first connection.

– Make sure you specify enough members in the XCF data set definitions. You must specify an XCF data set member for each server using the IMS-OTMA Procedural Application Adapter.

Setting up the IMS-APPC Procedural Application Adapter

The IMS-APPC Procedural Application Adapter allows WebSphere for z/OS to communicate through APPC/MVS with IMS on a remote or local system. APPC/MVS provides a programming interface (LU 6.2 architecture) that WebSphere for z/OS exploits to communicate on a peer-to-peer basis with application programs. Through settings in WebSphere for z/OS, you can determine whether an APPC conversation is protected. Three possibilities exist:

- You can require protected conversations. By using protected conversations (syncpt specified on the logical resource manager instance), APPC/MVS becomes a communications resource manager and has expressed interest in the outcome of a WebSphere for z/OS transaction, driving the IMS transaction running on another system under the same transactional scope as WebSphere for z/OS. All of the processing done on behalf of a distributed application is treated as an atomic, or single, operation. In other words, APPC/MVS, WebSphere for z/OS, and IMS coordinate their processing so that all application updates are either made (committed) or not made (rolled back). This coordination is most beneficial for applications that have a critical dependency on data integrity.

  This transaction management happens automatically when you create a server with the IMS-APPC Procedural Application Adapter and protected conversations. We say that the conversations have syncpoint capabilities; that is, data in your application is synchronized with data in the IMS database.

- You can allow unprotected conversations. If your application does not require the use of protected conversations, you may create a server with the IMS-APPC Procedural Application Adapter without syncpoint capabilities (none specified on the logical resource manager instance). With no syncpoint capabilities, there is no guarantee that data in the IMS database is synchronized with data in your client application—your client application becomes responsible to recheck the data in IMS if it makes an update. Although there is no synchronization of data, there are benefits:

  – Your application no longer pays the performance cost associated with distributed transactions.

  – Fewer IMS message processing regions (MPRs) become busy. In a transaction, a simple read/write operation requires two message processing regions to remain busy until a transaction commitment

occurs. If you use no syncpoint capabilities, one message processing region can serve a data request, then immediately become available for another request.

- You can allow WebSphere for z/OS to determine whether an APPC conversation is protected when the conversation is allocated (`autotran` specified on the logical resource manager instance). WebSphere for z/OS makes the determination based on the container transaction policy and the type of transaction the current execution thread is running under.

  Container transaction policies control the type of transaction under which an execution thread runs. Containers can require global transactions (TX_REQUIRED) or allow variations of local transactions started by your application (these variations are collectively called HYBRID_GLOBAL policies). If `autotran` is the setting when WebSphere for z/OS is about to allocate an APPC conversation, WebSphere for z/OS will:

  – Allocate a protected conversation if the execution thread is running under a global transaction (the container policy requires a global transaction)
  – Allocate an unprotected conversation if the execution thread is running under a local transaction (the container policy allows a local transaction)

  Before you set up the APPC connection, you must determine the transactional characteristics of your application and know the appropriate container transaction policy. Details on transaction policies are in *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836. You need to know these things because this determines whether you must define the APPC connection with syncpoint capabilities. If you plan to use the `syncpt` or `autotran` settings on the logical resource manager instance, you should define syncpoint capabilities for the APPC connection. If you plan to use the `none` setting on the logical resource manager instance, you do not need to define syncpoint capabilities for the APPC connection.

Setting up a server that uses IMS-APPC Procedural Application Adapter

To set up a server with the IMS-APPC Procedural Application Adapter, you must coordinate the configuration on both sides of the communication path, then define the connection to your WebSphere for z/OS server through the Administration application:

- On the WebSphere for z/OS side (which we designate as the local system), you must coordinate the configuration for VTAM and APPC.
- On the IMS side (which we designate as the partner system), you must coordinate the configuration for VTAM, APPC, and IMS.
- Finally, you must define the connection for your WebSphere for z/OS server through the Administration application.

The following table shows the subtasks and associated procedures for setting up a server that uses the IMS-APPC Procedural Application Adapter:

Subtask	Associated procedure (See . . .)
Setting up the WebSphere for z/OS (local) side	"Steps for setting up the WebSphere for z/OS (local) side"
Setting up the IMS (partner) side	"Steps for setting up the IMS (partner) side" on page 278
Defining the connection to your WebSphere for z/OS server	"Step for defining the connection to your WebSphere for z/OS server" on page 281

For more information on configuring APPC/MVS, see *z/OS MVS Planning: APPC/MVS Management*, SA22-7599.

Steps for setting up the WebSphere for z/OS (local) side

Before you begin: You must have VTAM and APPC installed on your WebSphere for z/OS system. Also, you must decide whether your application requires syncpoint capabilities.

Perform the following steps to set up the WebSphere for z/OS (local) side:

1. Define a logical unit (LU) to VTAM in its APPL definitions.

   - To enable syncpoint capabilities for the LU, you must code the VTAM APPL definition with SYNCLVL=SYNCPT and ATNLOSS=ALL. Also, you must configure RRS and make it active.
   - To run without syncpoint capabilities, you do not need to specify either the SYNCLVL or ATNLOSS keywords.

   Recommendation: Create an LU specifically for WebSphere for z/OS because you can manage the LU more easily. You need define only one LU through which all WebSphere for z/OS-initiated conversations will pass. For sample LU definitions, see the sample in SYS1.SAMPLIB(ATBAPPL).

   _____

2. Create at least one APPC TP Profile data set. See SYS1.SAMPLIB(ATBTPVSM) for a sample job that creates an APPC TP Profile data set.

   _____

3. Define an APPC LU that matches the LU you defined for VTAM. On the TPDATA keyword, specify the APPC TP Profile data set you created in step 2.

Tip: Since this LU will likely support outbound conversations only, you can avoid starting up a transaction scheduler and increasing resource overhead by specifying NOSCHED on the LU.

The LU names are defined in the APPCPMxx member in SYS1.PARMLIB. For a sample member, see SYS1.SAMPLIB(APPCPMxx).

_____

4. To implement syncpoint capabilities, define the ATBAPPC.LU.LOGNAMES log stream to the system logger.

   Note: If WebSphere for z/OS and IMS are on different systems in the same sysplex, you must use the coupling facility for the log stream. APPC/MVS supports a DASD-only log stream in a single system environment only.

   _____

5. Ensure you have VTAM connectivity to the IMS system. You can use VTAM Subarea, VTAM APPN, or SNA over TCP/IP network configurations.

   _____

6. Enable the VTAM APPL into the VTAM configuration.

   _____

7. Start APPC with the new WebSphere for z/OS LU defined or dynamically activate the new WebSphere for z/OS LU into the APPC configuration. Issue the following command to verify that the local LU is active. If you want syncpoint capabilities, check that Protected=YES:

   `DISPLAY APPC,LU,ALL`

   _____

You know you are done when you see the local LU active and, if you want syncpoint capabilities, that Protected=YES.

Steps for setting up the IMS (partner) side
Before you begin: You must have VTAM and APPC installed on your IMS system. Also, you must decide whether your application requires syncpoint capabilities.

Perform the following steps to set up the IMS (partner) side:

1. Define a logical unit (LU) to VTAM that is associated with IMS. This is the LU with which WebSphere for z/OS will allocate a conversation to establish communications with IMS.

- To enable syncpoint capabilities for the LU, you must code the VTAM APPL definition with SYNCLVL=SYNCPT and ATNLOSS=ALL. Also, you must configure RRS and make it active. For sample LU definitions, see SYS1.SAMPLIB(ATBAPPL).
- To run without syncpoint capabilities, you do not need to specify either the SYNCLVL or ATNLOSS keywords.

Rule: This partner LU must be able to accept a user ID without a password when communication is initiated (WebSphere for z/OS already verifies the password). You can set this up through the VTAM APPL definition, in which you specify the parameter SECACPT=ALREADYV. An alternative is to set up a RACF APPCLU profile, in which you specify CONVSEC(ALREADYV). Details on APPC security are in the chapter on security in *z/OS MVS Planning: APPC/MVS Management*, SA22-7599.

_____

2. Create at least one APPC TP Profile data set. See SYS1.SAMPLIB(ATBTPVSM) for a sample job that creates an APPC TP Profile data set.

_____

3. Define an APPC LU that matches the partner LU you defined in VTAM. On the TPDATA keyword, specify the APPC TP Profile data set you created in step 2. Specify the SCHED keyword with the value of the IMS System Identifier on the LU definition.

   The LU names are defined in the APPCPMxx member in SYS1.PARMLIB. For a sample member, see SYS1.SAMPLIB(APPCPMxx).

_____

4. To implement syncpoint capabilities, make sure you have a log stream defined for APPC on the IMS side:
   - If IMS is running on the same system as WebSphere for z/OS, APPC needs a DASD-only or coupling facility log stream called ATBAPPC.LU.LOGNAMES.
   - If IMS is running on a different system in the sysplex than WebSphere for z/OS, APPC needs a log stream called ATBAPPC.LU.LOGNAMES to be defined to use the coupling facility. That is because APPC/MVS supports a DASD-only log stream in a single system environment only.
   - If IMS is running on a remote system (not on the same system or sysplex as WebSphere for z/OS), it needs a log stream called ATBAPPC.LU.LOGNAMES on that remote system. The log stream can use either a DASD-only or coupling facility configuration.

_____

5. Set the IMS parallel scheduling limit to 0 (any number of transactions can be scheduled).

_____

6. Size the number of message processing regions IMS must have. The sizing depends on whether you use syncpoint capabilities:

   • Using syncpoint capabilities. A transaction in a WebSphere for z/OS application may result in several transactions in IMS. For instance, within a transactional scope in a WebSphere for z/OS application, a program may perform a findByPrimaryKey, three setters, and three getters, resulting in three separate IMS transactions. This multiplying effect on transactions affects the number of message processing regions IMS must have. You must specify the number of message processing regions in the DFSMPR job to equal the number of transactions that could result from a WebSphere for z/OS transaction.

   Example: Iif you have a WebSphere for z/OS transaction that could generate 5 IMS transactions, set the number of message processing regions to 5.

   • Not using syncpoint capabilities. Specify the number of message processing regions according to the number of simultaneous operations you expect IMS to process.

   If additional WebSphere for z/OS applications generate additional IMS transactions on the same database, set the number of message processing regions according to the maximum number of transactions that could be generated from all applications.

_____

7. Enable the VTAM APPL into the VTAM configuration.

_____

8. Start APPC with the new IMS LU defined or activate the new IMS LU dynamically into the APPC configuration.

_____

9. To enable the APPC-IMS LU, issue the following IMS command from the MVS or IMS console:
   ```
   /START APPC
   ```

_____

10. Issue the following command to verify that the Local LU is active:
    ```
    DISPLAY APPC,LU,ALL
    ```

_____

You know you are done when APPC starts successfully. If you want syncpoint capabilities, check that Protected=YES.

Step for defining the connection to your WebSphere for z/OS server
Before you begin: You must have WebSphere for z/OS installed, including the Administration application.

Perform the following step to define the connection to your WebSphere for z/OS server:

⇔ Use the Administration application to define the logical resource manager (LRM) for that server. Choose IMS_APPC_PAA as the LRM subsystem type and identify the following for the logical resource manager instance connection data:

Local LU name
Fill in the logical unit (LU) name associated with WebSphere for z/OS. This local LU name is defined in an LUADD statement in the APPCPMxx parmlib member for the system on which WebSphere for z/OS runs.

Look for the LUADD statement for the LU associated with WebSphere for z/OS. Use the value specified on the ACBNAME parameter as the **local** LU name.

Rule: Use only the value specified on the ACBNAME parameter, which is the network LU name. If you specify a network-qualified (or fully qualified) name for the local LU, you will receive error message BBOU0106E, which indicates that the local LU name is not valid.

Partner LU name
Fill in the name of the LU with which the WebSphere for z/OS server will initiate an APPC conversation. This partner LU is defined in an LUADD statement in the APPCPMxx parmlib member for the system on which IMS runs. The IMS subsystem may be, but does not have to be, on a system other than the one on which the WebSphere for z/OS server runs.

Look for the LUADD statement for the LU associated with IMS (an LU associated with IMS has the IMS subsystem name specified for the SCHED parameter on the LUADD statement). Use the value specified on the ACBNAME parameter as the **partner** LU name.

Tip: When you specify the partner LU name, you may use one of the following forms:
- Only the value specified on the ACBNAME parameter (in other words, the network LU name)
- A network-qualified name (in the form *networkID*.*networkLUname*)

  networkID is the value specified for the VTAM start option NETID and *networkLUname* is the value specified on the ACBNAME parameter.

- A VTAM generic resource name, if your installation is configured to use generic resources.

VTAM logmode name

Fill in the name of the VTAM logmode that designates the network properties to be associated with any APPC conversations between this local LU and its partner LU. Logmode names appear in the VTAM logon mode table, which reside in your installation's VTAMLIB data set.

APPC conversation time-out value

Specify the length of time, in minutes, for the WebSphere for z/OS server to wait for a response to the Allocate call and any subsequent calls the server issues during its conversation with IMS. Valid time-out values range from 0 through 1440, which is 24 hours.

If you specify a value that is less than the value set for the OTS_DEFAULT_TIMEOUT environment variable, the APPC conversation time-out value will have no effect. Look for the OTS_DEFAULT_TIMEOUT environment variable setting that you use for the application server's control and server regions.

APPC sync level

Fill in one of the values listed in following table. This value controls the type of APPC/MVS conversation the WebSphere for z/OS server uses to communicate with IMS. Base your choice on the transaction policies you select for containers in this server configuration, and the characteristics of the applications to be deployed in this server.

Recommendation: Use a sync level value that corresponds with the transactional context of the request that the server is currently processing. The easiest way to match the sync level and context is to select **Autotran**, which lets the system determine the matching sync level.

If this LRM is connected to:	Then specify this sync level value:	Notes
One or more containers that **all** use the TX Required transaction policy	**Syncpt** (in certain cases, **None** is also acceptable)	Because this transaction policy enforces the use of a global transaction, the most logical value for the APPC sync level is **Syncpt**. With **Syncpt**, the server allocates a protected conversation, which preserves the global transactional context for the interaction between the server and the IMS subsystem, and allows the system to recover any resources if conversation errors or failures occur. In certain cases, however, you might consider using **None** when your application's processing does not depend on the ability to recover resources at this point in its processing. With **None**, APPC/MVS, WebSphere for z/OS, and IMS do not coordinate any processing done on behalf of a distributed application; without the overhead of coordination, your application's performance improves. **Recommendations:** • Use **Syncpt** if you cannot guarantee that your server application will always run on the same z/OS or OS/390 system on which the IMS subsystem runs. • Use **None** judiciously. In this case, resources that the application uses might be in inconsistent states if conversation errors or failures occur.
One or more containers that use a transaction policy other than TX Required	**Autotran**	Use **Autotran** with these policies, so the system can determine which conversation type, Syncpt or None, is appropriate for the transactional context associated with the current thread of execution. In other words, if the current thread has a local transactional context, the server uses a sync level of None; for a global transactional context, the server uses Syncpt.

If you need additional information about the transaction policies for containers, see *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836.

___

You know you are done when you save the logical resource manager and receive the message that the system added the logical resource manager.

Guideline for recovery of IMS transactions

Consider automatic restart management and what would happen if a system fails and WebSphere for z/OS is restored on a second system in the sysplex. As long as RRS is not running on the failed system, transactions could complete on the restored system if you had an LU with the same name and attributes as that on the failed system. However, you cannot set up two LUs in the sysplex with the same name and attributes in anticipation of a failure, because VTAM will not allow it (it would be like having the same telephone number in two places). Rather, you could manually reactivate the WebSphere for z/OS LU on the restored system after a failure on the first system (similar to moving a telephone number to a new location).

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 environment 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 or OS/390 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 weeks.

Overview of code upgrading methods

There are several methods used to install a new functional level of WebSphere for z/OS. The methods are classified by the kind of change that is made to WebSphere for z/OS and the way you start the Daemon server. The methods are:

- Cold start
- Warm start
- Hot start
- Quick start

Before we describe the various upgrading methods, we need to discuss an important point about your sysplex HFS structure.

Overview of creating the proper HFS structure for upgrades

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.

Obviously, if you have WebSphere for z/OS running in a monoplex, or on a single system in a sysplex, and you need to install a new code level that requires shutting down WebSphere for z/OS, you have no choice except to disrupt service to your clients. Also, if you have the luxury of being able to disrupt service to your clients, even when you have WebSphere for z/OS running in a sysplex, then interruption of service to your clients is not a problem. But, if you cannot disrupt service while upgrading code levels, you must set up your HFSes in a sysplex as we describe in this section.

To use the rolling upgrade method, 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 what we call 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 16.

Note: For information about an alternate HFS structure that accomplishes the same objective, see "Appendix C. Using an alternate HFS structure for product upgrades" on page 345.
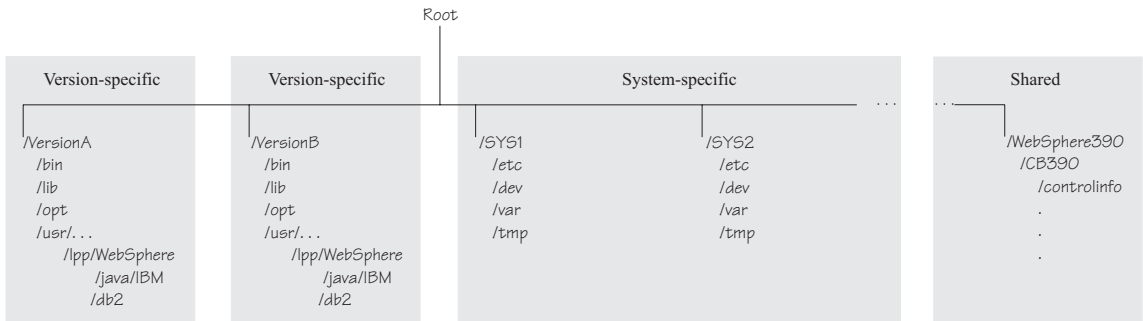
Figure 16. HFS structure for the rolling upgrade method

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.ptf10.jdbc.hfs at /VersionA/usr/lpp/db2

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 17.
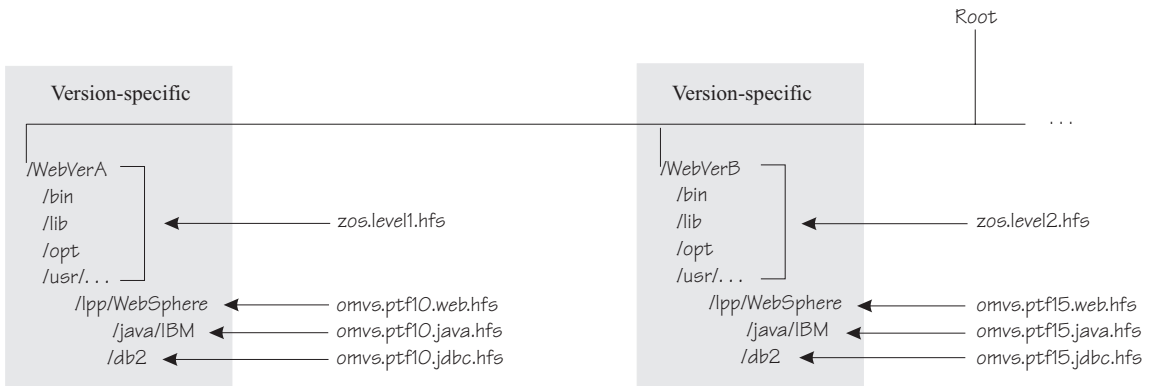
Figure 17. Mount point configuration for WebSphere for z/OS

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, in warm start mode, which enables the new function. More on this in "Warm start" on page 290.

Cold start

Cold start is a method for:

- Installing WebSphere for z/OS initially. For your first installation and customization of WebSphere for z/OS, we provide a default system configuration. Those procedures are described in "Chapter 3. Installing and customizing your first run time" on page 55.
- Recreating WebSphere for z/OS when, for some reason, your WebSphere for z/OS databases have been corrupted and cannot be recovered.

Release changes after WebSphere for z/OS V4.0 do not require the cold start method.

Note: The prepare for cold start function described in this topic is not suitable for backing up your system. If you want to back up all persistent data for your WebSphere for z/OS system, you must back up:

- The system management database
- The LDAP database tables containing the naming space and the interface repository
- The directory where WebSphere for z/OS run time information is written (the value of the CBCONFIG environment variable. The default is /WebSphere390/CB390.)
- WebSphere for z/OS PROCLIBs

Once you have completed your initial installation and customization of
WebSphere for z/OS, there are two main tasks you must do to cold start
WebSphere for z/OS:

1. Prepare the system for cold start, which off-loads the existing
   configuration to off-load files.
2. Shut down WebSphere for z/OS (or the entire host cluster, if WebSphere
   for z/OS is running in a sysplex), install the functional changes, and
   restart the Daemon with the cold start option.

The cold start process

The following describes how the cold start process works.

Stage	Description
Prepare for cold start.	From the Administration application, a system programmer executes prepare for cold start. WebSphere for z/OS stops all application servers. The system stores configuration data in XML format in the HFS. The system also stores environment variable data for the servers into files in the HFS.
Shut down WebSphere for z/OS (or the entire host cluster, if running in a sysplex).	The entire WebSphere for z/OS or host cluster must be shut down.
Make base component changes.	Install the new functional level of WebSphere for z/OS.
Run the customization dialog.	As you would for a first-time installation and customization, run the customization dialog to create the necessary jobs for the cold start. When finished, you have a customized set of instructions to follow.
Follow the customized instructions, but stop after job BBOMCFG.	You need to stop at this point and drop the LDAP database.

Stage	Description
Continue with the customized instructions.	Continue with the customized instructions where you left off. The instructions have you: • Recreate the system management and LDAP databases by running the following jobs: BBOMCRDB, BBOLDTBD, and BBOLDTBC. • Run the first bootstrap by starting the Daemon with the cold start option (-ORBCBI COLD). The system reads the XML off-load file and the environment files to restore the configuration and environment variable data. • Run the Naming and Interface Repository bootstrap programs. • Run the second phase of the WebSphere for z/OS bootstrap.
Start one server instance for each application server	Start a server instance for each application server one after the other. Wait for the naming registration to complete for each server instance before starting the next server instance.
Rerun any application initialization routines.	This stage creates application naming contexts or persistent data.
Rerun any Interface Repository routines.	This stage recreates Interface Repository entries for applications.

Backout plan for cold start

If you need to restore the previous WebSphere for z/OS functional level, use the same cold start process, but restore the previous functional level of WebSphere for z/OS after you prepare for cold start and shut down WebSphere for z/OS. Any new configuration data generated during preparation for cold start will be ignored by the previous level.

Warm start

Warm start is a method to move from one functional level of WebSphere for z/OS to another that requires changes to persistent data (for example, the system management database). If performed in a sysplex with the proper HFS structure, the method does not disrupt WebSphere for z/OS service to clients.

To be non-disruptive for your running applications, the warm start method requires that you have WebSphere for z/OS set up in a sysplex as explained in "Enabling WebSphere for z/OS on a sysplex" on page 219. In particular, you need to have a shared HFS with two mount points, each holding a level of WebSphere for z/OS. The method allows you to bring down one clustered host instance at a time, then switch the HFS for that instance to the new level.

Because other clustered host instances are operating when one is down, clients still get service from WebSphere for z/OS.

The warm start process
The following describes the warm start process in general. For detailed procedures about release migrations that use the warm start method, see the procedures that follow this topic. For detailed procedures about service level changes that use the warm start method, see the documentation accompanying the service level changes.

Stage	Description
Code installation	Install the new WebSphere for z/OS code and mount it in an HFS at a mount point that is not currently in use.
System backup	During this stage, you back up the system management database, the LDAP database, files in the HFS (default /WebSphere390/CB390) containing run-time information, WebSphere for z/OS PROCLIBs, and WebSphere for z/OS LOADLIBs.
Cancellation of WebSphere for z/OS operations	Stop WebSphere for z/OS. If you have WebSphere for z/OS enabled on your sysplex, stop WebSphere for z/OS (the clustered host instance) on one system. Other systems continue with WebSphere for z/OS operations.
New level of code enabled	On the system that you stopped WebSphere for z/OS: • Perform the necessary migration actions, such as running jobs to alter databases • Update the run-time start procedures to point to the data sets with the new code • Load new run-time modules into LPA and change the link list • Switch the HFS the system references to the one with the new code

Stage	Description
WebSphere for z/OS Daemon and application servers restarted	Restart the WebSphere for z/OS Daemon and application servers.
	If you have WebSphere for z/OS enabled on a sysplex, do the following for each clustered host instance, one at a time:
	• Stop the Daemon
	• Update the run-time start procedures to point to the HFS with the new code
	• Load new run-time modules into LPA
	• Switch the HFS the system references to the one with the new code
	• Restart the WebSphere for z/OS Daemon and application servers
WebSphere for z/OS restarted in warm start mode	Restart the Daemon with the warm start option. Restart your application server instances with the warm start option.
	If you have WebSphere for z/OS enabled on a sysplex, restart the Daemon and application server instances in warm start mode for each system in the sysplex, one at a time.

Backout plan for warm start
If you need to restore your previous WebSphere for z/OS system:

- Stop WebSphere for z/OS. On a sysplex, you must stop the entire host cluster.
- Change your server instance start procedures to reference the HFS with the previous version's code.
- Load the previous version's run-time modules into LPA and restore the old link list.
- Switch the HFS that your system references with the SETOMVS command. Use the command to change the $VERSION symbolic.

  Example: To switch back to the previous version's HFS (VersionA) that a system references, issue:

  setomvs version=VersionA

- Restore the databases, environment variable and configuration files, PROCLIBs, and LOADLIBs that you backed up.
- Restore the previous levels of the Administration application and Operations application.
- Start the Daemon. On a sysplex, start the Daemon for each clustered host instance.

- Start your application servers. On a host cluster, start all your application server instances.

Hot start

Hot start is a method that allows you to change a functional level of WebSphere for z/OS that does not require changes to the WebSphere for z/OS databases. If performed in a sysplex with the proper HFS structure, the method does not disrupt WebSphere for z/OS service to clients. You will likely use this method for most service level upgrades for WebSphere for z/OS.

To be non-disruptive for your running applications, the hot start method requires that you have WebSphere for z/OS set up in a sysplex as explained in "Enabling WebSphere for z/OS on a sysplex" on page 219. In particular, you need to have a shared HFS with two mount points, each holding a service level of WebSphere for z/OS. The method allows you to bring down one clustered host instance at a time, then switch the HFS for that instance to the new service level. Because other clustered host instances are operating when one is down, clients still get service from WebSphere for z/OS.

The hot start process
The following describes how the hot start process works.

Stage	Description
Code installation	Install the new WebSphere for z/OS code and mount it in an HFS at a mount point that is not currently in use.
System backup	During this stage, you back up the system management database, the LDAP database, files in the HFS (usually /WebSphere390/CB390) containing run-time information, WebSphere for z/OS PROCLIBs, and WebSphere for z/OS LOADLIBs.
Cancellation of WebSphere for z/OS operations	Stop WebSphere for z/OS. If you have WebSphere for z/OS enabled on your sysplex, stop WebSphere for z/OS (the clustered host instance) on one system. Other systems continue with WebSphere for z/OS operations.
New level of code enabled	On the system that you stopped WebSphere for z/OS: • Update the run-time start procedures to point to the data sets with the new code • Load new run-time modules into LPA and update the link list • Switch the HFS the system references to the one with the new code

Stage	Description
WebSphere for z/OS restarted	Restart the Daemon. Restart your application server instances.
	If you have WebSphere for z/OS enabled on a sysplex, do the following to each system, one at a time:
	• Update the run-time start procedures to point to the data sets with the new code
	• Load new run-time modules into LPA and update the link list
	• Switch the HFS the system references to the one with the new code
	• Restart the Daemon and application server instances on each clustered host instance, one at a time.

Backout plan for hot start

If you need to restore your previous WebSphere for z/OS system:

• Stop WebSphere for z/OS. In a sysplex, stop one clustered host instance while the other instances remain running.

• Change your server instance start procedures to reference the HFS with the previous version's code.

• Load the previous version's run-time modules into LPA and restore the old link list.

• Switch the HFS that your system references with the SETOMVS command. Use the command to change the $VERSION symbolic.

   Example: To switch back to the previous version's HFS (VersionA) that a system references, issue:

   setomvs version=VersionA

• Start the Daemon.

• Start your application servers.

• In a sysplex, repeat the above for each clustered host instance, one at a time.

Quick start

Quick start is a method you use when no changes to the WebSphere for z/OS databases are required, there are code changes only to data sets, and only single servers need to be restarted selectively. In this case, a single server instance on a clustered host instance is brought down, the code installed, then the server instance is restarted. Because other server instances are running, the server is still available for client requests.

Service updates are likely candidates for quick start. You will always receive specific instructions on how to do the quick start for a given service update.

The quick start process
The following describes how the quick start process works.

Stage	Description
Stop the server instance on the first clustered host instance.	An operator or system programmer stops one server instance only.
Make the necessary code updates.	At this stage, install the new WebSphere for z/OS code.
Restart the server instance.	The server instance restarts and can now serve clients.
Repeat this process for each additional server instance, one at a time.	

Procedures for upgrading WebSphere for z/OS code

This topic provides procedures for you to follow for upgrading WebSphere for z/OS code.

Base your choice of which procedure to use on the following table:

If you are upgrading WebSphere for z/OS and the upgrade requires . . .	Then follow . . .	Notes
A warm start (either on a monoplex or a sysplex)	"Steps for performing a warm start from WebSphere for z/OS V4.0 to V4.0.1" on page 296	The procedure uses a migration path through the customization dialog. If you did not use the customization dialog to customize WebSphere for z/OS V4.0, and prefer to use the sample jobs instead, see "Appendix B. Migrating from WebSphere for z/OS V4.0 to V4.0.1 without the customization dialog" on page 337.
A hot start, and you can interrupt service to clients (either on a monoplex or a sysplex)	"Steps for performing a hot start with a system or sysplex-wide restart" on page 299	
A hot start on a sysplex, but you cannot interrupt service to clients	"Steps for performing a rolling hot start" on page 301	A rolling hot start requires the dual HFS structure described in "Overview of creating the proper HFS structure for upgrades" on page 285.

You can now perform the steps for the decision you have made.

Steps for performing a warm start from WebSphere for z/OS V4.0 to V4.0.1

Before you begin:

Requirement: Your V4.0 system must have the proper level of service installed. See "Overall migration tasks to go from WebSphere for z/OS V4.0 to WebSphere for z/OS V4.0.1" in *WebSphere Application Server V4.0.1 for z/OS and OS/390: Migration*, GA22-7860 and consult the PSP bucket for the latest service information.

Rules: Regarding your display:
- Your logon display must support a minimum of 32 rows by 80 columns (32 x 80) in order for the ISPF customization dialog to run.
- You cannot use the split screen function in ISPF. Using split screen will obscure lines at the bottom of the dialog panels.

Perform the following steps to do the warm start.

1. Use SMP/E to install the new WebSphere for z/OS code. Be sure to install the new WebSphere for z/OS code into a separate set of MVS and HFS data sets.

   _____

2. Back up your current system. This includes:
   - The system management database
   - The LDAP database tables containing the naming space and the interface repository
   - Files in the HFS containing WebSphere for z/OS run-time information (usually mounted at /WebSphere390/CB390).
   - WebSphere for z/OS PROCLIBs
   - WebSphere for z/OS LOADLIBs

   For more information, see "Guidelines for backup of the WebSphere for z/OS system" on page 207.

   _____

3. 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.

hlq(*value***)**

Specifies the data set qualifier(s) for the WebSphere for z/OS product data sets. The default value is BBO.

appl(*value***)**

Specifies the ISPF application name. The default value is BBO.

lang(*value***)**

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

Example:

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

Result: You see the splash screen:

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


    WebSphere Application Server V4.0.1 for z/OS and OS/390
    Licensed Material - Property of IBM

    5655-F31 (C) Copyright IBM Corp. 2001
    All Rights Reserved.
    U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
    Disclosure restricted by GSA-ADP schedule contract with IBM Corp.

    Status  = H28W401

    Version = 4.01.001


                    Press ENTER to continue.
```

_____

4. Press Enter.

Result: You see the following panel:

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

   Use this dialog to customize WebSphere for z/OS for the first time
   or to migrate releases. Specify an option and press ENTER.


   1  New customization. If you are customizing WebSphere for
      z/OS for the first time, use this option.

   2  Migration with saved variables. If you have previously saved the
      customization variables using the dialog, use this option to
      migrate from WebSphere for z/OS V4.0 to V4.0.1.

   3  Migration without saved variables. If you have never run the
      customization dialog, or have not previously saved the
      customization variables, use this option to migrate from
      WebSphere for z/OS V4.0 to V4.0.1.
```

5. If you used the customization dialog in V4.0 and saved your
   customization variables, choose option 2.

   Result: You see the main dialog panel:

```
-----------------        WebSphere for z/OS Customization      ------------------
Option ===>                                                    Appl: BBO
   Migration with Saved Variables

   Use these panels to define WebSphere for z/OS variables and generate
   customization jobs for your migration.  Specify the HLQ for
   WebSphere product data sets, an option, and press ENTER.

   HLQ for WebSphere product data sets: BBO

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

   2  Define variables. Define your customization variables for
      migration.

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

   4  View the generated migration instructions.



   Options for WebSphere 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.
```

6. If you did not use the customization dialog in V4.0 or did not save your
   customization variables, choose option 3.

   Result: You see the main dialog panel:

```
---------------        WebSphere for z/OS Customization        ------------------
Option ===>                                                     Appl: BBO
 Migration without saved variables.
   Use these panels to define all WebSphere for z/OS customization variables
   and to generate migration jobs for your installation.  Specify the HLQ for
   WebSphere product data sets, an option, and press ENTER.

   HLQ for WebSphere product data sets: BBO


   1  Allocate target data sets. The data sets will contain the
      WebSphere migration 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 migration jobs and instructions.

   4  View the generated migration instructions.



   Options for WebSphere 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.
```

_____

7. Follow the customization dialog to allocate target data sets, define
   variables, generate the customization jobs, and view the migration
   instructions.

_____

8. Use the migration instructions to complete your migration to WebSphere
   for z/OS V4.0.1.

_____

You are done when you complete the migration instructions.

Steps for performing a hot start with a system or sysplex-wide restart

Before you begin: You must be prepared to stop WebSphere for z/OS. If you
have WebSphere for z/OS running in a sysplex as a host cluster, this
procedure has you shut down the entire host cluster. If you have WebSphere
for z/OS running in a sysplex as a host cluster and want to maintain service
to your clients during the hot start, see "Steps for performing a rolling hot
start" on page 301.

Perform the following steps to do the hot start.

1. Use SMP/E to install the new WebSphere for z/OS code. Be sure to install the new WebSphere for z/OS code into a separate set of MVS and HFS data sets.

   _____

2. Back up your current system. This includes:
   - The system management database
   - The LDAP database tables containing the naming space and the interface repository
   - Files in the HFS containing WebSphere for z/OS run-time information (usually mounted at /WebSphere390/CB390).
   - WebSphere for z/OS PROCLIBs
   - WebSphere for z/OS LOADLIBs

   For more information, see "Guidelines for backup of the WebSphere for z/OS system" on page 207.

   _____

3. On either the monoplex or each system in the sysplex, do the following:
   a. Stop the application servers and the WebSphere for z/OS Daemon.
   b. 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.
   c. 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.
   d. Verify that the newly-serviced HFS data sets are correctly mounted.
   e. Perform any other migration actions (such as DB2 binds) as instructed in PTF or APAR cover letters.
   f. Start the Daemon and application servers.

      _____

You are done when WebSphere for z/OS and all your application servers are running.

Steps for performing a rolling hot start

Before you begin: Read "Recommendation for the HFS structure" on page 223 to understand the HFS structure you need.

Perform the following steps to do a hot start:

1. Use SMP/E to install the new WebSphere for z/OS code. Be sure to install the new WebSphere for z/OS code into a separate set of MVS and HFS data sets.

   Example: Install a service level (PTF15) and copy the code to OMVS.PTF15.WEB.HFS.

   _____

2. Mount the new data set and the appropriate Java and JDBC data sets at the alternate mount point.

   Example:
   ```
   mount omvs.ptf15.web.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
   ```

   _____

3. Select a clustered host instance to begin the hot start process. On that clustered host instance:

   a. Stop the application servers and the WebSphere for z/OS Daemon.

   b. 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.

   c. 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.

   d. Switch the HFS that your system references with the SETOMVS command. Use the command to change the $VERSION symbolic.

      Example: Previously, the $VERSION symbolic was VersionA for all systems in the sysplex. Through the use of a built-in symbolic link, references to /usr resolved to VersionA/usr. To switch the HFS that this system references to, say, VersionB, issue:
      ```
      setomvs version=VersionB
      ```

  e. Perform any other migration actions (such as DB2 binds) as instructed in PTF or APAR cover letters.

  f. Start the Daemon and application servers.

  _____

4. Repeat step 3 on page 301 for each remaining clustered host instance, one at a time.

  _____

You are done when you have completed the hot start on each system in the sysplex.

Appendix A. Environment files

This appendix provides reference information for environment files and environment variables.

Environment files and environment variables

This section describes:

- How WebSphere for z/OS manages environment variables and environment files.
- How run-time server start procedures point to their environment files.
- Environment variables for z/OS or OS/390 clients.
- The syntax and meaning of the run-time environment variables.

How WebSphere for z/OS manages server environment variables and environment files

After the bootstrap process during installation and customization, WebSphere for z/OS manages environment data through the Administration application and writes the environmental data into the system management database. To add or change environment variable data, you must enter environment data pairs (an environment variable name and its value) on the sysplex, server, or server instance properties form. When you activate a conversation or prepare for a cold start, the environment variable data is written to HFS files. WebSphere for z/OS determines which values are the most specific for an environment file. For instance, a setting for a server instance takes precedence over the setting for the same variable for its server, and a setting for a server takes precedence over the setting for the same variable for its sysplex.

If you modify an environment file directly and not through the Administration application, any changes are overwritten when you activate a conversation or prepare for a cold start.

When you activate a conversation or prepare for a cold start, WebSphere for z/OS writes the environment data to an HFS file for each server instance. The path and name for each environment file is:

CBCONFIG/controlinfo/envfile/*SYSPLEX*/*SRVNAME*/current.env

where

CBCONFIG
   Is a read/write directory that you specify at installation time as the directory into which WebSphere for z/OS is to write configuration data

and environment files. At installation time, we call this directory TARGETDIR. The default is `/WebSphere390/CB390`.

Rule: The System Management group (default CBCFG1) and user ID (default CBSYMSR1) must own each directory and subdirectory in CBCONFIG. If the System Management group and user ID do not own CBCONFIG, use the chown command to make them the owner of each directory and subdirectory in CBCONFIG. Thus, if you use the default CBCONFIG, you must use the chown command to give the System Management group and user ID ownership of /WebSphere390 and /WebSphere390/CB390.

Example:
```
chown -R CBSYMSR1:CBCFG1 /WebSphere390
```

SYSPLEX
Is the name of your sysplex. WebSphere for z/OS derives this name from the predefined &SYSPLEX JCL variable.

SRVNAME
Is the server instance name.

Except for the initial installation of WebSphere for z/OS, you must manage the environment variables through the Administration application. At initial installation, the customization dialog modifies an initial environment file, which the bootstrap job uses.

There are, therefore, two distinct situations in which you define environmental data for your servers. Matching those situations are two distinct ways you create the environment data:

1. Prior to the bootstrap process, the customization dialog creates the environment file for you. The bootstrap job reads the file and places the environmental data into the system management database.
2. Defining and managing environmental data through the Administration application. In this situation, you enter environment data pairs (an environment name and its value—no "=") through a panel in the Administration application.

How run-time server start procedures point to their environment files

WebSphere for z/OS run-time server start procedures must point to an environment file for configuration information. The start procedures use a BBOENV DD statement with a PATH parameter that points to an HFS file. The BBOENV DD statement is:

```
//BBOENV    DD   PATH='&CBCONFIG/&RELPATH/&SYSPLEX/&SRVNAME/current.env'
```

where

&CBCONFIG

    Is a variable you set in the start procedure. It must match the read/write directory that you specify at installation time as the directory into which WebSphere for z/OS is to write configuration data and environment files. The default is `WebSphere390/CB390`.

&RELPATH

    Is a subdirectory (`controlinfo/envfile`). Its value must not change.

&SYSPLEX

    Is the name of your sysplex. Because it is a predefined JCL variable, you do not need to set it in your start procedure.

&SRVNAME

    Is the server instance name. By specifying the server instance name when you start the procedure, you can use the same start procedure for other server instances.

    Example: To pass the server instance name BBOASR1A to its start procedure, specify:

```
s bboasr1.bboasr1a,srvname='BBOASR1A'
```

    To use the same start procedure for server instance BBOASR1B, specify:

```
s bboasr1.bboasr1b,srvname='BBOASR1B'
```

Environment variables for z/OS or OS/390 clients

The Administration application does not manage environment variables for z/OS or OS/390 clients. You must create and manage z/OS or OS/390 client environment files and point to them from client programs. Table 42 on page 308 tells you which environment variables are required or optional for z/OS or OS/390 clients.

Note on using substitution variables

You cannot use variable substitution ($ variables) in environment statements. The variable substitution that is used in UNIX shell environments is not implemented in the Language Environment (LE). Because WebSphere for z/OS processes environment variables in the Language Environment, use of variables such as $PATH in a path environment variable will fail.

Example:

UNIX shell environments often set up paths by appending the new path to the existing path, like this:

```
PATH=yourdir
PATH=$PATH/mydir
```

The resulting path is PATH=yourdir/mydir after substitution for the $PATH variable. However, because WebSphere for z/OS processes the environment

variables in the Language Environment, where no variable assignment is made, the resulting path would be PATH=$PATH/mydir.

Environment variable syntax

You must follow this syntax only when defining your initial environment file before the bootstrap process.

Rules: The following are the syntax rules:

- The syntax of the environment variables follows this pattern:

  VARIABLE=VALUE

  Where:

  VARIABLE

  is the environment variable.

  VALUE

  is the setting for the variable. The descriptions define possible values for each variable.

- Leading and trailing white space (blanks or tabs) for both variables and values is ignored.

  Example: The two following lines yield the same result:

  VARIABLE1=VALUE1

  and

         VARIABLE1   =   VALUE1

- "=" is required.
- Blank lines are ignored.
- Code upper and lowercase characters as documented in this topic.
- To comment out an environment variable, simply add a character, such as '#', to the variable. For example, you could change TRACEALL=0 to #TRACEALL=0. The system ignores such coding because the variable does not begin with an alphabetic character.
- Language Environment limits the size of environment variables to 2K.

Environment variable use

Not all environment variables need to be used for each server or client. Table 42 on page 308 tells you where to use a given environment variable. Here are the meanings for what appears in each column:

- "R" means required.
- "O" means optional.
- "F" means required in a future release.
- A blank in the Default column means the variable is not set.
- A blank in other columns means the variable is not used.

Footnotes appear at the end of the table.

Note: The default settings and examples use the standard _CEE_ENVFILE syntax. You do not use this syntax when defining environmental data in the Administration application.

Table 42. Where to use environment variables

Environment variable=<default>	Daemon server instance	System Management server instance	Naming server instance	Interface Repository instance	Business application server instance	z/OS or OS/390 client
BBOC_HTTP_IDENTITY=*USER_ID*					R[1]	
BBOC_HTTP_INPUT_TIMEOUT =*n*					R[1]	
BBOC_HTTP_LISTEN_IP_ADDRESS=*IP_ADDRESS*					R[1]	
BBOC_HTTP_OUTPUT_TIMEOUT=*n*					R[1]	
BBOC_HTTP_PERSISTENT_SESSION_TIMEOUT =*n*					R[1]	
BBOC_HTTP_PORT=*n*					R[1]	
BBOC_HTTP_SESSION_GC=*n*					R[1]	
BBOC_HTTP_TRANSACTION_CLASS= *TRANSACTION_CLASS*					R[1]	
BBODUMP=3	O	O	O	O	O	
BBODUMP_CEE3DMP_OPTIONS=	O	O	O	O	O	
BBOLANG=ENUS	O	O	O	O	O	O
BEAN_DELETE_SLEEP_TIME=4200		R[2]				
CBCONFIG=/WebSphere390/CB390	R	R	R	R	R	
CLASSPATH=		O	O	O	O[3]	
CLIENT_DCE_QOP=NO_PROTECTION						O
CLIENT_HOSTNAME=						O
CLIENTLOGSTREAMNAME=						O
CLIENT_RESOLVE_IPNAME=<value for RESOLVE_IPNAME>		O	O	O	O	O
CLIENT_TIMEOUT=			O			
com.ibm.ws.naming.ldap.containerdn= <ibm-wsnTree=t1,o=<org>, c=<country>>			O			

Table 42. Where to use environment variables (continued)

Environment variable=<default>	Daemon server instance	System Management server instance	Naming server instance	Interface Repository instance	Business application server instance	z/OS or OS/390 client
com.ibm.ws.naming.ldap.domainname= *domain name*			O			
com.ibm.ws.naming.ldap.masterurl= ldap://<ip name>:<port>			O			
CONFIGURED_SYSTEM=	R[4]	R[4]	R[4]	R[4]	R[4]	
DAEMON_IPNAME=	R	O				
DAEMON_PORT=5555	O[5]	O[5]				
DEFAULT_CLIENT_XML_PATH=						O[6]
DEFAULT_UNAUTH_CLIENT_ID=CBGUEST	O	O				
DM_GENERIC_SERVER_NAME=CBDAEMON	O[5]	O[5]				
DM_SPECIFIC_SERVER_NAME=DAEMON01	O[7]	O[7]	O[7]	O[7]	O[7]	
HOME=					O	O
IBM_OMGSSL=0		O			O	
ICU_DATA=/usr/lpp/WebSphere/bin/		R				
IR_GENERIC_SERVER_NAME=CBINTFRP		O				
IR_SPECIFIC_SERVER_NAME=INTFRP01	O[7]	O[7]	O[7]	O[7]	O[7]	
IRPROC=BBOIR	O	O				
IVB_DEBUG_ENABLED=					O[8]	O[8]
IVB_DRIVER_PATH= /usr/lpp/WebSphere		R				
IVB_TRACE_HOST=						O[8]
IVB_TRACE_PORT=2102						O[8]
java.naming.security.credentials=<password>		O				
java.naming.security.principal=<userid>		O				

Table 42. Where to use environment variables (continued)

Environment variable=<default>	Daemon server instance	System Management server instance	Naming server instance	Interface Repository instance	Business application server instance	z/OS or OS/390 client
JAVA_COMPILER=					O	O
JAVA_IEEE754=						O[9]
JVM_BOOTCLASSPATH=		O			O	
JVM_BOOTLIBRARYPATH=		O			O	
JVM_DEBUG=		O			O	
JVM_DEBUG_PORT=					O	O[8]
JVM_ENABLE_CLASS_GC=		O			O	
JVM_ENABLE_VERBOSE_GC=		O			O	
JVM_EXTRA_OPTIONS=					O	
JVM_HEAPSIZE=256					O	
JVM_LOCALREFS=		O			O	
JVM_LOGFILE=					O	O
JVM_MINHEAPSIZE=		O			O	
LDAPBINDPW=		F	R[10]			
LDAPCONF=		F	R[10]			
LDAPHOSTNAME=		F	R[10]			
LDAPIRBINDPW=		F		R[11]		
LDAPIRCONF=		F		R[11]		
LDAPIRHOSTNAME=		F		R[11]		
LDAPIRNAME=		F		R[11]		
LDAPIRROOT=		F		R		
LDAPNAME=		F	R[10]			

Table 42. Where to use environment variables (continued)

Environment variable=<default>	Daemon server instance	System Management server instance	Naming server instance	Interface Repository instance	Business application server instance	z/OS or OS/390 client
LDAPROOT=		F	R			
LIBPATH=		O	O	O	O[3]	
LOGSTREAMNAME=	O	O				
MIN_SRS=[0 for MOFW, 1 for J2EE]		O			O	
NM_GENERIC_SERVER_NAME=CBNAMING						
NM_SPECIFIC_SERVER_NAME=NAMING01	O[7]	O[7]	O[7]	O[7]	O[7]	
NMPROC=BBONM	O	O				
OTS_DEFAULT_TIMEOUT=30	O	O	O	O	O	
OTS_MAXIMUM_TIMEOUT=60	O	O	O	O	O	
PATH=					O	O
RAS_MINORCODEDEFAULT= NODIAGNOSTICDATA						
REM_DCEPASSWORD=						O
REM_DCEPRINCIPAL=						O
REM_PASSWORD=		O[12]	O[12]	O[12]		O
REM_USERID=		O[12]	O[12]	O[12]		O
RESOLVE_IPNAME=		O[13]	O[14]	O[14]	O[14]	R[15]
RESOLVE_PORT=900		O	O	O	O	O
SM_DEFAULT_ADMIN= CBADMIN		O				
SM_GENERIC_SERVER_NAME=CBSYSMGT		O				
SM_SPECIFIC_SERVER_NAME=SYSMGT01	O[7]	O[7]	O[7]	O[7]	O[7]	
SMPROC=BBOSMS	O	O				
SOMOOSQL=					O	

Table 42. Where to use environment variables (continued)

Environment variable=<default>	Daemon server instance	System Management server instance	Naming server instance	Interface Repository instance	Business application server instance	z/OS or OS/390 client
SRVIPADDR=	O	O	O	O	O	
SSL_KEYRING=						O
SYS_DB2_SUB_SYSTEM_NAME=DB2	R	R	R	R	R	
TRACEALL=1	O	O	O	O	O	O
TRACEBASIC=	O	O	O	O	O	O
TRACEBUFFCOUNT=4	O	O	O	O	O	
TRACEBUFFLOC=(Server: BUFFER, Client: SYSPRINT)	O	O	O	O	O	O
TRACEBUFFSIZE=1M	O	O	O	O	O	
TRACEDETAIL=	O	O	O	O	O	O
TRACEMINORCODE=						
TRACEPARM=00	O					

Table 42. Where to use environment variables (continued)

Environment variable=<default>	Daemon server instance	System Management server instance	Naming server instance	Interface Repository instance	Business application server instance	z/OS or OS/390 client

Notes:

1. Required if using the HTTP Transport Handler function of the J2EE server.

2. Required when stateful session beans in J2EE servers are activated based on a transaction, rather than activated only once.

3. Required for server regions that use Java, including the IMS PAA and CICS PAA.

4. This environment variable is automatically added to each server instance's environment file and should not be edited.

5. If you specify a value for the Daemon Server, you must provide the same value for the System Management Server control region.

6. Required when the client uses the System Management Scripting API.

7. You must specify this for the second and subsequent systems in a sysplex.

8. Required only when you are using the IBM Object Level Trace and Distributed Debugger Tools to trace and/or debug client and server application components.

9. Required for Java clients that run on z/OS or OS/390.

10. LDAPCONF is mutually exclusive with LDAPBINDPW, LDAPHOSTNAME, and LDAPNAME. Either LDAPCONF is required, or LDAPBINDPW, LDAPHOSTNAME, and LDAPNAME are required.

11. LDAPIRCONF is mutually exclusive with LDAPIRBINDPW, LDAPIRHOSTNAME, and LDAPIRNAME. Either LDAPIRCONF is required, or LDAPIRBINDPW, LDAPIRHOSTNAME, and LDAPIRNAME are required.

12. Used when a server becomes a remote client of another server.

13. For the control region, the default is the value of DAEMON_IPNAME during bootstrap.

14. For the server region, the default is the local system IP name. Generally, do not code.

15. Optional if a Daemon Server is on the same system as the client, in which case the default is the local system IP name.

Environment variable descriptions

BBOC_HTTP_IDENTITY=*USER_ID*

Specifies a valid SAF user ID which will be used as the current security principal for this HTTP request. The user ID will be treated as an authenticated user by the Web container. If this variable is not specified, the request will be executed under the SCO's "local Identity" (Local_Identity()).

Example:

BBOC_HTTP_IDENTITY=SECURITY1

BBOC_HTTP_INPUT_TIMEOUT=*n*

The time in seconds that the J2EE server will allow for the complete HTTP request to be received before cancelling the connection. The default value is 10 seconds.

Example:

BBOC_HTTP_INPUT_TIMEOUT=10

BBOC_HTTP_LISTEN_IP_ADDRESS=*IP_ADDRESS*

Specifies the IP address, in dotted decimal format, that WebSphere for z/OS J2EE servers use to listen for HTTP client connection requests. This IP address is used by the server to bind to TCP/IP. Normally, the server will listen on all IP addresses configured to the local TCP/IP stack. However, if you want to fence the work or allow multiple heterogeneous servers to listen on the same port, you can use BBOC_HTTP_LISTEN_IP_ADDRESS. The specified IP address becomes the only IP address over which this control region receives inbound HTTP requests.

Example:

BBOC_HTTP_LISTEN_IP_ADDRESS=9.117.43.16

BBOC_HTTP_OUTPUT_TIMEOUT=*n*

The time, in seconds, that the J2EE server will wait from the time the complete HTTP request is received until output is available to be sent to the client. The default value is 120 seconds.

Example:

BBOC_HTTP_OUTPUT_TIMEOUT=120

BBOC_HTTP_PERSISTENT_SESSION_TIMEOUT =*n*

Specifies the time, in seconds, that the J2EE server will wait between requests issued over a persistent connection from an HTTP client. After the server sends a response, it uses the persistent timeout to determine how long it should wait for a subsequent request before cancelling the persistent connection. The default value is 30 seconds.

Example:

BBOC_PERSISTENT_SESSION_TIMEOUT=10

BBOC_HTTP_PORT=*n*

Specifies the port at which the J2EE server listens for HTTP requests. Any requests received over the HTTP port will be directed to the Web container for processing.

If this variable is not specified, the J2EE server will not listen for HTTP requests directly.

The use of this HTTP port does not preclude the use of the WebSphere for z/OS plug-in with this J2EE server instance. The Web container is capable of simultaneously processing requests received directly through the HTTP port as well as from the WebSphere for z/OS plug-in.

Note: Currently, HTTP requests received over this HTTP port are not able to be authenticated using the mechanisms described in the J2EE Specification.

Example:

BBOC_HTTP_PORT=8080

BBOC_HTTP_SESSION_GC=*n*

An integer value indicating the maximum number of HTTP requests that will be processed over a single connection from an HTTP client. When the maximum number of requests have been processed, the client connection will be closed. Set this value to 0 or 1 to turn off persistent connection processing. The default value is 50.

Example:

BBOC_HTTP_SESSION_GC=50

BBOC_HTTP_TRANSACTION_CLASS=*TRANSACTION_CLASS*

A valid WLM transaction class, which will be used in the creation of the WLM enclave for all HTTP requests. If a valid WLM transaction class is not specified, no transaction class will be set for the enclave.

Example:

BBOC_HTTP_TRANSACTION_CLASS=TCLASSA

BBODUMP=*n*

Specifies the default dump used by the signal handler. Valid values and their meanings are:

0   No dump is generated.

1   A ctrace dump is taken.

2   A cdump dump is taken.

3   A csnap dump is taken.

4   A CEE3DMP dump is taken. CEE3DMP generates a dump of Language Environment and the member language libraries. Sections of the dump are selectively included, depending on dump options specified, either by default or through the BBODUMP_CEE3DMP_OPTIONS environment variable. By default, this value passes THREAD(ALL) BLOCKS to CEE3DMP. You can override the default options for CEE3DMP through the BBODUMP_CEE3DMP_OPTIONS environment variable.

For more information about CEE3DMP and its options, see *z/OS Language Environment Programming Reference*, SA22-7562.

If you do not specify BBODUMP, the default value is 3 (a csnap dump is taken).

Example:
BBODUMP=3

BBODUMP_CEE3DMP_OPTIONS=*options*
Specifies dump options to be used with a CEE3DMP. This environment variable is used when you specify BBODUMP=4. For an explanation of CEE3DMP and valid dump options, see *z/OS Language Environment Programming Reference*, SA22-7562.

Rule: The maximum length of the option string on this environment variable is 255. If the option string is longer than 255, you receive message BBOU0514W and the CEE3DMP dump options are set to THREAD(ALL) BLOCKS.

Example:
BBODUMP_CEE3DMP_OPTIONS=NOTRACEBACK NOFILES

BBOLANG=*LANGUAGE*
The name of the WebSphere for z/OS message catalog used. The default is ENUS.

BEAN_DELETE_SLEEP_TIME=n
The time in seconds allowed before an expired stateful session bean's state is deleted from its backing datastore (DB2). The default time is 4200 seconds (70 minutes). You can increase the time to 2147483 seconds (24.85 days). **Recommendation:** Do not set this variable less than 300 seconds (5 minutes).

Example: BEAN_DELETE_SLEEP_TIME=1000000

CBCONFIG=*path*
Specifies a read/write directory in the HFS into which WebSphere for z/OS writes configuration and environment files when a conversation is activated. The &CBCONFIG variable in control and server region start procedures must match this value. In this way, WebSphere for z/OS can

find the appropriate environment file for a server when those start procedures are executed. The default is /WebSphere390/CB390.

Example: `CBCONFIG=/WebSphere390/CB390`

Rule: The System Management group (default CBCFG1) and user ID (default CBSYMSR1) must own each directory and subdirectory in CBCONFIG. If the System Management group and user ID do not own CBCONFIG, use the chown command to make them the owner of each directory and subdirectory in CBCONFIG. Thus, if you use the default CBCONFIG, you must use the chown command to give the System Management group and user ID ownership of /WebSphere390 and /WebSphere390/CB390.

Example:

`chown -R CBSYMSR1:CBCFG1 /WebSphere390`

CLASSPATH=*path1***:[***path2***]:...**
Specifies Java class files—.jar files and classes.zip files—for use by Java business objects in server regions. Specify your Java business object's .jar files when you use Java business objects. The entire CLASSPATH statement must be on one line only.

Example:

`CLASSPATH=/usr/lpp/db2/db2710/classes/db2j2classes.zip: . . .`

CLIENT_DCE_QOP=*value*
The level of DCE message protection used by a local z/OS or OS/390 client to apply to the current transaction flows. Normally, you would set DCE security for an z/OS or OS/390 client that accesses servers on remote systems. Note that the DCE level for a server is set through the Administration application.

When enabled on client and server, DCE authentication offers each proof of the other's legitimacy with a handshake message exchange using DCE's third-party authentication scheme. Once this exchange has taken place, messages can be assigned one of three levels of protection, which are the values of this environment variable:

NO_PROTECTION
DCE assures only that the messages and their replies are from the legitimate sender. This is the default.

INTEGRITY
DCE assures that the message is from the legitimate sender and it has not been modified in any way since the sender sent it.

CONFIDENTIALITY
DCE encrypts the message so that none but the legitimate receiver can read it.

CLIENT_HOSTNAME=
Allows an z/OS or OS/390 client to determine its host IP name when no Daemon is running on the same system. When a client program issues the CBSeriesGlobal::hostName() method, the system checks the CLIENT_HOSTNAME environment variable first and returns this value, if it is set. If the value is not set, the system returns the IP name of the Daemon running on that system, if the Daemon is running. The default value is null.

Example: `CLIENT_HOSTNAME=MYSYS.SYS.COM`

CLIENTLOGSTREAMNAME=*LOG_STREAM_NAME*
The WebSphere for z/OS error log stream to which an z/OS or OS/390 client ORB writes error information.

Example: `CLIENTLOGSTREAMNAME=MY.CLIENT.ERROR.LOG`

CLIENT_RESOLVE_IPNAME=*IP_NAME*
The Internet Protocol name that an z/OS or OS/390 client, or server region acting as a client, uses to access the bootstrap server (that is, when the client or server region invokes the resolve_initial_references method). The default is the value specified by the RESOLVE_IPNAME environment variable, which is the Internet Protocol name associated with the System Management Server (the default bootstrap server). If RESOLVE_IPNAME is not set, the value is the system on which the client or server region is running.

The CLIENT_RESOLVE_IPNAME environment variable allows you to specify a bootstrap server running on a remote system, while other clients use a local bootstrap server defined by the RESOLVE_IPNAME environment variable.

Note: The TCP/IP port number for the CLIENT_RESOLVE_IPNAME is defined by the RESOLVE_PORT environment variable.

The value of CLIENT_RESOLVE_IPNAME can be up to 255 characters.

Example: `CLIENT_RESOLVE_IPNAME=REMHOST`

CLIENT_TIMEOUT=*n*
Sets the time-out value for response from a client method call. The values are in integers and signify the time in tenths of seconds (thus, a value of 10 is 1 second). The default value is 0, which means no time-out value is set.

Example: `CLIENT_TIMEOUT=20`

com.ibm.ws.naming.ldap.containerdn=*ibm-wsnTree=t1,o=org,c=country*
The starting point of WsnName tree. Only the Naming server uses this environment variable. By default, the system expects the value to be

ibm-wsnTree=t1,o=WASNaming,c=us. If you take the default, delete this
environment variable from your environment file.

This value must match the value specified in LDAP initialization file (our
sample is bboldif.cb). If you've modified the organization or country in
your bboldif.cb file, use the same value on this environment variable.
Note that case does not matter in LDAP, though it does matter for the
environment variables. The "o=,c=" portion must also be specified as a
suffix in bboslapd.conf.

Example:

```
suffix   "o=WASNaming,c=us"
```

Tip: The suffix statement appears as:

```
suffix          "<ws_rdn>"
```

in the sample bboslapd.conf we ship.

Example:

```
com.ibm.ws.naming.ldap.containerdn=ibm-wsnTree=t1,o=WASNaming,c=us
```

com.ibm.ws.naming.ldap.domainname=*domain name*
Uniquely identifies the host root and is the basis for partitioning the JNDI
global name space. Only the Naming server uses this environment
variable. By default, the system expects the value to be the domain name
of the sysplex on which Naming Server is running. If you want the
default, delete this environment variable from the environment file. If you
want a different domain name, specify it.

Example:

```
com.ibm.ws.naming.ldap.domainname=plex1
```

com.ibm.ws.naming.ldap.masterurl=ldap://*IP_name:port*
The LDAP Server IP Name and port number. Only the Naming server
uses this environment variable. By default, the system expects the IP name
to be the same as the system on which the Naming Server runs and the
port to be 1389. If your LDAP server is running on a system other then
the one the Naming Server runs on or uses a port other than 1389, update
this environment variable. Otherwise, delete this environment variable.

Example:

```
com.ibm.ws.naming.ldap.masterurl=ldap://wsldap:1389
```

CONFIGURED_SYSTEM=*system*
Specifies the name of the system to which the server instance was
originally configured. During prepare for cold start, cold start, and server
activation, the run time adds this environment variable to each server
instance's environment file automatically.

Rule: Do not manually add or change this environment variable at any time, such as:
- In the initial environment file before bootstrap
- Through the Administration application (SM EUI)
- In an existing server environment file.

DAEMON_IPNAME=*IP_NAME*
The Internet Protocol name that the Daemon Server registers with the Domain Name Service (DNS). Any CORBA client communication with WebSphere for z/OS requires this IP name.

You must define the DAEMON_IPNAME environment variable at installation time, before you start the Daemon bootstrap process. Otherwise, WebSphere for z/OS issues an error message and terminates the Daemon.

The bootstrap process sets, among other things, the Daemon IP name in the system management database. After bootstrap, WebSphere for z/OS uses the value in the system management database. It is possible that, after bootstrap, the value of the DAEMON_IPNAME environment variable could change to a value other than what is in the system management database. If this happens, an error message is issued, but the Daemon initializes with the Daemon IP name from the system management database.

To place Daemon server instances in the same host cluster, you must code the same DAEMON_IPNAME value for each server instance.

Rules:
- The value for DAEMON_IPNAME must be a fully-qualified long name.
- The first-level qualifier can be from 1 to 18 characters.
- Once chosen, the port and IP name for the Daemon should not change, since every object reference includes the port and IP name—if you change them, existing objects will no longer be accessible.

Example: DAEMON_IPNAME=CBQ091.PDL.POK.IBM.COM

DAEMON_PORT=*n*
The port number at which the Daemon Server listens for requests. The default is 5555. If you specify a value, you must provide the same value for the System Management Server control region.

Example: DAEMON_PORT=5555

DEFAULT_CLIENT_XML_PATH=*path*
Specifies the location of a set of XML files that hold default parameter

lists used by the System Management Scripting API. You must set this environment variable for clients that use the System Management Scripting API.

IBM provides a set of sample XML files that contain default parameter lists. After installation, these samples reside in `/usr/lpp/WebSphere/samples/smapi`. For information about the XML files and the parameter lists, see *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management Scripting API*, SA22-7839.

You can override the default behavior of the System Management Scripting API in two ways:

1. Specifying the parameters explicitly in the REXX script that calls the System Management Scripting API. By specifying parameters explicitly, you do not have to modify the XML samples IBM provides. You simply need to code

   `DEFAULT_CLIENT_XML_PATH=/usr/lpp/WebSphere/samples/smapi`

   in your client environment file.

2. Copying the XML files to another directory (the samples IBM provides are read-only), making modifications to the parameter lists, then changing the DEFAULT_CLIENT_XML_PATH to point to the new directory. Making these changes is required only if you want to override permanently the default behavior of the System Management Scripting API.

Example: `DEFAULT_CLIENT_XML_PATH=/usr/lpp/WebSphere/samples/smapi`

DEFAULT_UNAUTH_CLIENT_ID=`user_id`

The default local and remote user ID that the System Management server associates with servers. If you allow unauthenticated client requests on a server, and do not explicitly specify your own local and remote user ID for that server, those requests run under the authority of this user ID.

If you do not define this environment variable, the default local and remote user ID is CBGUEST.

You must define this user ID to z/OS or OS/390 and give it appropriate security authorizations (for example, RACF permissions and LDAP permissions).

This environment variable is used only by the System Management server. Using this environment variable in the environment file for other servers takes no effect. That is, you cannot use this environment variable for other servers to define the default local and remote ID that is used by those servers. Rather, you must define the default through the server properties panel in the Administration application. To do this

- Select the "Allow non-authenticated clients" checkbox. The Administration application supplies the value for the local and remote identity from the value on the DEFAULT_UNAUTH_CLIENT_ID variable (or, if not specified, it supplies CBGUEST).
- Type over the supplied values with your value.

The System Management server uses this environment variable during bootstrap. After bootstrap, you can modify the value only at the sysplex level through the Administration application.

Example: DEFAULT_UNAUTH_CLIENT_ID=DUDE

DM_GENERIC_SERVER_NAME=*SERVER_NAME*
The server name for the Daemon Server. The default is CBDAEMON. If you specify a value, you must provide the same value for the System Management Server control region.

Example: DM_GENERIC_SERVER_NAME=CBDAEMON

DM_SPECIFIC_SERVER_NAME=*SERVER_INSTANCE_NAME*
A server instance name of the Daemon Server. The default is DAEMON01. You must specify this environment variable for all server instances in the second and subsequent systems in a sysplex.

Example: DM_SPECIFIC_SERVER_NAME=DAEMON01

HOME=*path*
Specifies the home directory. This variable is set automatically from the security product user profile when the user logs in to the UNIX shell.

IBM_OMGSSL=[0 | 1]
Specifies whether only CORBA-compliant security tags will be exported by the server. The value 1 means only CORBA-compliant tags are exported. The value 0 (the default) means CORBA-compliant and non-compliant tags are exported.

Use value 1 when the server uses only SSL basic authentication for its security and clients (such as CICS or other OEM ORBs) use CORBA-compliant tags. This is only in the case when the server uses SSL basic authentication. If your server supports SSL client certificates as well, you do not have to set this variable.

Use value 0 (or take the default) when your server uses SSL basic authentication and interoperates with WebSphere clients on distributed platforms or WebSphere Application Server Enterprise Edition for OS/390 V3.02.

Example: IBM_OMGSSL=1

ICU_DATA=*path*
The path to binary files required by the XML Parser used by the System

Management server during bootstrap and import server processing. If you installed the WebSphere for z/OS code in the default directory, you do not need to change this path. The default path is /usr/lpp/WebSphere/bin/.

Example: `ICU_DATA=/usr/lpp/WebSphere/bin/`

IR_GENERIC_SERVER_NAME=*SERVER_NAME*
The server name of the Interface Repository Server. The default is CBINTFRP. You must define a workload management (WLM) application environment using this name for the Interface Repository Server server regions to work.

IR_SPECIFIC_SERVER_NAME=*SERVER_INSTANCE_NAME*
A server instance name of the Interface Repository Server. The default is INTFRP01. You must specify this environment variable for all server instances in the second and subsequent systems in a sysplex.

IRPROC=*PROC_NAME*
The start procedure used by the Daemon Server to start the Interface Repository Server. The default is BBOIR. You can supply the name of your own start procedure. If you do so, copy the information from the default start procedure to your new start procedure.

Example: `IRPROC=BBOIR`

IVB_DEBUG_ENABLED=1
Enables the z/OS or OS/390 client and the application server to load the object level trace run time, and to use object level trace for tracing and/or debugging client and server application components. The value 1 is required for the application server, and for both C++ or Java clients running on z/OS or OS/390, when debugging C++ or Java business objects, servlets, JSPs, or Enterprise beans.

IVB_DRIVER_PATH=*path*
The name of the directory where WebSphere for z/OS files reside after SMP/E installation. The default is /usr/lpp/WebSphere.

Example: `IVB_DRIVER_PATH=/usr/lpp/WebSphere`

IVB_TRACE_HOST=*IP_ADDRESS (or HOSTNAME)*
Specifies the workstation IP address (or host name if you have the DNS server setup correctly) where the object level trace viewer runs. Use this when you are tracing and/or debugging your client and server components with the IBM Object Level Trace and Distributed Debugger Tools.

Example: `IVB_TRACE_HOST=MYHOST.IBM.COM`

IVB_TRACE_PORT=*port*
Specifies the same port as the TCP/IP port specified for the object level

trace server. Use this when you are tracing and/or debugging your client and server components with the IBM Object Level Trace and Distributed Debugger Tools. The default is 2102.

Example: `IVB_TRACE_PORT=2102`

java.naming.security.credentials=*password*

The password used by the distinguished name specified by java.naming.security.principal. The password must match the password defined for the administrator access ID (default is WASAdmin) by the LDAP initialization file during initial system customization. IBM provides the WASAdmin access ID in a sample LDIF file called bboldif.cb. The default value is `secret`.

Example: `java.naming.security.credentials=secret`

Recommendation: You should change the IBM-supplied password.

java.naming.security.principal=*distinguished_name*

Distinguished name (user ID) defined to have write access to WsnName directory. Specify this only if you want to provide read/write access to all JNDI users. The distinguished name must match the one defined for the administrator access ID (default is WASAdmin) by the LDAP LDIF file during initial system customization. IBM provides the WASAdmin access ID in a sample LDAP initialization file called bboldif.cb. The default value is `cn=WASAdmin,o=WASNaming,c=us`.

Example:
`java.naming.security.principal=cn=WASAdmin,o=WASNaming,c=us`

Recommendation: We suggest you keep the WASAdmin access ID.

JAVA_COMPILER=

Specifies the use of the just-in-time (JIT) compiler.

If you use the environment variable, a null value (`JAVA_COMPILER=`) turns the JIT compiler on. Any other value turns the JIT compiler off.

By default, a Java virtual machine (JVM) running on z/OS or OS/390 uses the JIT compiler, so you do not have to explicitly set this environment variable. If you are debugging Java business objects or J2EE application components, however, turn off the JIT compiler by specifying a non-null value.

Example: `JAVA_COMPILER=NONE`

JAVA_IEEE754=EMULATION

Specifies the correct executable code for the system to load for the Java virtual machine (JVM) in which Java clients on z/OS or OS/390 run. This environment variable setting is required only for Java clients that run on z/OS or OS/390.

JVM_BOOTCLASSPATH=*path1***:***[path2]*
>   Enables the use of bootclasspath. This option is equivalent to the
>   `-Xbootclasspath/p:` Java invocation option.

JVM_BOOTLIBRARYPATH=*path1***:***[path2]*
>   Enables the use of bootlibrarypath. This option is equivalent to the
>   `-Dsun.boot.library.path=` Java invocation option.

JVM_DEBUG=1
>   This option is equivalent to the `–verbose:class,jni` Java invocation
>   option. It reroutes JVM messages to SYSOUT for debugging purposes. Set
>   JVM_DEBUG=1 to invoke JVM messaging.

JVM_DEBUG_PORT=*port*
>   Specifies a TCP/IP port that the distributed debugger uses to connect to
>   the JVM.

JVM_ENABLE_CLASS_GC=1
>   Enables class objects to be garbage collected. The value 1 is required for
>   enabling class object garbage collection. This option is equivalent to the
>   `-Xnoclassgc` Java invocation option.

JVM_ENABLE_VERBOSE_GC=1
>   Sets verbose garbage collection on or off. The value 1 is required for
>   enabling garbage collection messages. This option is equivalent to the
>   `-verbose:gc` Java invocation option.

JVM_EXTRA_OPTIONS=*string*
>   Allows you to specify one new Java environment variable that is not
>   already predefined by IBM (those predefined variables start with `JVM_`).
>   With `JVM_EXTRA_OPTIONS`, *string* is the new Java option or property that
>   you want to specify.

JVM_HEAPSIZE=*n*
>   Sets the maximum size (in megabytes) of the JVM heap. The default is 256
>   MB. This option is equivalent to the `-Xmx=xxxM` Java invocation option.
>
>   **Example:** `JVM_HEAPSIZE=256 # specifies a 256 MB heap`

JVM_LOCALREFS=
>   Should only be used under the direction of IBM support. The default is
>   128.

JVM_LOGFILE=*filename*
>   Specifies the HFS file in which messages from the JVM will be logged.
>
>   **Recommendation:** Use this variable only in a single-server environment.
>   If you use `JVM_LOGFILE` in a multiple-server environment, all the servers
>   write to the same file, so you might have difficulty using the file for
>   diagnostic purposes. In a multiple-server environment, use `JVM_DEBUG=1` to
>   direct JVM messages to the SYSOUT for a specific server.

JVM_MINHEAPSIZE=*n*

>Sets the mimimum size (in megabytes) of the JVM heap. The default is 256 MB. This option is equivalent to the `-Xms=xxxM` Java invocation option. For optimal performance, specify the same value for `JVM_HEAPSIZE` and `JVM_MINHEAPSIZE`.

LDAPBINDPW=*password*

>The password the Naming Server uses to bind to the LDAP server. Used in conjunction with LDAPNAME.

LDAPCONF=*filename*

>The LDAP configuration file used by WebSphere for z/OS. If you designate a file in the HFS, do not use quotes. If you designate an MVS data set, enclose the data set in single quotes.

>**Example:** `LDAPCONF='bbo.s21slapd.conf'`

LDAPHOSTNAME=*name:port*

>The host name of the LDAP server that the Interface Repository Server uses as its data store.

LDAPIRBINDPW=*password*

>The password the Interface Repository Server uses to bind to the LDAP server. Used in conjunction with LDAPIRNAME.

LDAPIRCONF=*filename*

>The LDAP configuration file used by the LDAP server that the Interface Repository Server uses as its data store. If you designate a file in the HFS, do not use quotes. If you designate an MVS data set, enclose the data set in single quotes.

LDAPIRHOSTNAME=*name:port*

>The host name of the LDAP server that the Interface Repository Server uses as its data store.

LDAPIRNAME

>The LDAP entry name that the Interface Repository Server uses to authenticate itself to the LDAP server that it uses as its data store.

LDAPIRROOT=*root*

>The LDAP entry name at which the Interface Repository Server anchors its data.

>**Example:** `LDAPIRROOT=o=BOSS,c=U`

LDAPNAME

>The LDAP entry name that the Naming Server uses to authenticate itself to the LDAP server that it uses as its data store.

LDAPROOT=*root*

>The LDAP entry name at which the Naming Server anchors its data.

Example: `LDAPROOT=o=BOSS,c=US`

LIBPATH=_path1:[path2]:..._
Specifies the DLL search paths for Java in the hierarchical file system (HFS). Specify system, WebSphere for z/OS, and Java DLLs.

Example:

`LIBPATH=`*db2_install_path*`/lib:/usr/lpp/java/J1.3/bin:/usr/lpp/java/J1.3/bin/classic:/usr/lpp/WebSphere/lib`

where *db2_install_path* is the HFS where you installed DB2.

LOGSTREAMNAME=_LOG_STREAM_NAME_
The WebSphere for z/OS error log stream name the Daemon and System Management servers use during bootstrap. If not specified in the environment file for the Daemon and System Management servers during bootstrap, the system uses the following algorithm to form an error log stream name. WebSphere for z/OS:

1. Takes the first qualifier in the Daemon Server's IP name.
2. If the first qualifier is more than 8 characters, divides the qualifier into 8-character strings and separates them with periods.
3. Adds a high-level qualifier "BBO".

For example, if the Daemon IP name is MYDAEMONSERVER.IBM.COM, the algorithm would produce an error log stream name BBO.MYDAEMON.SERVER.

After bootstrap, you can create or change an error log stream name for the entire sysplex, a server, or a server instance through the Administration application. A server error log stream setting overrides the general WebSphere for z/OS setting, and a server instance setting overrides a server setting. Thus, you can set up general error logging, but direct error logging for servers or server instances to specific log streams.

During processing, if the specified log stream is not found or not accessible, a message is issued and errors are written to the server's joblog.

Example: `LOGSTREAMNAME=MY.CB.ERROR.LOG`

Tip: Do not put the log stream name in quotes. Log stream names are not data set names.

MIN_SRS=_nn_
The number of server regions to be kept running once those server regions have initialized. That is, workload management will not direct the server region to shut down even though it becomes inactive. Use this

environment variable when the response time for the workload requires that several server regions are always ready to process work.

The default for J2EE servers is 1. For MOFW servers, the default is 0. The maximum value is 20. If you specify more than 20, the variable is set to 20.

WebSphere for z/OS garbage collection may cause a server region to refresh, but the minimum number of server regions will not fall below the value specified on this environment variable.

Example: `MIN_SRS=2`

NM_GENERIC_SERVER_NAME=_SERVER_NAME_
The server name of the Naming Server. The default is CBNAMING. You must define a workload management (WLM) application environment using this name for the Naming Server server regions to work.

Example: `NM_GENERIC_SERVER_NAME=CBNAMING`

NM_SPECIFIC_SERVER_NAME=_SERVER_INSTANCE_NAME_
The server instance name of the Naming Server. The default is NAMING01. You must specify this environment variable for all server instances in the second and subsequent systems in a sysplex.

Example: `NM_SPECIFIC_SERVER_NAME=NAMING01`

NMPROC=_PROC_NAME_
The start procedure used by the Daemon Server to start the Naming Server. The default is BBONM. You can supply the name of your own start procedure. If you do so, copy the information from the default start procedure to your new start procedure.

Example: `NMPROC=BBONM`

OTS_DEFAULT_TIMEOUT=n
The amount of time (in seconds) given by default to an application transaction to complete. This amount of time is given to the application transaction if it does not set its own time-out value through the `current –> set_timeout` method.

The default is 30 seconds and the maximum value is 2147483 seconds (24.85 days). You should not use a null or 0 value.

Note: When a conversation is activated, the system performs special processing for the System Management server instances **only**.
- If the OTS_DEFAULT_TIMEOUT variable is not set, it is added.
- If the value for OTS_DEFAULT_TIMEOUT is less than 3600 (seconds), it is set to 3600.

This special processing is performed for the System Management server instances because the server instances sometimes perform long-running transactions. Other server instances do not require such lengthy transaction defaults.

Example: `OTS_DEFAULT_TIMEOUT=30`

OTS_MAXIMUM_TIMEOUT=*n*

The maximum allowable amount of time (in seconds) given to an application transaction to complete. If an application assigns a greater amount of time, the system limits the time to the OTS_MAXIMUM_TIMEOUT value.

The default is 60 seconds and the maximum value is 2147483 seconds (24.85 days). You should not use a null or 0 value.

Note: When a conversation is activated, the system performs special processing for the System Management server instances **only**.
- If the OTS_MAXIMUM_TIMEOUT variable is not set, it is added.
- If the value for OTS_MAXIMUM_TIMEOUT is less than 3600 (seconds), it is set to 3600.

This special processing is performed for the System Management server instances because the server instances sometimes perform long-running transactions. Other server instances do not require such lengthy transaction defaults.

Example: `OTS_MAXIMUM_TIMEOUT=60`

PATH=*path*

Specifies the path.

RAS_MINORCODEDEFAULT=*value*

Determines the default behavior for gathering documentation about system exception minor codes. Use only under the guidance of IBM Service.

CEEDUMP

Captures callback and offsets.

Tip: It takes time for the system to take CEEDUMPs and this may cause transaction timeouts. For instance, your OTS_DEFAULT_TIMEOUT may be set to 30 seconds, but, since taking a CEEDUMP can take longer than 30 seconds, your application transaction may time out. To prevent this from happening, either:
- Increase the transaction timeout value.

  or

- Code RAS_MINORCODEDEFAULT=NODIAGNOSTICDATA.
  Be sure TRACEMINORCODE is **not** in the environment file.

TRACEBACK
Captures Language Environment and z/OS UNIX traceback data.

SVCDUMP
Captures an MVS dump (but will not produce a dump in the client).

NODIAGNOSTICDATA
The default. This setting will not cause the gathering of a CEEDUMP, TRACEBACK, or SVCDUMP.

Note: Sometimes results depend on the setting of another environment variable, TRACEMINORCODE. If you code TRACEMINORCODE=(null value) and RAS_MINORCODEDEFAULT=TRACEBACK you get a traceback. But, if you code RAS_MINORCODEDEFAULT=NODIAGNOSTICDATA and TRACEMINORCODE=ALL, you also get a traceback. So, specifying RAS_MINORCODEDEFAULT=NODIAGNOSTICDATA does not cancel TRACEBACK; it simply does not cause a TRACEBACK to be gathered.

REM_DCEPASSWORD=_password_
The password of the remote DCE principal passed in the security context when an z/OS or OS/390 client makes a request to a system outside the sysplex and SSL Type 1 authentication is being used. The password must conform to DCE requirements for passwords.

Example: REM_DCEPASSWORD=mydcePW

REM_DCEPRINCIPAL=_principal_
The principal passed in the security context when a client makes a request to a system outside the sysplex and SSL Type 1 authentication is being used. This principal must be defined on the target server. The value must conform to DCE requirements for principals.

Example: REM_DCEPRINCIPAL=myDCEprin

REM_PASSWORD=_password_
The password used in the security context when a client makes a request to a remote z/OS or OS/390 system and user ID/password security or SSL security is being used.

Example: REM_PASSWORD=MYPASSW

REM_USERID=_USER_ID_
The user ID used in the security context when a client makes a request to a remote z/OS or OS/390 system and user ID/password security or SSL security is being used.

Example: `REM_USERID=MCOX`

RESOLVE_IPNAME=*IP_NAME*

The Internet Protocol name that the System Management Server registers with the Domain Name Service (DNS). Any CORBA client communication with WebSphere for z/OS requires this IP Name. If not set, the Resolve IP Name is the system on which the program is running.

Rule: The value for RESOLVE_IPNAME should be a fully-qualified name, but it cannot exceed 255 characters.

Example: `RESOLVE_IPNAME=CBQ091.COMPANY.NY.COM`

RESOLVE_PORT=*n*

The port number at which the System Management Server listens for requests. The default is 900. This is a well-known port for Object Request Brokers, so IBM advises that you do not change this variable. If you already have an application that uses this port, consider using TCP/IP bind-specific support and the SRVIPADDR environment variable.

Example: `RESOLVE_PORT=900`

SM_DEFAULT_ADMIN=*USER_ID*

The user ID for the administrator who uses the Administration and Operations applications. This environment variable is used by the System Management bootstrap during installation—setting this environment variable after the System Management bootstrap runs has no effect. If you do not define this environment variable, the default user ID is CBADMIN. You must define this user ID to z/OS or OS/390 and give it appropriate security authorizations (for example, RACF permissions and LDAP permissions).

Note: After the System Management bootstrap runs, you can define additional administrator user IDs only through the Administration application. Those user IDs do not replace the user ID defined by SM_DEFAULT_ADMIN.

Example: `SM_DEFAULT_ADMIN=DUDE`

SM_GENERIC_SERVER_NAME=*SERVER_NAME*

The server name of the Systems Management Server. The default is CBSYSMGT. You must define a workload management (WLM) application environment using this name for the Systems Management Server server regions to work.

Example: `SM_GENERIC_SERVER_NAME=CBSYSMGT`

SM_SPECIFIC_SERVER_NAME=*SERVER_INSTANCE_NAME*

The server instance name of the Systems Management Server. The default

is SYSMGT01. You must specify this environment variable for all server instances in the second and subsequent systems in a sysplex.

Example: SM_SPECIFIC_SERVER_NAME=SYSMGT01

SMPROC=*PROC_NAME*
The start procedure used by the Daemon Server to start the Systems Management Server. The default is BBOSMS. You can supply the name of your own start procedure. If you do so, copy the information from the default start procedure to your new start procedure.

Example: SMPROC=BBOSMS

SOMOOSQL=*value*
Improves performance for client applications that use object-oriented SQL queries on string attributes. By using SOMOOSQL=1, string comparisons are pushed down to the database.

The default value is null (SOMOOSQL=).

Rule: You can use SOMOOSQL=1 only when the database and server region address spaces have been declared to run in the same locale.

SRVIPADDR=*IP_ADDRESS*
The IP address in dotted decimal format that WebSphere for z/OS servers use to listen for client connection requests.

This IP address is used by the server to bind to TCP/IP. Normally, the server will listen on all IP addresses configured to the local TCP/IP stack. However if you want to fence the work or allow multiple heterogeneous servers to listen on the same port, you can use SRVIPADDR. The specified IP address becomes the only IP address over which WebSphere for z/OS receives inbound requests. Normally, you also have to map the Daemon IP name, resolve IP name, or host name of the server that you are on to this particular SRVIPADDR.

SSL_KEYRING=*keyring*
The name of the z/OS or OS/390 client's key ring used in SSL processing. This key ring must reside in RACF.

Example: SSL_KEYRING=IVPRING

SYS_DB2_SUB_SYSTEM_NAME=*NAME*
The DB2 name used by Daemon and System Management servers to connect to the database. Use either the DB2 subsystem name or group attachment name. The default is DB2. If the default is not correct for your installation, change the environment variable to match the correct value.

Example: SYS_DB2_SUB_SYSTEM_NAME=DB21

TRACEALL=*n*

    Specifies the default tracing level for WebSphere for z/OS. Valid values and their meanings are:

0   No tracing

1   Exception tracing, the default

2   Basic and exception tracing

3   Detailed tracing, including basic and exception tracing

    Use this variable in conjunction with the TRACEBASIC and TRACEDETAIL environment variables to set tracing levels for WebSphere for z/OS subcomponents. Do not change this variable unless directed by IBM service personnel.

    Example: `TRACEALL=1`

TRACEBASIC=*n* **| (***n*,...**)**

    Specifies tracing overrides for particular WebSphere for z/OS subcomponents. Subcomponents, specified by numbers, receive basic and exception traces. If you specify more than one subcomponent, use parentheses and separate the numbers with commas. Contact IBM service for the subcomponent numbers and their meanings. Other parts of WebSphere for z/OS receive tracing as specified on the TRACEALL environment variable. Do not change TRACEBASIC unless directed by IBM service personnel.

    Example: `TRACEBASIC=3`

TRACEBUFFCOUNT=*n*

    Specifies the number of trace buffers to allocate. Valid values are 4 through 8. The default is 4.

TRACEBUFFLOC=SYSPRINT | BUFFER

    Specifies where you want trace records to go: either to sysprint (SYSPRINT) or to a memory buffer (BUFFER), then to a CTRACE data set. The default is to direct trace records to sysprint for the client and to a buffer for all other WebSphere for z/OS processes. For servers, you may specify one or both values, separated by a space. For clients, you may specify TRACEBUFFLOC=SYSPRINT only.

    Example: `TRACEBUFFLOC=SYSPRINT BUFFER`

TRACEBUFFSIZE=*n*

    Specifies the size of a single trace buffer in bytes. You can use the letters "K" (for kilobytes) or "M" (for megabytes). Valid values are 128K through 4M. The default is 1M.

TRACEDETAIL=*n* | (*n*,...)

Specifies tracing overrides for particular WebSphere for z/OS subcomponents. Subcomponents, specified by numbers, receive detailed traces. If you specify more than one subcomponent, use parentheses and separate the numbers with commas. Contact IBM service for the subcomponent numbers and their meanings. Other parts of WebSphere for z/OS receive tracing as specified on the TRACEALL environment variable. Do not change TRACEDETAIL unless directed by IBM service personnel.

Examples:

```
TRACEDETAIL=3
```

```
TRACEDETAIL=(3,4)
```

TRACEMINORCODE=*value*

Enables traceback of system exception minor codes. Use only when instructed by IBM Service. Values are:

ALL|all

Enables traceback for all system exception minor codes.

minor_code

Enables traceback for a specific minor code. Specify the code in hex, such as X'C9C21234'.

(null value)

The default. This setting will not cause gathering of a traceback.

Note: Sometimes results depend on the setting of another environment variable, RAS_MINORCODEDEFAULT. If you code TRACEMINORCODE=ALL and RAS_MINORCODEDEFAULT=NODIAGNOSTICDATA, you get a traceback. But, if you code TRACEMINORCODE=(null value) and RAS_MINORCODEDEFAULT=TRACEBACK you also get a traceback. So, specifying TRACEMINORCODE=(null value) does not cancel TRACEBACK; it simply does not cause a TRACEBACK to be gathered.

TRACEPARM=*SUFFIX* | *MEMBER_NAME*

Identifies the CTRACE PARMLIB member. The value can be either a two-character suffix, which is added to the string CTIBBO to form the name of the PARMLIB member, or the fully-specified name of the PARMLIB member. For example, you could use the suffix "01", which the system resolves to "CTIBBO01". A fully-specified name must conform to the naming requirements for a CTRACE PARMLIB member. For details, see *z/OS MVS Diagnosis: Tools and Service Aids*, GA22-7589.

The default value is 00.

If this environment variable is specified and the PARMLIB member is not found, the default PARMLIB member, CTIBBO00, is used. If neither the specified nor the default PARMLIB member is found, tracing is defined to CTRACE, but there is no connection to a CTRACE external writer. For details on the PARMLIB member and the use of the CTRACE external writer, see *WebSphere Application Server V4.0.1 for z/OS and OS/390: Messages and Diagnosis*, GA22-7837.

Note that the Daemon Server is the only server that recognizes this environment variable.

Example: TRACEPARM=01

Appendix B. Migrating from WebSphere for z/OS V4.0 to V4.0.1 without the customization dialog

This topic has procedures for migrating to V4.0.1 without using the customization dialog. Base your choice of which procedure to use on the following table:

If you are migrating to WebSphere for z/OS V4.0.1 and . . .	Then follow . . .	Notes
You can interrupt service to clients (either on a monoplex or a sysplex)	"Steps for performing a warm start from WebSphere for z/OS V4.0 to V4.0.1 with a system or sysplex-wide restart and without the customization dialog"	
You cannot interrupt service to clients	"Steps for performing a rolling warm start from WebSphere for z/OS V4.0 to V4.0.1 without the customization dialog" on page 340	A rolling warm start requires the dual HFS structure described in "Overview of creating the proper HFS structure for upgrades" on page 285.

Steps for performing a warm start from WebSphere for z/OS V4.0 to V4.0.1 with a system or sysplex-wide restart and without the customization dialog

Before you begin:

Requirement: Your V4.0 system must have the proper level of service installed. See "Overall migration tasks to go from WebSphere for z/OS V4.0 to WebSphere for z/OS V4.0.1" in *WebSphere Application Server V4.0.1 for z/OS and OS/390: Migration*, GA22-7860 and consult the PSP bucket for the latest service information.

You must be prepared to stop WebSphere for z/OS. If you have WebSphere for z/OS running in a sysplex as a host cluster, this procedure has you shut down the entire host cluster.

If you have WebSphere for z/OS running in a sysplex as a host cluster and want to maintain service to your clients during the warm start, see "Steps for performing a rolling warm start from WebSphere for z/OS V4.0 to V4.0.1 without the customization dialog" on page 340.

Perform the following steps to do the warm start.

1. Use SMP/E to install the new WebSphere for z/OS code. Be sure to install the new WebSphere for z/OS code into a separate set of MVS and HFS data sets.

   _____

2. Back up your current system. This includes:
   - The system management database
   - The LDAP database tables containing the naming space and the interface repository
   - Files in the HFS containing WebSphere for z/OS run-time information (usually mounted at `/WebSphere390/CB390`).
   - WebSphere for z/OS PROCLIBs
   - WebSphere for z/OS LOADLIBs

   For more information, see "Guidelines for backup of the WebSphere for z/OS system" on page 207.

   _____

3. Edit the BBOMCFG, BBOMPAT2, BBOBIND, BBOIVPP, BBOIBN, and BBOWCMIG members in the new SBBOJCL data set according to comments in the members.

   _____

4. Stop all application servers and WebSphere for z/OS (on a sysplex, stop all clustered host instances).

   _____

5. Unmount the V4.0 HFSes and mount the WebSphere for z/OS V4.0.1 HFSes. The following are the defaults:
   - `/usr/lpp/WebSphere`
   - `/usr/lpp/java/IBM/J1.3`

   _____

6. If not already authorized, APF-authorize the following data sets:
   - *hlq*.SBBOLPA
   - *hlq*.SBBOLOAD
   - *hlq*.SBBOLD2

   where *hlq* is the high-level qualifier for the V4.0.1 data sets.

7. On either the monoplex or one system in the sysplex, do the following:
   a. Submit the following jobs in this order:
      - BBOMCFG
      - BBOMPAT2

- BBOBIND
- BBOIVPP
- BBOIBN

b. Either re-catalog your system PROCLIB or modify your server start procedures, PROGxx, and link list to point to the data sets with the new code.

c. 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.

d. Start the Daemon and application servers.

_____

8. If you are running on a sysplex, for each system, one at a time, do the following:

a. Either re-catalog your system PROCLIB or modify your server start procedures, PROGxx, and link list to point to the data sets with the new code.

b. 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.

c. Start the Daemon and application servers.

Result: When all run-time and application servers throughout the monoplex or sysplex have been restarted, you receive messages that the servers are ready for a warm start.

```
BBOU0579I CB SERIES SERVER server IS READY FOR WARMSTART.
```

where *server* is the name of the server.

Tips:
- There is no time limit set when you must warm start WebSphere for z/OS.
- The Operations application flags servers ready for warm start with a green bullet.

_____

9. Download and install the new level of the Administration application (SM EUI).

Notes:
a. The new level of the Administration application is designed to function within a sysplex in which some WebSphere for z/OS systems have been warm-started and some have not.

b. During the first connection between the Administration application and the system management server, each exchanges its code level

information and the Administration application adjusts its processing accordingly. Administration application messages indicate potential mismatches.

c. In a sysplex, the WebSphere for z/OS may be reconnected from one system management server instance to another one during the warm start phase. The reconnection may cause a functional level switch. If this happens, you will receive a message requesting an explicit reconnection. Because the functional level of the Administration application depends on the functional level of the system management server to which it is connected, new functions may become visible or invisible during functional level switches. For this reason, start using the new functions only after you have warm-started all systems successfully.

_____

10. When all servers are ready for warm start, on either the monoplex or each system in the sysplex, one at a time, do the following:

a. Stop the application servers and the Daemon.

b. Start the Daemon with the warm start option:

    s bbodmn,srvname='...',parms='-ORBCBI WARM'

c. Start your application servers with the warm start option:

    s *server_proc*,srvname='...',parms='-ORBCBI WARM'

where *server_proc* is the application server start procedure.

You can also do the warm start for the application servers through the Operations application.

_____

11. Run the BBOWCMIG job.

_____

12. Re-run the installation verification programs. See "Running the WebSphere for z/OS installation verification programs (IVPs)" on page 192.

_____

You are done when WebSphere for z/OS and the installation verification programs run successfully.

Steps for performing a rolling warm start from WebSphere for z/OS V4.0 to V4.0.1 without the customization dialog

Before you begin:

Requirement: Your V4.0 system must have the proper level of service installed. See "Overall migration tasks to go from WebSphere for z/OS V4.0 to WebSphere for z/OS V4.0.1" in *WebSphere Application Server V4.0.1 for z/OS and OS/390: Migration*, GA22-7860 and consult the PSP bucket for the latest service information.

You must have an HFS structure as described in "Recommendation for the HFS structure" on page 223 and "Overview of creating the proper HFS structure for upgrades" on page 285.

Perform the following steps to do the warm start.

1. Use SMP/E to install the new WebSphere for z/OS code. Be sure to install the new WebSphere for z/OS code into a separate set of MVS and HFS data sets.

   _____

2. Back up your current system. This includes:
   - The system management database
   - The LDAP database tables containing the naming space and the interface repository
   - Files in the HFS containing WebSphere for z/OS run-time information (usually mounted at /WebSphere390/CB390).
   - WebSphere for z/OS PROCLIBs
   - WebSphere for z/OS LOADLIBs

   For more information, see "Guidelines for backup of the WebSphere for z/OS system" on page 207.

   _____

3. Set up your version-specific HFS for the new level of code.

   _____

4. Edit the BBOMCFG, BBOMPAT2, BBOBIND, BBOIVPP, BBOIBN, and BBOWCMIG members in the new SBBOJCL data set according to comments in the members.

   _____

5. If not already authorized, APF-authorize the following data sets:
   - *hlq*.SBBOLPA
   - *hlq*.SBBOLOAD
   - *hlq*.SBBOLD2

   where *hlq* is the high-level qualifier for the V4.0.1 data sets.

   _____

6. Select a clustered host instance to begin the warm start process. On that clustered host instance:

   a. Stop the application servers and the WebSphere for z/OS Daemon.

   b. Switch the HFS that your system references with the SETOMVS command. Use the command to change the $VERSION symbolic.

      Example: Previously, the $VERSION symbolic was VersionA for all systems in the sysplex. Through the use of a built-in symbolic link, references to /usr resolved to VersionA/usr. To switch the HFS that this system references to, say, VersionB, issue:

      ```
      setomvs version=VersionB
      ```

   c. Submit the following jobs in this order:
      - BBOMCFG
      - BBOMPAT2
      - BBOBIND
      - BBOIVPP
      - BBOIBN

   d. Either re-catalog your system PROCLIB or modify your server start procedures, PROGxx, and link list to point to the data sets with the new code.

   e. 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.

   f. Start the Daemon and application servers.

   _____

7. For each of the remaining clustered host instance, one at a time, do the following:

   a. Stop the application servers and the WebSphere for z/OS Daemon.

   b. Either re-catalog your system PROCLIB or modify your server start procedures, PROGxx, and link list to point to the data sets with the new code.

   c. 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.

   d. Switch the HFS that your system references with the SETOMVS command. Use the command to change the $VERSION symbolic.

      Example: Previously, the $VERSION symbolic was VersionA for all systems in the sysplex. Through the use of a built-in symbolic link, references to /usr resolved to VersionA/usr. To switch the HFS that this system references to, say, VersionB, issue:

      ```
      setomvs version=VersionB
      ```

   e. Start the Daemon and application servers.

Result: When all run-time and application servers throughout the sysplex have been restarted, you receive messages that the servers are ready for a warm start.

```
BBOU0579I CB SERIES SERVER server IS READY FOR WARMSTART.
```

where *server* is the name of the server.

Tips:
- There is no time limit set when you must warm start WebSphere for z/OS.
- The Operations application flags servers ready for warm start with a green bullet.

8. Download and install the new level of the Administration application (SM EUI).

Notes:

a. The new level of the Administration application is designed to function within a sysplex in which some WebSphere for z/OS systems have been warm-started and some have not.

b. During the first connection between the Administration application and the system management server, each exchanges its code level information and the Administration application adjusts its processing accordingly. Administration application messages indicate potential mismatches.

c. In a sysplex, the WebSphere for z/OS may be reconnected from one system management server instance to another one during the warm start phase. The reconnection may cause a functional level switch. If this happens, you will receive a message requesting an explicit reconnection. Because the functional level of the Administration application depends on the functional level of the system management server to which it is connected, new functions may become visible or invisible during functional level switches. For this reason, start using the new functions only after you have warm-started all systems successfully.

9. On each system in the sysplex do the following, one at a time:

a. Stop the application servers and the Daemon.

b. Start the Daemon with the warm start option:

```
s bbodmn,srvname='...',parms='-ORBCBI WARM'
```

c. Start your application servers with the warm start option:

```
s server_proc,srvname='...',parms='-ORBCBI WARM'
```

where *server_proc* is the application server start procedure.

You can also do the warm start for the application servers through the
Operations application.

_____

10. Run the BBOWCMIG job.

_____

11. Re-run the installation verification programs. See "Running the
    WebSphere for z/OS installation verification programs (IVPs)" on
    page 192.

_____

You are done when WebSphere for z/OS and the installation verification
programs run successfully.

Appendix C. Using an alternate HFS structure for product upgrades

The HFS structure is key to using the rolling upgrade method. See "Chapter 6. Installing new releases and maintenance levels of WebSphere for z/OS" on page 285 for more information about the rolling upgrade method and IBM's recommended HFS structure used for upgrading WebSphere for z/OS. This topic introduces an alternate HFS structure and appropriate procedures.

Overview of creating the alternate HFS structure for upgrades

The alternate HFS structure does not mount product HFSes directly off the version-specific subdirectories (referenced by the $VERSION symbolic). 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 18.
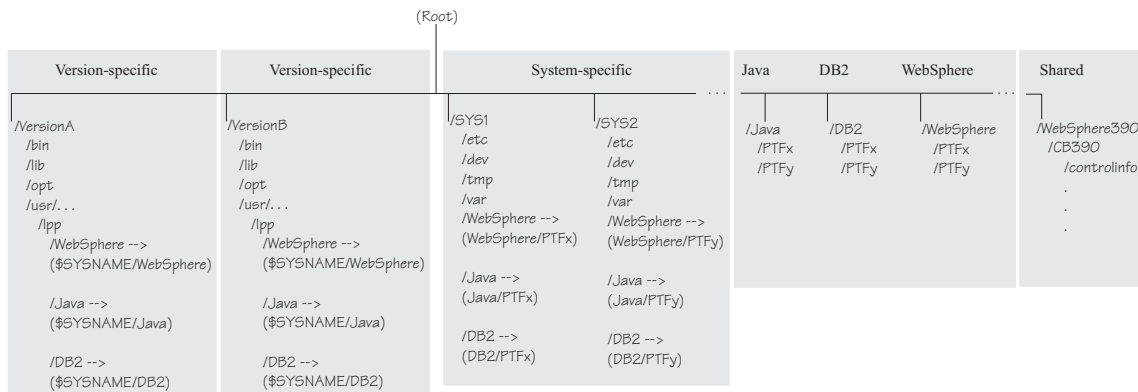
Figure 18. Alternate HFS structure

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.

Procedures for upgrading WebSphere for z/OS code using the alternate HFS structure

This topic covers the procedures for doing a warm start and a hot start with the alternate HFS structure.

Steps for performing a rolling warm start from WebSphere for z/OS V4.0 to V4.0.1 with the alternate HFS structure

Before you begin: You must have an HFS structure as described in "Overview of creating the alternate HFS structure for upgrades" on page 345.

Requirement: Your V4.0 system must have the proper level of service installed. See "Overall migration tasks to go from WebSphere for z/OS V4.0 to

WebSphere for z/OS V4.0.1" in *WebSphere Application Server V4.0.1 for z/OS and OS/390: Migration*, GA22-7860 and consult the PSP bucket for the latest service information.

Perform the following steps to do the warm start.

1. Use SMP/E to install the new WebSphere for z/OS code. Be sure to install the new WebSphere for z/OS code into a separate set of MVS and HFS data sets.

   _____

2. Back up your current system. This includes:
   - The system management database
   - The LDAP database tables containing the naming space and the interface repository
   - Files in the HFS containing WebSphere for z/OS run-time information (usually mounted at /WebSphere390/CB390).
   - WebSphere for z/OS PROCLIBs
   - WebSphere for z/OS LOADLIBs

   For more information, see "Guidelines for backup of the WebSphere for z/OS system" on page 207.

   _____

3. Set up new directories for the V4.0.1 level of code for WebSphere for z/OS, Java, and DB2 (JDBC).

   _____

4. Edit the BBOMCFG, BBOMPAT2, BBOBIND, BBOIVPP, BBOIBN, and BBOWCMIG members in the new SBBOJCL data set according to comments in the members.

   _____

5. If not already authorized, APF-authorize the following data sets:
   - *hlq*.SBBOLPA
   - *hlq*.SBBOLOAD
   - *hlq*.SBBOLD2

   where *hlq* is the high-level qualifier for the V4.0.1 data sets.

   _____

6. Select a clustered host instance to begin the warm start process. On that clustered host instance:
   a. Stop the application servers and the WebSphere for z/OS Daemon.
   b. Change the symbolic references for WebSphere for z/OS, Java, and DB2 (JDBC) for the system.

c. Submit the following jobs in this order:
   - BBOMCFG
   - BBOMPAT2
   - BBOBIND
   - BBOIVPP
   - BBOIBN

d. Either re-catalog your system PROCLIB or modify your server start procedures, PROGxx, and link list to point to the data sets with the new code.

e. 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.

f. Start the Daemon and application servers.

_____

7. For each of the remaining clustered host instance, one at a time, do the following:

   a. Stop the application servers and the WebSphere for z/OS Daemon.

   b. Either re-catalog your system PROCLIB or modify your server start procedures, PROGxx, and link list to point to the data sets with the new code.

   c. 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.

   d. Change the symbolic references for WebSphere for z/OS, Java, and DB2 (JDBC) for the system.

   e. Start the Daemon and application servers.

      Result: When all run-time and application servers throughout the sysplex have been restarted, you receive messages that the servers are ready for a warm start.

      ```
      BBOU0579I CB SERIES SERVER server IS READY FOR WARMSTART.
      ```

      where *server* is the name of the server.

      Tips:
      - There is no time limit set when you must warm start WebSphere for z/OS.
      - The Operations application flags servers ready for warm start with a green bullet.

_____

8. Download and install the new level of the Administration application (SM EUI).

Notes:

a. The new level of the Administration application is designed to function within a sysplex in which some WebSphere for z/OS systems have been warm-started and some have not.

b. During the first connection between the Administration application and the system management server, each exchanges its code level information and the Administration application adjusts its processing accordingly. Administration application messages indicate potential mismatches.

c. In a sysplex, the WebSphere for z/OS may be reconnected from one system management server instance to another one during the warm start phase. The reconnection may cause a functional level switch. If this happens, you will receive a message requesting an explicit reconnection. Because the functional level of the Administration application depends on the functional level of the system management server to which it is connected, new functions may become visible or invisible during functional level switches. For this reason, start using the new functions only after you have warm-started all systems successfully.

_____

9. On each system in the sysplex do the following, one at a time:

a. Stop the application servers and the Daemon.

b. Start the Daemon with the warm start option:

```
s bbodmn,srvname='...',parms='-ORBCBI WARM'
```

c. Start your application servers with the warm start option:

```
s server_proc,srvname='...',parms='-ORBCBI WARM'
```

where *server_proc* is the application server start procedure.

You can also do the warm start for the application servers through the Operations application.

_____

10. Run the BBOWCMIG job.

_____

11. Re-run the installation verification programs. See "Running the WebSphere for z/OS installation verification programs (IVPs)" on page 192.

_____

You are done when WebSphere for z/OS and all your application servers are running.

Steps for performing a rolling hot start with the alternate HFS structure

Before you begin: Read "Overview of creating the alternate HFS structure for upgrades" on page 345 to understand the HFS structure you need.

Perform the following steps to do a hot start:

1. Use SMP/E to install the new WebSphere for z/OS code. Be sure to install the new WebSphere for z/OS, Java, and DB2 (JDBC) code into their own HFS data sets.

   Example: Install a service level (PTFx) and copy the code to for WebSphere for z/OS into OMVS.PTFX.WEB.HFS, the code for Java into OMVS.PTFX.JAVA.HFS, and the code for DB2 (JDBC) into OMVS.PTFX.DB2.HFS.

   _____

2. Mount the new data sets under their component mount points.

   Example:
   ```
   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)
   ```

   _____

3. Select a clustered host instance to begin the hot start process. On that clustered host instance:

   a. Stop the application servers and the WebSphere for z/OS Daemon.

   b. Either re-catalog your system PROCLIB or modify your server start procedures, PROGxx, and link list to point to the data sets with the new code.

   c. 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.

   d. Change the symbolic references for WebSphere for z/OS, Java, and DB2 (JDBC) for the system.

   e. Start the Daemon and application servers.

   _____

4. Repeat step 3 for each remaining clustered host instance, one at a time.

   _____

You are done when you have completed the hot start on each system in the sysplex.

Appendix D. Configuring the name space

During system installation and configuration, configure the name space using a special naming configuration file. This file is specified on the NCONFIG DD statement in the naming client start procedure (BBONMC). IBM supplies a sample naming configuration file, called SBBOEXEC(BBOCNFG), that you can modify. This topic explains the syntax for naming configuration files.

The naming configuration file contains the following information:
- The location of a currently existing inter-domain root (IDR) or an indicator that says to create it locally.
- The name of hosts that contain cells to be bound to the IDR and the names of those cells. This identifies any non-WebSphere for z/OS hosts that may have already been configured and that house cell name space segments that should be made visible under the IDR. WebSphere for z/OS will traverse from the local root naming context of the specified host to its primary parent cell and bind that cell into the IDR using the supplied name.
- The names of cells to be created on this WebSphere for z/OS host.
- The names of workgroups to be created on this WebSphere for z/OS host along with the name of the primary and alternate cell relative to the IDR.
- The name of the host segment in the single local to be created on this host. The names of that local's primary and alternate parent workgroups and cells will also be provided relative to the IDR.

Note: Currently, z/OS or OS/390 LDAP supports a maximum distinguished name size of 1000 characters. If the name of an object or context binding exceeds that limit, the system issues an InvalidName exception. This may happen even if what you specify is much shorter than 1000 bytes because a name is mapped onto a significantly longer internal LDAP name. For example, if you specify

```
a/b/c
```

LDAP creates the following distinguished names:

```
TypelessRDN=c,TypelessRDN=b,TypelessRDN=a,TypelessRDN=/,o=BOSS,c=US
TypelessRDN=c,TypelessRDN=b,TypelessRDN=a,TypelessRDN=/,o=WASNaming,c=US
```

The syntax of the naming configuration file uses stanzas as follows:

```
[NamingIDR]                    // Cells that currently exist on other
                               // machines that should be bound under
                               // the WebSphere for z/OS IDR.

IDRLocation=host:port          // Specifies the location of a remote host
```

```
                                        // where the IDR lives or 'local' if we
                                        // create one here.

RemoteHost1=host:port                   // Remote host where a cell lives

RemoteCell1=cell                        // Name of that remote cell when bound
                                        // under IDR.

RemoteMemberHost1.1=host:port           // Bind remote host belonging to RemoteCell1
                                        // into the NameSpace
RemoteHost2=host:port

RemoteCell2=cell

⋮


[Cells]                                 // Names of new cells to create on this
                                        // machine and bind to IDR.

Cell1=cell
Cell2=cell

⋮


[Workgroups]                            // Names of new workgroups to create on this
                                        // machine and the name of the cells to
                                        // bind them to.

WorkGroup1=workgroup                    // Name for this new workgroup.
PrimaryCell1=cell                       // Primary cell bound to this workgroup.
AlternateCell1.1=cell                   // Alternate cell bound to this workgroup.

Workgroup2=workgroup
PrimaryCell1=cell
AlternateCell1.1=cell
AlternateCell1.2=cell

⋮


[Hosts]                                 // Locals to create on this machine
                                        // identified by their host name.  Also
                                        // specifies the name of the workgroup
                                        // and cells to bind the host under.

Host1=host|&DAEMON_IPNAME.              // Either the host name or variable for
                                        // the Daemon IP Name

PrimaryCell1=cell
AlternateCell1=cell
PrimaryWorkgroup1=workgroup
AlternateWorkgroup1.1=workgroup
```

The first stanza, NamingIDR, provides information that will allow the naming configuration to add any previously existing cells to the IDR. The IDR is supported on WebSphere for z/OS only. Thus, cells created on Component Broker for Windows NT must be specified in this way if they are to be visible from the IDR.

The IDR Location variable in the NamingIDR stanza indicates either to build the IDR locally or provide the location of a currently existing IDR. If the IDR is to be built locally, then specify IDRLocation=local. If a currently existing IDR is to be used, then specify a host name and port. The naming configuration utility will bootstrap to this host and navigate to the IDR to obtain its reference.

The RemoteHostn variable in the NamingIDR stanza is used to specify the host name and port number of a host whose primary cell should be visible under the IDR. Naming configuration processing will bootstrap to the specified host and resolve from that host's local root naming context to obtain the cell naming context.

Multiple remote hosts can be specified in the NamingIDR stanza. Each host is identified by the postfix modifier n on the RemoteHostn variable. The modifiers used should begin at 1 and be numbered sequentially for the multiple remote hosts specified. The RemoteCelln variable supplies the name relative to the IDR for the corresponding remote cell.

The RemoteMemberHost$n.n$ binds remote hosts belonging to a RemoteCell into the NameSpace. A link from the host to the IDR is created (that is, the global IDR context is bound into the host's root context under the name "...", thus allowing users to navigate directly from their local host into the IDR, and thus into the entire federated name space). There should be a RemoteMemberHost statement for each host belonging to the cell being handled.

The Cells stanza specifies the names relative to the IDR of new cells to be created on this host. The Celln variable specifies the name using the same postfix notation as used previously.

The Workgroups stanza specifies the name of new workgroups to create on this host via the WorkGroupn variable. The primary and alternate cells under which to bind each new workgroup must be specified as well. A single primary cell is specified on the PrimaryCelln where n identifies the workgroup postfix. Multiple alternate cells are specified via the AlternateCellnz variable where n identifies the workgroup postfix and z is the alternate cell in the case of the workgroup stanza with respect to the name space structure. However, the new workgroup must be successfully bound with the primary cell in order for the build to be considered successful.

The Hosts stanza is used to guide the creation of the local name space segment on the current system. A single local name space segment must be built per system in the current release of WebSphere for z/OS. However, multiple local segments may be allowed in a future release. The name of the host portion of the new local name space segment is specified via the Host*n* variable, where *n* must be 1 in the current release (more hosts specifications in the file are tolerated—they are simply ignored). The names of primary and alternate cells and workgroups must also be specified.

Alternately, instead of Host*n*, use the variable &DAEMON_IPNAME. The variable name must be in uppercase letters, and it must be terminated with a period. The option is relevant if you set up federated name spaces, in which case the host names of the systems involved must be different. This variable allows you to change the local host name in the file, without modifying the file, when moving it across sysplexes.

There is a distinction between primary and alternate in the case of the local name space segment with respect to name space structure. The primary cell and primary workgroup can be resolved relative to the local root name context via cell and workgroup respectively. The primary cell and workgroup can also resolve down to the host. Alternate workgroup and cells also contain pointers down to the host. The distinction is that the host contains no direct pointers to the alternate cells and workgroups.

The primary and alternate cells for the host are specified on the PrimaryCelln and AlternateCellnz variables in the same manner as that for the Workgroup stanza. The names of primary and alternate workgroups are specified relative to the IDR on the PrimaryWorkgroupn and AlternateWorkgroupnz variables.

In the current release of WebSphere for z/OS, it is possible to run the naming configuration utility multiple times with different naming configuration files to build additional name space segments. Additional alternate segments can also be added. For example, a workgroup can be made to point to an additional alternate cell. However, it will not be possible to delete name space segments or modify their primary parents.

When subsequently running the naming configuration utility to build additional segments, it is permissible to simply update an existing configuration file. Any currently existing segments will be flagged with informational messages.

Scenarios

These scenarios show some of the configuration possibilities.

Scenario 1

A single, local workgroup and cell will be built on WebSphere for z/OS. One or more Component Broker for Windows NT hosts will build a local that is bound into the WebSphere for z/OS name space as an alternate. In Component Broker for Windows NT, the primary workgroup and cell must be on the Component Broker for Windows NT machine. A WebSphere for z/OS can be bound in as an alternate. The steps are:

1. The activities must begin with WebSphere for z/OS. A WebSphere for z/OS configuration file is created. The NamingIDR stanza is empty in this case. The remaining stanzas describe the name space to be built in WebSphere for z/OS. Because WebSphere for z/OS is the first host being configured, the parents of name space segments built must also reside on this WebSphere for z/OS host. All connections between the various segments are added as required.

2. Component Broker for Windows NT uses an administrative interface that allows the required alternate members of links between name space segments to be added. The administrator would need to define the following links:

   a. Link from local to workgroup

   b. Link from local to cell

   c. Link from cell to host

   d. Link from workgroup to host

Scenario 2

In this scenario, local, workgroup, and cell name space segments will be created on a Component Broker for Windows NT system. A WebSphere for z/OS local will be created and it will be bound into the Component Broker for Windows NT workgroup and cell. The steps are:

1. Configure Component Broker for Windows NT as is done today.

2. Create a WebSphere for z/OS configuration file. This configuration file will have an entry in the NamingIDR stanza to bind the Component Broker for Windows NT cell under the WebSphere for z/OS IDR. The Workgroups and Cells stanzas of the WebSphere for z/OS configuration file would be empty. The Hosts stanza would specify the names relative to the IDR of parent workgroups and cells in the same manner as previous examples.

Scenario 3

In this scenario, a local, workgroup, and cell segment is created in both the Component Broker for Windows NT and WebSphere for z/OS name servers. However, later we want to come back and add a new workgroup to the WebSphere for z/OS that resides under the Component Broker for Windows NT cell. The steps are:

1. Start with WebSphere for z/OS. Build the WebSphere for z/OS name space segments as in "Scenario 1".

2. Build the Component Broker for Windows NT name space segments as in "Scenario 2" on page 357.

3. A cell was just created on Component Broker for Windows NT host. Since Component Broker for Windows NT has no awareness of the IDR, its cell must now be bound to the IDR so that it can be visible during future configuration activities. A second WebSphere for z/OS is created. This configuration file contains only the NamingIDR stanza to identify the Component Broker for Windows NT cell to be bound to the IDR. The naming configuration utility is then run again to bind the Component Broker for Windows NT cell to the IDR.

4. Sometime later, the new workgroup is created and bound to the Component Broker for Windows NT cell. A third WebSphere for z/OS naming configuration file is created and specifies only the Workgroups stanza to identify the information for the new workgroup. This information can be specified as usual, since the Component Broker for Windows NT cell is bound to the WebSphere for z/OS IDR.

Appendix E. Setting up DCE

This topic explains WebSphere for z/OS's use of DCE security, guidelines and requirements for this support, and instructions about setting up DCE security for z/OS or OS/390 clients and servers. For information about DCE and Component Broker for Windows NT, consult *WebSphere Application Server Enterprise Edition Component Broker System Administration Guide*.

Overview of WebSphere for z/OS and DCE

On z/OS or OS/390, the DCE Security Server is a component of the z/OS or OS/390 Security Server, which is an optional feature of z/OS or OS/390. RACF is another component in the z/OS or OS/390 Security Server, but you do not need to operate it with the DCE Security Server—you may use another security product, provided it can translate a DCE account's principal into a z/OS or OS/390 user ID (and vice versa) and can operate with the System Authorization Facility (SAF) interface. Whenever we cite RACF, you may substitute another security product that interoperates with the DCE Security Server.

Through DCE, WebSphere for z/OS supports CORBA standards for security. Whenever work requests come into or go out of the system (that is, the work request is remote), WebSphere for z/OS uses DCE security, provided it is called for, and maps the DCE account's principal to its corresponding z/OS or OS/390 user ID or vice versa.

DCE implements a form of the Kerberos security model where both clients and servers trust the security server but not each other. The security server operates as a third-party authenticator so that a client and a server can establish trust to effectively interoperate.

WebSphere for z/OS supports three quality-of-protection types through DCE: no protection (that is, two-way—mutual— authentication), message integrity, and message confidentiality (encryption). DCE quality-of-protection options for out-of-order messages and no-message-replay are not supported. In addition to the basic DCE support, message confidentiality requires you to implement the Data Encryption Standard (DES) feature in the DCE Security Server and DCE Base Services.

z/OS or OS/390 client quality of protection is enabled through the CLIENT_DCE_QOP environment variable (see "Appendix A. Environment files" on page 303). Server quality of protection is enabled by setting an attribute with the Administration application.

Important characteristics of WebSphere for z/OS support for DCE are:

- Server control regions, local clients, and remote clients participating in DCE security must be configured in the same DCE cell.

  Note: If a WebSphere for z/OS entity is going to use an *unauthenticated* transaction, that entity need not be in a DCE cell or could be in another DCE cell, but it cannot use WebSphere for z/OS with DCE security.

- Each z/OS or OS/390 system in the sysplex participating in DCE security must have its own DCE Security Replica Server operating properly within the same DCE cell. This requirement is due to a special DCE-WebSphere for z/OS DLL required by WebSphere for z/OS.

- You must maintain copies of keytab files on each z/OS or OS/390 system HFS where a server control region needs to reference the information in that file.

Guidelines and requirements for configuring DCE for use with WebSphere for z/OS

Implement DCE with WebSphere for z/OS like any other DCE configuration, but follow these guidelines and requirements:

- Familiarize yourself with the following books:
  - *z/OS DCE Planning*
  - *z/OS DCE Configuring and Getting Started*
  - *z/OS DCE Administration Guide*
  - *z/OS DCE Command Reference*
  - *z/OS SecureWay Security Server RACF Security Administrator's Guide*

- Place all WebSphere for z/OS entities (server control regions, local clients, and remote clients) using DCE security into the same DCE cell.

- Create a DCE Security Replica Server on each z/OS or OS/390 system within the same DCE cell.

- For each WebSphere for z/OS system, a DCE Security Server Replica must be running in its own address space named DCESECD.

  Notes:
  1. A DCE Security Replica requires the DCE Base Services environment operating on that system.
  2. The default settings for the DCE Kernel assume a DCE Security Server running in its own address space rather than as part of the Kernel itself.
  3. The Cell Directory Service, if configured, defaults to a separate address space as DCECDSD.

- We strongly recommend that you set up all Security Server Replicas and the Security Server Master on platforms that have high availability. DCE remote clients and DCE administrative functions can be impacted by TCP/IP protocol timeouts when systems in the DCE cell that operate with Security Replica Servers are not available. If a system will not be available for a long period of time, consider deconfiguring the Security Server Replica to avoid server resolution processing delays. You can use environment variables to direct work requests to operating servers and override the normal Cell Directory Service process, but we advise you use this method only in test environments or error recovery processes.
- Set up and maintain keytab files in the HFS for each z/OS or OS/390 system that has servers (control regions) that use DCE security.
- Set up a fully-configured TCP/IP Domain Name Server for DCE. You do not have to put the DNS on z/OS or OS/390.
- To use WebSphere for z/OS message confidentiality quality of protection, install the DCE Base Services and Security Server Replica with the DES Feature of DCE.
- In addition to DCE account establishment, administration, and maintenance, you must match DCE accounts with RACF user IDs. RACF holds some of this information in the resources of the RACF DCE segment definitions that are cross-referenced to the RACF resources in the RACF DCEUUIDS Class. It is this inter-relationship of RACF user IDs and DCE accounts that allows remote Component Broker clients and servers to operate securely using privileges set up for their RACF-mapped user IDs.
- If using RACF, see the RACF interoperability topic in the *z/OS DCE Administration Guide* for information on how to set up RACF to interoperate with DCE. Grant the appropriate RACF authority to the user IDs associated with the server control regions to allow them to resolve DCE account information into RACF user ID privileges. You must define the IRR.RDCERUID profile in the RACF Facility Class and grant the server control region user IDs READ privilege to this profile. Also, activate the DCEUUIDS class.

  We create a customized RACF sample when you run the customization dialog that includes these definitions. See "Running the customization dialog" on page 61.

  Note: If you plan to use DCE with a security product other than RACF, your security product must be able to map a DCE principal to a user ID.

Steps for setting up a server with DCE security

Before you begin: You must have the WebSphere for z/OS run-time server instances and the Administration application installed. See "Chapter 3. Installing and customizing your first run time" on page 55.

Follow the guidelines and requirements for setting up DCE in "Guidelines and requirements for configuring DCE for use with WebSphere for z/OS" on page 360.

Note: Consult *WebSphere Application Server Enterprise Edition Component Broker System Administration Guide* for security information on Windows NT servers.

Perform the following steps to set up a server with DCE security:

1. If you are not creating a new conversation with the Administration application, create a new one. For information about how to start a conversation, see *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838.

   _____

2. Select or create the server that you want to make secure with DCE.

   _____

3. In the properties form, select the DCE allowed check box. Depending on whether you want other forms of security, select other check boxes.

   _____

4. In the properties form, select the type of DCE quality of protection you want. Types are no protection (that is, two-way—mutual— authentication), message integrity, and message confidentiality (encryption).

   _____

5. Enter the keytab file. The default is `/krb5/v5srvtab`.

   _____

6. In the security preference table, set DCE to 1. Depending on whether you want other forms of security, set the preferences for them.

   _____

7. Complete any other definitions in your conversation, then validate, commit, and activate the conversation.

   _____

You know you are done when the conversation is successfully activated.

Steps for setting up a z/OS or OS/390 client with DCE security

Before you begin: Follow the guidelines and requirements for setting up DCE in "Guidelines and requirements for configuring DCE for use with WebSphere for z/OS" on page 360.

Note: Consult *WebSphere Application Server Enterprise Edition Component Broker System Administration Guide* for security information on Windows NT clients.

Perform the following steps to set up a z/OS or OS/390 client with DCE security:

1. Map the DCE principal associated with the client to a z/OS or OS/390 user ID.

   _____

2. In your environment file, set the environment variable CLIENT_DCE_QOP. If not set, the default is NO_PROTECTION. See the description of this environment variable in "Appendix A. Environment files" on page 303.

   _____

3. In your environment file, set the environment variable RESOLVE_IPNAME to the host system to which the z/OS or OS/390 client will communicate.

   _____

4. Save the environment file.

   _____

You know you are done when the z/OS or OS/390 client successfully connects to the server using DCE security.

Appendix F. 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	OS/390
CICS	RACF
DB2	RETAIN
DFSMS	RMF
ES/3090	RS/6000
ES/4381	S/390
ES/9000	S/390 Parallel Enterprise Server
ESA/390	SecureWay
IBM	System/390
IMS	VisualAge
IMS/ESA	VTAM
Language Environment	WebSphere
Multiprise	z/OS
Open Class	

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.

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:

- Sun Microsystems Glossary of Java Technology-Related Terms, located on the Internet at:

  `http://java.sun.com/docs/glossary.html`

- *IBM Glossary of Computing Terms*, located on the Internet at:

  `http://www.ibm.com/ibm/terminology/`

- The Sun Web site, located on the Internet at:

  `http://www.sun.com/`

Index

IBM ®

Program Number: 5655–F31

Printed in the United States of America