WebSphere Application Server for z/OS V5

Operations and Administration

SA22-7912-00

WebSphere Application Server for z/OS V5

Operations and Administration

SA22-7912-00

Note

Before using this information and the product it supports, be sure to read the general information under Appendix C, "Notices", on page 151.

First edition (May 2003)

This edition applies to WebSphere Application Server for z/OS V5 (5655-F31), and to all subsequent releases and modifications until otherwise indicated in new editions.

The most current versions of the WebSphere Application Server for z/OS V5 publications are at this Web site:

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

Contents

Figures

Tables

About this book

This book describes operations and administration procedures for WebSphere for z/OS.

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

Who should read this book

This book is for WebSphere for z/OS system operators and administrators. Practical experience using WebSphere for z/OS, OE, RRS, and WLM is recommended, but not essential. To get familiar with WebSphere for z/OS, the operator or administrator should first read the Administrative Model under the ″Must Read″ label in the ″What's New″ section of WebSphere Application Server for z/OS V5.0 InfoCenter, which gives an overview of system administration using the adminstration console, the WebSphere administrative scripting program, and the command line tools. You can access the InfoCenter through the WebSphere for z/OS library Web site at `http://www.ibm.com/softrware/webservers/appserv/zos_os390/library.html`. You can also visit the WebSphere Application Server Web site at `http://www.ibm.com/software/webservers/appserv/zos_os390/` for related information and publications.

How this book is organized

The following is an overview of the chapter order and contents:

- Chapter 1, "Introduction", on page 1 is an overview of WebSphere for z/OS operations and administration.
- Chapter 2, "Identifying where to perform WebSphere for z/OS operations", on page 5 lists common operations tasks and describes when to use the administrative console or the MVS console to perform these tasks.
- Chapter 3, "Operating WebSphere for z/OS", on page 11 describes basic WebSphere for z/OS operations tasks.
- Chapter 4, "Operational considerations for z/OS or OS/390 subsystems", on page 47 describes helpful hints for when customizing z/OS or OS/390 subsystems.
- Chapter 5, "WebSphere for z/OS backup guidelines and procedures", on page 49 describes WebSphere for z/OS backup guidelines and procedures.
- Chapter 6, "Monitoring and recovering WebSphere for z/OS and dependent subsystems", on page 51 describes guidelines for monitoring and recovering WebSphere for z/OS and its dependent subsystems.
- Chapter 7, "WebSphere for z/OS administration procedures", on page 63 describes WebSphere for z/OS administration tasks.
- Chapter 8, "Security administration", on page 67 describes security issues within WebSphere for z/OS.
- Chapter 9, "WebSphere for z/OS performance tuning guidelines", on page 69 describes WebSphere for z/OS performance tuning guidelines.
- Chapter 10, "WebSphere for z/OS performance monitoring procedures", on page 101 describes WebSphere for z/OS performance monitoring guidelines.

- Chapter 11, “Systems Management Facility (SMF) recording and monitoring”, on page 111 describes Systems Management Facility (SMF) recording and monitoring for WebSphere for z/OS.
- “SMF record type 120 (WebSphere for z/OS)” on page 121 describes Systems Management Facility (SMF) record type 120 for WebSphere for z/OS.
- Appendix B, “Naming conventions for application servers”, on page 147 describes how to establish a naming convention for your application servers.
- Appendix C, “Notices”, on page 151 provides notices about programming interfaces, examples used in this book, and trademarks.

Where to find related information, tools, and supplements

Most of the information about WebSphere for z/OS appears in task-oriented articles in the WebSphere Application Server for z/OS V5 InfoCenter, which you can access through the WebSphere for z/OS library Web site:

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

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

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

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

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

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

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

How to send your comments

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

- To send comments on articles in the WebSphere Application Server for z/OS V5 InfoCenter:

1. Display the article in your Web browser and scroll to the end of the article.
2. Click on the `Feedback` label at the bottom of the article, and a separate window containing an e-mail form appears.
3. Fill out the e-mail form as instructed, and click on `Submit feedback`.

- To send comments on PDF books, you can e-mail your comments to:

  `wasdoc@us.ibm.com`

  or fax them to 919-254-0206.

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

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

Chapter 1. Introduction

Managing information technology (IT) for the effective delivery of IT services is a complex business challenge. The demand for high system availability is critical for enterprises seeking to become business leaders.

One key success factor to control complex environments like WebSphere for z/OS is to implement robust monitoring and operations to maximize system availability and performance. Every enterprise should carefully assess its business needs, then determine what is possible with the current technology and the availability of its resources to implement that technology. Very few enterprises can afford long, planned or unscheduled outages. The need for high availability will always be required, and continuous availability will increasingly become a major competitive advantage.

Availability can mean different things to different organizations:

High availability

A system characteristic that minimizes or masks the effects of **unscheduled** outages. It attempts to keep applications running during planned service hours. It involves redundancy of components to ensure that service is always delivered, regardless of component failures. It also involves thorough testing to ensure that potential problems are detected before they affect the production environment.

Continuous operations

A system characteristic that minimizes or masks the effects of **scheduled** outages. It attempts to deliver IT services to customers without outages, planned or otherwise. This is not that difficult to achieve. There are many examples of specialized systems, such as a Communication Management Configuration, which can run for many months without any type of outage. However, this requires few or no changes to the system, which is an unrealistic scenario in an actual production system.

Continuous availability

A system characteristic that minimizes or masks the effects of **all** outages. It is the result of combining high availability and continuous operations. It means that the IT services provided by applications will remain available across scheduled and unscheduled system outages.

Overview of WebSphere for z/OS operations

The WebSphere for z/OS operations application lets you manage WebSphere for z/OS clusters and servers using the administrative console that runs on the Internet. You can display the status of all servers, stop application servers and servers, cancel application servers and servers, cancel and restart clusters and servers, and filter the operations window. For information on how to use the WebSphere for z/OS administrative console, see ″Using the Administrative Console″ in the WebSphere Application Server for z/OS V5.0 InfoCenter, access to which can be obtained through the WebSphere forz/OS library Web site http://www.ibm.com/software/webservers/appserv/zos_os390/library.html.

There are two basic kinds of configurations:

- Base Run-time Configuration

  This configuration includes a location service daemon and one node which includes an Application Server with a controller and any number of servants. In this configuration, the MVS operator starts the controller, which in turn starts the location service daemon. WLM then starts the servant regions.
- Network Deployment (ND) Run-time configuration

  A typical WebSphere for z/OS network depolyment run-time includes two nodes (both housed in a cell, which can span systems)—one for the Deployment Manager and one for the Application Server. Each node has an associated location service daemon also in the cell. The Deployment Manager node includes a Deployment Manager with a controller and any number of servants. The Application Server node includes a node agent, a JMS server, and an Application Server with a controller and any number of servants. In this configuration, the MVS operator starts the node agent which starts the location service daemon. The MVS opererator then starts the Deployment Manager which can then be used to start the Application Server controller regions through the administrative console.

For a more detailed description, refer to *WebSphere Application Server for z/OS V5.0: Installation and Customization*, GA22-7909.

This book, *WebSphere Application Server for z/OS V5.0: Operations and Administration*, SA22-7912, provides guidelines and procedures for operating and administering WebSphere for z/OS. It includes:

- Performing operations tasks from the z/OS or OS/390 console
- Hints and tips for managing servers
- Operations guidelines
- Tuning dependent subsystems to improve system performance
- Recovery scenarios and guidelines
- Monitoring and backup guidelines.

Overview of WebSphere for z/OS administration

Administering WebSphere for z/OS involves the use of both the MVS console and the WebSphere for z/OS administrative console. For example:

- Use MVS commands issued from the MVS console to start the Base AppServer controller region, and the ND Node Agent and Deployment Manager.
- In a base application server configuration, you must start the first server with an MVS operator command. Once the first server is started, you can then use the administrative console, if it has this application, to start other application servers in the node. Once the Deployment Manager and Node Agent are active (in an ND configuration), you can use the administrative console to start and stop application servers.
- WLM starts all servant regions using Address Space Create (ASCRE)

The WebSphere for z/OS administrative console allows you to display and modify WebSphere for z/OS applications and the environment in which they run. For information on how to use the WebSphere for z/OS administrative console, see ″Using the Administrative Console″ in the WebSphere Application Server for z/OS V5.0 InfoCenter, access to which can be obtained through the WebSphere forz/OS library Web site http://www.ibm.com/software/webservers/appserv/zos_os390/library.html. Additional administration tasks and guidelines are in Chapter 7, “WebSphere for z/OS administration procedures”, on page 63 and Chapter 10, “WebSphere for z/OS performance monitoring procedures”, on page 101.

Overview of required WebSphere for z/OS elements and subsystems

For a list of required WebSphere for z/OS elements and subsystems, refer to *WebSphere Application Server for z/OS V5.0: Installation and Customization*, GA22-7909

Chapter 2. Identifying where to perform WebSphere for z/OS operations

This chapter lists the main WebSphere for z/OS operations tasks and directs you to information that helps to perform these tasks. WebSphere for z/OS activities and operations can be performed from:

- A z/OS or OS/390 MVS console (most operations)
- The WebSphere administrative console (some operations)
- TSO or RRS panels (some operations)

For more information on the administrative console, see ″Using the Administrative Console″ in the WebSphere Application Server for z/OS V5.0 InfoCenter, access to which can be obtained through the WebSphere forz/OS library Web site http://www.ibm.com/software/webservers/appserv/zos_os390/library.html..

Table 1. WebSphere for z/OS operations tasks

Task	MVS console	WebSphere Administration Console	TSO panel	Reference to associated procedure
Start operations				
Starting the WebSphere for z/OS environment / location service daemon	Yes	No	No	See “Starting the WebSphere for z/OS environment” on page 12.
Determining if the location service daemon is up	Yes	Yes	No	See “Starting clusters and servers” on page 14.
Starting a cluster or application server	Yes	Application server only	No	See “Starting clusters and servers” on page 14 and ″Using the Administrative Console″ in the WebSphere Application Server for z/OS V5.0 InfoCenter.
Cold starting WebSphere for z/OS	Yes	No	No	See “Cold starting WebSphere for z/OS” on page 15.
Hot starting WebSphere for z/OS	Yes	Application server only	No	See “Hot starting WebSphere for z/OS” on page 15.
Warm starting WebSphere for z/OS	Yes	Application server only	No	See “Warm starting WebSphere for z/OS” on page 15.
Quick starting WebSphere for z/OS	Yes	Application server only	No	See “Quick starting WebSphere for z/OS” on page 15.
Stop operations				
Stopping the WebSphere for z/OS environment / location service daemon	Yes	No	No	See “Stopping the WebSphere for z/OS environment” on page 25.

Table 1. WebSphere for z/OS operations tasks (continued)

Task	MVS console	WebSphere Administration Console	TSO panel	Reference to associated procedure
Stopping a cluster or application server	Yes	Application server only	No	See “Stopping servers and servants from the administrative console” on page 25 and ″Using the Administrative Console″ in the WebSphere Application Server for z/OS V5.0 InfoCenter.
Cancel operations				
Canceling the location service daemon	Yes	No	No	See “Canceling the location service daemon” on page 26.
Canceling a cluster or application server	Yes	Yes	No	See “Steps for stopping the WebSphere for z/OS environment” on page 25.
Display operations				
Displaying the status of ARM-registered address spaces including WebSphere for z/OS clusters and servants	Yes	No	No	See “Displaying the status of ARM-registered address spaces including WebSphere for z/OS servers and server instances” on page 27.
Displaying units of work (threads) for DB2	Yes	No	No	See “Displaying units of work (threads) for DB2” on page 28.
Displaying indoubt units of work (threads) for DB2	Yes	No	No	See “Displaying indoubt units of work (threads) for DB2” on page 28.
Displaying units of work for RRS	No	No	Yes	See “Displaying units of work for RRS” on page 29. See *z/OS MVS Programming: Resource Recovery*, for information on how to display units of work for RRS.
Displaying units of work for CICS	Yes	No	Yes	See “Displaying units of work for CICS” on page 29. See *CICS Operations and Utilities Guide*, SC34-5991, for details on displaying units of work for CICS.
Displaying units of work (transactions) for IMS	Yes	No	No	See “Displaying units of work (transactions) for IMS” on page 29. Also see *IMS/ESA Summary of Operator Commands*, SC26-8766.

Table 1. WebSphere for z/OS operations tasks (continued)

Task	MVS console	WebSphere Administration Console	TSO panel	Reference to associated procedure
Displaying the status of a cluster or servant	Yes	Yes	No	See Chapter 3, “Operating WebSphere for z/OS”, on page 11 and *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838.
Displaying active address spaces	Yes	No	No	See “Displaying active address spaces” on page 28.
Displaying active replies	Yes	No	No	See “Displaying active replies” on page 28.
Modify operations				
Getting help for the modify command	Yes	No	No	See “Getting help for the modify command” on page 36.
Canceling application clusters and servers	Yes	No	No	See “Canceling application clusters and servers with the modify command” on page 36.
Modifying trace variables dynamically	Yes	No	No	See “Modifying trace variables” on page 37.
Modifying the Java trace string	Yes	No	No	See “Modifying the Java trace string” on page 38.
Displaying status	Yes	No	No	See “Using the modify command to display information” on page 38.
Other WebSphere for z/OS operations				
ARM and restart	Yes	No	No	See “ARM and restart” on page 42.
Setting up error log streams for different clusters and servants	No	You can associate a log stream with a cluster from the administrative console	No	See “Setting up error log streams for different servers and server instances” on page 43. See *WebSphere Application Server for z/OS V5.0: Installation and Customization*, GA22-7909, for details on setting up error log streams.

Reviewers note:
Where should we point reader now that environment variables are not in the I & C?

Table 1. WebSphere for z/OS operations tasks (continued)

Task	MVS console	WebSphere Administration Console	TSO panel	Reference to associated procedure
Setting up SMF recording	Yes	Enable it from here, but initiate it from the MVS console.	No	See “Functions of SMF recording” on page 113. Also see ″Using the Administrative Console″ in the WebSphere Application Server for z/OS V5.0 InfoCenter.
Shutting down the WebSphere for z/OS environment	Yes	No	No	See “Stopping the WebSphere for z/OS environment” on page 25.
Taking a WebSphere for z/OS system cluster out of service	Yes	Application server only; You cannot take a WebSphere for z/OS system cluster out of service from the administrative console	No	
Workload Management				
Displaying the status of a WLM application environment	Yes	No	No	See “Using Workload Management for WebSphere for z/OS operations” on page 43.
Handling workload management and cluster failures	Yes	No	No	See “Using Workload Management for WebSphere for z/OS operations” on page 43.
Getting out of the stopped state and back to the available state	Yes	No	No	See “Using Workload Management for WebSphere for z/OS operations” on page 43.
Checking and managing the workload management application environment (display, stop/queisce, restart/resume)	Yes	No	No	See “Displaying the status of a WLM application environment” on page 43.

Operating WebSphere for z/OS from the Administrative Console

The operations application of the administrative console allows you to perform the following tasks to operate your WebSphere for z/OS environment:

- Start a cluster or server.
- Stop a cluster or server.
- Cancel a cluster or server.

- Cancel and restart a cluster or server.
- Filter the operations window.
- Perform a warm start for a cluster or server (application servers only).

For more information, see ″Using the Administrative Console″ in the WebSphere Application Server for z/OS V5.0 InfoCenter, access to which can be obtained through the WebSphere forz/OS library Web site http://www.ibm.com/software/webservers/appserv/zos_os390/library.html.

Operating WebSphere for z/OS from the MVS console

WebSphere for z/OS operations can be automated using standard console automation products. Please see Chapter 3, “Operating WebSphere for z/OS”, on page 11 for a description of basic WebSphere for z/OS operating procedures that you can run from the MVS console.

Note: All automations for the WebSphere for z/OS environment are done using interfaces from the MVS console. Products such as Netview are presented copies of messages that are to be displayed on the MVS console. These automation products can also enter commands into the system using a ″virtual″ MVS console as a source.

Chapter 3. Operating WebSphere for z/OS

This chapter describes basic WebSphere for z/OS operating procedures that you can run from the MVS console.

Before performing these tasks, please review the following terms:

z/OS or OS/390 system

A computer and its associated devices where z/OS or OS/390 and WebSphere for z/OS are running.

sysplex

A set of z/OS or OS/390 systems communicating and cooperating with each other through certain multi-system hardware components and software services to process customer workloads. A sysplex is a single-image system complex. It is two or more z/OS or OS/390 systems that, together, provide a single system complex (for example, it may be two LPARs on the same hardware). This means that while a sysplex is composed of multiple systems, it acts and reacts like a single instance.

cell

A logical collection of nodes from the network. You can have more than one cell in a network, but a cell can't span networks.

cluster

A logical collection of replicated servers in a node.

node

A logical grouping of managed servers. A node usually corresponds to a physical computer system with a distinct IP host address. Node names usually are identical to the host name for the computer. A node agent manages all WebSphere Application Server servers on a node. The node agent represents the node in the management cell.

location service daemon

The initial point of contact within the WebSphere for z/OS cell. It publishes a network address that other servers or clients use to make requests to the WebSphere for z/OS system. The location service daemon accepts the requests, determines which server in the node can provide the function requested, and then routes the request to the server.

server

A functional component on which WebSphere for z/OS applications run. It is an instance of a replicated server that can provide all the functions that the server makes available. All servers within a cluster are identical in structure.

You can manage a server through the administrative console or the MVS console using the server's unique name.

A server has two kinds of address spaces: a controller and one or more servants. Application server code runs in a servant. A servant can be replicated based on the workload demands of the system. The controller queues messages to the servant.

Start operations

Starting the WebSphere for z/OS environment

This section describes how to start up the WebSphere for z/OS environment. See *WebSphere Application Server for z/OS V5.0: Installation and Customization*, GA22-7909, for details about the required subsystems that must be in place before you can start up the environment.

A typical WebSphere for z/OS run time includes two nodes:

- Deployment Manager node—includes a location service daemon and a Deployment Manager with a controller and any number of servants
- Application Server node— includes a location service daemon, a node agent with a controller, and an Application Server with a controller and any number of servants

Steps for starting a server

Before you begin: Before you start any of the application servers, you must validate that any resource managers required by your applications (DB2, CICS, etc) are available.

For information on how to check that your resource managers are available, refer to the System Administration section of the WebSphere Application Server for z/OS V5.0 InfoCenter, access to which can be obtained through the WebSphere for z/OS library Web site http://www.ibm.com/software/webservers/appserv/zos_os390/library.html.

Start all prerequisite subsystems (see *WebSphere Application Server for z/OS V5.0: Installation and Customization*, GA22-7909).

Perform the following steps to start an application server:

1. To start a server, issue the following command:

   ```
   S controlregionprocname,JOBNAME=server_shortname,ENV=cell_shortname.Node_shortname.Server_shortna
   ```

   where:

   controlregionprocname
   Is the JCL procedure name in the proclib that is used to start the server.

   server_shortname
   Is the short name of the server (or the step name used to start the proc). This allows you to identify the address space that is running when you view it in the SDSF panels.

   cell_shortname.Node_shortname.Server_shortname
   The ENV variable is a concatenation of the cell shortname, the node shortname, and the server shortname.

   Note: This command should be all upper case.

   For example,

   ```
   S BBO5ACR.BBOS001,ENV=SY1.SY1.BBOS001
   ```

   You know the control region is up when you get the following messages:

   ```
   $HASP100 BBO5ACR  ON STCINRDR
   $HASP373 BBO5ACR  STARTED
   ```

```
BBOO0001I WEBSPHERE FOR Z/OS CONTROL PROCESS BBODMNB/SY1/BBOC001/BBOS001 IS STARTING.

IRR812I PROFILE BBO*.* (G) IN THE STARTED CLASS WAS USED TO START BBOS001S WITH JOBNAME BBOS00
$HASP100 BBOS001S ON STCINRDR
$HASP373 BBOS001S STARTED

+BBOO0004I WEBSPHERE FOR Z/OS SERVANT PROCESS BBODMNB/SY1/BBOC001/BBOS001 IS STARTING.
+BBOO0020I INITIALIZATION COMPLETE FOR WEBSPHERE FOR Z/OS SERVANT PROCESS BBOS001.
BBOO0019I INITIALIZATION COMPLETE FOR WEBSPHERE FOR Z/OS CONTROL PROCESS BBOS001.
```

2. The Controller Region will automatically start the daemon by issuing a command that looks like:

   ```
   S <dmn_proc>,JOBNAME=<dmn_jobname>,ENV=<cell_shortname.Node_shortname.daemon_instancename>
   ```

   Following is an example of the messages displayed during daemon startup:

   ```
   BBOO0001I WEBSPHERE FOR Z/OS CONTROL PROCESS BBODMNB/SY1/BBOC001/BBOS001 IS STARTING.

   IRR812I PROFILE BBO*.* (G) IN THE STARTED CLASS WAS USED TO START BBO5DMN WITH JOBNAME BBO5DMN
   $HASP100 BBO5DMN  ON STCINRDR
   $HASP373 BBO5DMN  STARTED

   BBOO0007I WEBSPHERE FOR Z/OS DAEMON BBODMNB/SY1/BBODMNB/SY1 IS STARTING.

   IEC130I STEPLIB  DD STATEMENT MISSING
   ITT102I CTRACE WRITER BBOWTR IS ALREADY ACTIVE.

   BBOO0215I PRODUCT 'WAS FOR Z/OS' SUCCESSFULLY REGISTERED WITH IFAED SERVICE.
   BBOO0015I INITIALIZATION COMPLETE FOR DAEMON SY1.
   ```

3. WLM will start servant address spaces with a command that looks like:

   ```
   S <Srv_Reg_Proc>,JOBNAME=<Server_shortname>,ENV=<Cell_shortname.Node_shortname.Server_shortnam
   ```

4. Determine if the Daemon is up. See “Determining if the location service daemon is up” on page 13. If it is up, go to the next step to start the server.

5. To start a server from the administrative console, click **Servers** > **Applcation Server** in the console navigation tree to access the Application Server page.

6. Place a checkmark in the check box beside the application server that you want started and click **Start**.

7. View the **Status** value and any messages or logs to see whether the server starts.

Determining if the location service daemon is up

Before starting either clusters or servers, you need to know if the location service daemon is up. To determine this, issue one of the following four commands:

`d a,l`
   Displays a list of *all* address spaces.

`d a,a`
   Displays a list of *all active* address spaces.

`d a,`*`address-space-name`*

Example: `d a,bboasr1`. Displays only the address space in which you are interested.

Note: This command is recommended over the first two because it will not yield such a lengthy list on a production system. Of course, you need to know the name of the address space for which you are looking.

`d a,bbo*`

Displays a list of all active address spaces that start with ″BBO″.

You will know the system is up if you see the address space you are looking for.

Starting clusters and servers

This section describes how to start clusters and servers.

Notes:

1. Before you start any cluster or server, make sure that the location service daemon is up by issuing one of the display commands (described in “Determining if the location service daemon is up” on page 13). If the location service daemon is not up, you must start it. Please see “Steps for starting a server” on page 12 for this procedure.
2. From the MVS console, you need to start each individual server that you wish to run. The administrative console, however, gives you the option to either do that or start a cluster which will start all defined servers automatically.

Steps for starting a cluster from the MVS console

Perform the following steps to start a cluster:

1. Determine if the location service daemon is up. If it is up, go to the next step to start the cluster.
2. To start a cluster from the MVS console, you need to start each individual server that you wish to have started. See “Steps for starting a server” on page 12.

Steps for starting a cluster from the administrative console

Perform the following steps to start a cluster:

1. Determine if the location service daemon is up. If it is up, go to the next step to start the cluster.
2. To start a cluster from the administrative console, click **Servers** > **Clusters** in the console navigation tree to access the Server Cluster page.
3. Put a checkmark in the check boxes beside those clusters whose members you want started.
4. Click **Start** or **RippleStart**.
   - **Start** launches the server process of each member of the cluster by calling the node agent for each server to start the servers. After all servers are running, the state of the cluster changes to websphere.cluster.running. If the call to a node agent for a server fails, the server will not start.

- **RippleStart** combines stopping and starting operations. It first stops and then restarts each member of the cluster.

Cold starting WebSphere for z/OS

- For information on preparing for a cold start, refer to the System Administration section of the WebSphere Application Server for z/OS V5.0 InfoCenter, access to which can be obtained through the WebSphere for z/OS library Web site http://www.ibm.com/software/webservers/appserv/zos_os390/library.html.
- For the cold start procedure, *WebSphere Application Server for z/OS V5.0: Installation and Customization*, GA22-7909.

Note: If this is not an initial cold start, use your current configuration files that you saved for the cold start rather than the initial file sent with WebSphere for z/OS. This set of files includes your main configuration file and your application code files:

- J2EE applications have ear files in the HFS.

Hot starting WebSphere for z/OS

For the hot start procedure, refer to *WebSphere Application Server for z/OS V5.0: Installation and Customization*, GA22-7909.

Warm starting WebSphere for z/OS

For the warm start procedure, refer to *WebSphere Application Server for z/OS V5.0: Installation and Customization*, GA22-7909.

Quick starting WebSphere for z/OS

For the quick start procedure, refer to *WebSphere Application Server for z/OS V5.0: Installation and Customization*, GA22-7909.

Restart operations

Peer restart and recovery

This section describes the WebSphere for z/OS support for recovery after failures when running in a sysplex of two or more systems.

Prerequisites

Make sure **every** system (your original system as well as any systems intended for recovery) has the following installed:

- z/OS v1.2
- WebSphere Application Server for z/OS V5
- RRS APAR OW51091
- DB2 APAR PQ57123

Note: The following products individually support peer restart and recovery, providing the above prerequisites are all properly installed:

- IMS 7.10
- CICS 1.3
- MQSeries 5.2

The products mentioned above may not work in conjunction with other subsystems in the same transaction.

To allow WebSphere for z/OS to restart on an alternate system, the prerequisites must be met on every participating system in the sysplex **before** reconfiguring the ARM policies to enable peer restart and recovery. Installing the SPE on all your systems will not hinder your current running atmosphere if you want to continue to only restart in place. If this is not done, there is a possibility that the control region will not be able to move back—OTS will attempt to restart on the alternate system and fail. If there are any URs that are unresolved with RRS once this happens, the control region will not be allowed to restart on the home system until RRS is cancelled on the alternate system. In a nutshell, you'll be stuck, so make sure your prerequisites are met beforehand! For more information on OTS and RRS, see *z/OS MVS Programming: Resource Recovery*, SA22-7616.

Note: If you do not plan to use peer restart, you do not need to abide by these functional prerequisites. Your system will instead use the restart in place function that already exists.

Overview of peer restart and recovery

The goal of every system is to have as little downtime as possible. Sometimes, however, system failures are inevitable (like if the power unexpectedly goes out in your main system). When this happens, a course of restart action you can take is to restart on a peer system in the sysplex—a function called peer restart and recovery.

If you experience a main system failure that results in indoubt transactions with unknown outcomes, you need to obtain those intended transactional outcomes (ideally correctly) before the data can be utilized again. Peer restart and recovery provides an automated means of accomplishing this by restarting the control region on a peer system so that the ″locks″ that block the data can be dropped and the outcomes determined. This is in contrast to how a system usually handles a failure by automatically rolling back.

Note: Starting a server on a system to which it was not configured will implicitly place it into peer restart and recovery mode.

In this environment, the work is transactional in nature and is distributed across the sysplex. The work accesses one or more resource managers which manage access to shared data. New work is not accepted while recovery takes place. Processing instead continues on servers that are running on the configured system of the original control region—the sysplex, replicated control regions, and data sharing.

Note: Clients will see a performance impact if the systems were running at capacity. In an attempt to minimize the memory and CPU impact on the alternate system, server regions are not restarted.

In general, this section will talk about system failures. It is possible to recover from a server instance failing as well, of course. The major difference between the two is that the resource managers (such as DB2) and the RRS component still survive when a server instance fails. This is not the case when the system fails. When the server instance fails, the ARM policy should restart the server instance in place rather than start it on an alternate system. For more information on ARM, please see "ARM and restart" on page 42.

What WebSphere for z/OS must recover

WebSphere for z/OS itself is not a recoverable *resource* manager. It is a recoverable *communication* manager. This means that WebSphere for z/OS has no recoverable locks of its own and it doesn't need to manage locks nor manage lock states in a log. Instead, WebSphere for z/OS needs to make sure that both callers and callees are connected in each of the communications sessions of a distributed transaction.

Example of WebSphere for z/OS as a recoverable communication manager: This example outlines the organization of servers in a recoverable communication manager setup. Use the following diagram to understand the sysplex layout:

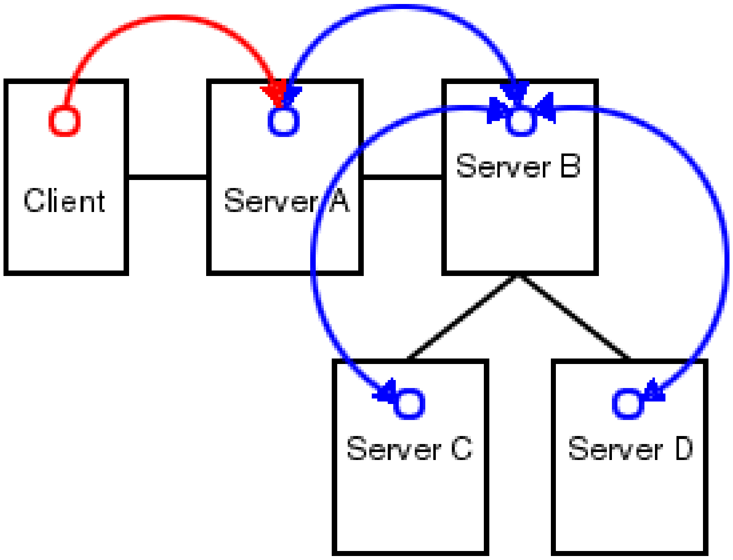


Figure 1. Organization of servers in a recoverable communication manager setup

Suppose there is a client that talks to server A. Server A then talks to server B, which in turn talks to servers C and D. In this example, server A is the superior, server B is the superior subordinate, and servers C and D are subordinates. With respect to server B, there is a communication session **from** server A and one each **to** servers C and D. OTS logs each of these communication sessions as *recoverable resource references* and prepares them for recovery so that, in the event of a failure, server B can reestablish the communication sessions to these servers.

When server B fails, server A and servers C and D cannot communicate to each other since server B is the intermediary between them. So, when server B recovers, it reads the log and reestablishes the communication sessions to the other servers. Further, once connectivity is reestablished, servers B, C, and D can determine the outcome of the transaction.

In-flight work versus indoubt work

This section describes recovery of in-flight versus indoubt work. When we talk about ″server A″, ″server B″, etc., we are referring to the servers from "Example of WebSphere for z/OS as a recoverable communication manager" on page 17.

In-flight work and presumed abort: If you have a distributed transaction that spans several servers, transactional locks may be held by resource managers involved in that work. When a failure occurs before that distributed transaction has started to commit, WebSphere for z/OS and the resource managers go into presumed abort mode. In this mode, the resource managers abort (rollback) the transaction.

Notes:

1. The effect of a server failure or communications failure will vary depending on which server is executing the work at the time of failure.
2. An OTS timeout may be required to rollback the subordinate branches of the distributed transaction tree.

Example: A common case of this is when you have a server B Web client that is driving a session bean in the same server. That session bean has executed work against entity beans in servers C and D. All of the servers are involved in the same distributed, global transaction. Suddenly, server B fails while the session bean is in-flight (meaning it hadn't started to commit yet). Servers C and D are waiting for more work or the start of the two-phase commit protocol, but, while in this state, the transactional locks may still be held by the resource managers. So, the server roles are as follows:

- Server A: Servlet/JSP executed
- Server B: Session bean accessed
- Server C: Entity bean accessed
- Server D: Entity bean accessed

Once the timeout occurs, since we were in-flight at the time of the failure, we will rollback the transaction branch.

When local resource managers are involved, RRS will ensure that they are called to perform presumed abort processing. When doing recovery, RRS will work with the resource managers to ensure that the recovery is done properly. When a failure occurs while work is in-flight, RRS will direct the resource managers involved in the local UR to rollback.

The WebSphere for z/OS runtime always assumes that there is recovery to do. Every time a server comes up, it does something different depending on what mode it is in:

- If the server is running in restart/recovery mode, it checks to see whether there is any recovery required. If so, it attempts to complete the recovery and either succeeds or terminates.
- If the server is running normally, the restart/recovery transaction does not have to complete before it takes on new work. Once it knows what the restart work is, it can begin to take in new work.

Indoubt work: In the event that there is some work in server B that was in the process of committing but had not yet completed, the server will have to recover. Often, this recovery consists of merely having the upstream server (the recovery coordinator (server A in our example)) tell the downstream servers (the resources (servers C and D in our example)) the outcome of the transaction. In order to do this, server B needs to reestablish communication with servers A, C, and D. In our implementation, this is done by saving, in the RRS log, the recovery coordinator references for the upstream servers and the resource references for the downstream servers. At recovery time, the logs are read and then communication is reestablished back to these servers.

`CLIENT_TIMEOUT` environment variable: The `CLIENT_TIMEOUT` environment variable designates the amount of time that the IIOP client will wait for a response to a method request that it has made to another server. Setting this environment variable to a reasonable value (at all times) is strongly suggested because, in the case of sysplex TCP/IP, there may not be a notification if the socket has a problem when the server to which you are talking fails in the dispatch. If a time-out value is not set in this case, the method will never return. This is obviously dangerous because, if that method happened to be an OTS method such as prepare, the two phase commit process would not continue until a response is received. The CLIENT_TIMEOUT variable guarantees some sort of a response. By default, CLIENT_TIMEOUT is set to 0 (unlimited). See *WebSphere Application Server for z/OS V5.0: Installation and Customization*, GA22-7909 for more information.

New work

The procedures for the recovery of inflight and indoubt work have been described in some detail, but how is new work handled on a recovered server? Once the indoubt and inflight work has been completed, the WebSphere for z/OS server shuts down. A new WebSphere for z/OS server configured for that system may now be started up to accept new work.

Note: Special considerations must be taken to begin new work on a WebSphere for z/OS using IMS Connect after recovering to an alternate system. Once the recovery has been completed, IMS Connect starts, but is not usable without some manual intervention. On the current IMS Connect WTOR perform the following commands nn,viewhws followed by nn,opends XXX where XXX is the IMS subsystem name displayed in the result of the nn,viewhws query. The IMS datastore needs to reflect ′active′ status, which can be seen in the example below:

```
*17 HWSC0000I *IMS CONNECT READY*  IMSCONN
 R 17,VIEWHWS
 IEE600I REPLY TO 17 IS;VIEWHWS
 HWSC0001I   HWS ID=IMSCONN     Racf=N
 HWSC0001I      Maxsoc=100  Timeout=12000
 HWSC0001I    Datastore=IMS      Status=ACTIVE
 HWSC0001I      Group=IMSGROUP Member=IMSCONN
 HWSC0001I      Target Member=IMSA
 HWSC0001I    Port=9999     Status=ACTIVE
 HWSC0001I      No active Clients
 HWSC0001I    Port=LOCAL    Status=ACTIVE
 HWSC0001I      No active Clients
```

Once this has been completed IMS Connect is ready for new work to be completed on the server.

When peer restart and recovery doesn’t work

The major reason for recovery failure is if you experience a network outage while in the process of recovering. If the system cannot reach the superior or subordinate because the network is dead, communications cannot reestablish and the transaction cannot completely resolve.

When WebSphere for z/OS cannot automatically resolve all of the URs returned from RRS at restart, RRS will not allow WebSphere to move back to the home (original) system. If WebSphere tries to go back while URs are still incomplete, you will receive an error code (C9C2186A) and a message describing an F02 return code from ATRIBRS. In order to get around this, manual resolution is required to mark the server for ″restart anywhere.″ RRS will do that once all of the URs in which WebSphere is involved are ″forgotten.″ If RRS fails to mark the server ″restart anywhere,″ the server, upon failure, is required to start on the recovery system. This is not good because it doesn’t allow you to move the server back to its true home system.

The ultimate goal of this is to resolve all transactions that WebSphere (the server instance- owned interests that could not complete recovery) is involved in, and then, if necessary, remove all of the WebSphere interests that remain in those URs. Once that is complete, browsing the RM data log will show if the resource manager is marked ″restart anywhere.″

You **want** to see:

```
RESOURCE MANAGER=BSS00.SY1.BBOASR4A.IBM
RESOURCE MANAGER MAY RESTART ON ANY SYSTEM
```

You do **not** want to see:

```
RESOURCE MANAGER=BSS00.SY2.BBOASR4A.IBM
RESOURCE MANAGER MUST RESTART ON SYSTEM SY2
```

Using RRS panels to resolve indoubt units of recovery

Note: There are RRS version requirements that you must heed when using peer restart and recovery. For more information on these requirements, please see *WebSphere Application Server for z/OS V5.0: Installation and Customization*, GA22-7909, and *z/OS MVS Programming: Resource Recovery*, SA22-7616.

If you get the following console message, you need to:

1. Note the server named in ″*string*″.
2. Go to SYSPRINT (the status queue for that server) and search for the named server (will be BBOU076*) to find any of those messages that occurred.
3. Follow the resulting messages.

- **BBOU0754D** OTS UNABLE TO RESOLVE ALL INCOMPLETE TRANSACTIONS FOR SERVER *string*. REPLY CONTINUE OR TERMINATE.

Note: RRS will not allow an operator to resolve an indoubt UR if the DSRM for that UR is active at the time, so you need to stop the server. To do this, reply ″TERMINATE″ to the CONTINUE/TERMINATE WTOR.

The following non-console messages, which can be used to trigger automation, provide details about daemon activities when attempting restart and recovery:

- **BBOU0003E** WEBSPHERE FOR z/OS CONTROL REGION *string* ENDED ABNORMALLY, REASON=*hstring*.
- **BBOU0009E** WEBSPHERE FOR z/OS DAEMON *string* ENDED ABNORMALLY, REASON=*hstring*.
- **BBOU0739I** WEBSPHERE FOR z/OS CONTROL REGION *string* NOT STARTING ON CONFIGURED SYSTEM *string*

For the full message explanations, see the Troubleshooting section of the WebSphere Application Server for z/OS V5.0 InfoCenter, access to which can be obtained through the WebSphere for z/OS library Web site http://www.ibm.com/software/webservers/appserv/zos_os390/library.html.

The following messages, which are written only in recovery and restart mode, provide details about transaction(s) that could not be resolved during restart and recovery:

- **BBOU0747I** TRANSACTION SERVICE RESTART INITIATED ON SERVER *string*
- **BBOU0748I** TRANSACTION SERVICE RESTART UR STATUS COUNTS FOR SERVER *string*: IN-BACKOUT=*dstring*, IN-DOUBT=*dstring*, IN-COMMIT=*dstring*
- **BBOU0749I** TRANSACTION SERVICE RESTART AND RECOVERY ON SERVER *string* IS COMPLETE
- **BBOU0750I** SERVER *string* IS COLD STARTING WITH RRS

- **BBOU0751I** SERVER *string* IS WARM STARTING WITH RRS
- **BBOU0752I** TRANSACTION SERVICE RESTART AND RECOVERY ON SERVER *string* IS COMPLETE. THE SERVER IS STOPPING.
- **BBOU0753I** TRANSACTION SERVICE RECOVERY PROCESSING FOR RRS URID *'string'* IN SERVER *string* IS COMPLETE.
- **BBOU0755I** TRANSACTION SERVICE RESTART AND RECOVERY FOR SERVER *string* IS NOT COMPLETE. THE SERVER IS STOPPING DUE TO OPERATOR REPLY.
- **BBOU0756I** TRANSACTION SERVICE RESTART AND RECOVERY FOR SERVER *string* IS CONTINUING DUE TO OPERATOR REPLY.
- **BBOU0757I** TRANSACTION SERVICE RESTART AND RECOVERY FOR SERVER *string* IS STILL PROCESSING *dstring* INCOMPLETE UNIT(S) OF RECOVERY.
- **BBOU0761W** UNABLE TO RESOLVE THE OUTCOME OF THE TRANSACTION BRANCH DESCRIBED BY

  URID: *'string'*
  XID FORMATID: *'string'*
  XID GTRID: *'string'*
  XID BQUAL: *'string'*

  BECAUSE THE OTS RECOVERY COORDINATOR FOR SERVER *string* ON HOST *string*:*dstring* COULD NOT BE REACHED.
- **BBOU0762W** UNABLE TO PROVIDE THE SUBORDINATE OTS RESOURCE IN SERVER *string* ON HOST *string*:*dstring* WITH THE OUTCOME OF THE TRANSACTION DESCRIBED ON THIS SERVER BY

  URID: *'string'*
  XID FORMATID: *'string'*
  XID GTRID: *'string'*
  XID BQUAL: *'string'*

  BECAUSE THIS SERVER HAS BEEN UNABLE TO RESOLVE THE OUTCOME WITH A SUPERIOR NODE.
- **BBOU0763W** UNABLE TO *string* THE SUBORDINATE OTS RESOURCE IN SERVER *string* ON HOST *string*:*dstring* FOR THE TRANSACTION DESCRIBED ON THIS SERVER BY

  URID: *'string'*
  XID FORMATID: *'string'*
  XID GTRID: *'string'*
  XID BQUAL: *'string'*

  OR ANOTHER RESOURCE INVOLVED IN THIS UNIT OF RECOVERY BECAUSE ONE OR MORE RESOURCES COULD NOT BE REACHED OR HAVE NOT YET REPLIED.
- **BBOU0764W** UNABLE TO FORGET THE TRANSACTION WITH HEURISTIC OUTCOME DESCRIBED ON THIS SERVER BY

  URID: *'string'*
  XID FORMATID: *'string'*
  XID GTRID: *'string'*
  XID BQUAL: *'string'*

BECAUSE THE SUPERIOR COORDINATOR FOR SERVER *string* ON HOST *string*:*dstring* HAS NOT INVOKED FORGET ON THE REGISTERED RESOURCE.

Pay particular attention to the URID, XID FormatId, XID Gtrid and XIDBqual attributes. These pieces of information will be used to manually resolve the relevant units of work via the RRS panels. For the full message explanations, see the Troubleshooting section of the WebSphere Application Server for z/OS V5.0 InfoCenter, access to which can be obtained through the WebSphere for z/OS library Web site http://www.ibm.com/software/webservers/appserv/zos_os390/library.html..

Steps for resolving indoubts if you receive messages BBOU0761W or BBOU0762W: Before you begin: These two messages indicate that this server could not determine the outcome from its superior. BBOU0762W, which describes the resource to which we could not provide an outcome, will always be accompanied by BBOU0761W (if you look in the server that is indicated by BBOU0762W, you should see a BBOU0761W message as well. See *z/OS MVS Programming: Resource Recovery*, SA22-7616 for more information on how to use the RRS panels and what administrative access (RACF access to the facility class, for example) is needed to resolve URs and remove interests.

Perform the following steps to view the outcome of other branches in the transaction and set the outcomes of indoubt branches to match.

1. Select option 3, ″Display/Update RRS Unit of Recovery information″ on the main RRS panel.

   ____

2. Attempt to resolve indoubt URs with an outcome identical to the outcome of other branches of the same transaction (unless those other branches show and outcome of ″InPrepare,″ in which case you should resolve to ″InBackout.″ . Complete these steps to view the outcome of other known transaction branches:
   - Specify the XID FormatId (in decimal) in the ″Format ID″ field.

     Note: The XID FormatId must be converted from hex to decimal.
   - Specify the XID Gtrid (in hex) in the ″GTRID Pattern″ fields on the query panel.

     Note: The XID Gtrid must be entered in 16 bytes to a line.
   - Press ″enter″ to execute the query.

   ____

3. The query results obtained in step 2 above will show all URIDs involved with the specified transaction. In the query results, take note of the outcomes in the ″state″ column. If any other branches of the transaction are ″InCommit″ or ″InBackout,″ match the corresponding ″InDoubt″ branches to read the same:
   a. In the left-hand column labeled ″S″, enter ″c″ to commit or ″b″ to backout the UR that was identified in message BBOU0761W.

      Note: Manually resolving the transaction can lead to mixed transaction outcomes across resource managers and servers.

   ____

4. Copy to the clipboard the URID on this panel.

   ____

5. Remove the interest for this URID (see “Steps for resolving indoubts if you receive message BBOU0763W”).

You know you are done when RRS marks the subordinate server as ″restart anywhere.″ Determine this by choosing option 1 under ″Browse and RRS log stream″ and then choosing suboption 4 under ″RRS Resource Manager Data log.″

Steps for resolving indoubts if you receive message BBOU0763W: Before you begin: This message indicates that a server has determined the transaction outcome but has not been able to communicate it to its subordinates. When this message is displayed, there is a possibility of an heuristic outcome. See *z/OS MVS Programming: Resource Recovery*, SA22-7616 for more information on how to use the RRS panels and what administrative access (RACF access to the facility class, for example) is needed to resolve URs and remove interests.

Perform the following steps to remove an expression of interest in this UR.

1. Select option 3, ″Display/Update RRS Unit of Recovery information″ on the main RRS panel.
2. To view the details of this URID, enter it in the ″URID Pattern″ field on the query panel. Press ″enter″ to execute the query.
3. The query results should display the UR. Take note of whether the state of this UR is ″InCommit,″ ″InBackout″ or ″InForget.″ In the column labeled ″S″, enter ″v″ to display the details for this UR.
4. The ″RRS Unit of Recovery Details″ panel will open. Near the bottom of the panel will be a heading titled ″Expressions of Interest″ followed by one or more rows below it. These rows represent each individual expression of interest in this UR. Complete these steps to remove the OTS interest:
   a. Find the row that represents the ″OTS″ interest.

      Note: This row will have an RM name in the form of ″BSS00.xxx.yyy.IBM″, where ′xxx′ is the system to which the server was configured and ′yyy′ is the specific server name.

   b. Type ″r″ in the column labeled ″S″ to indicate that you want to remove this interest.
   c. Press ″enter″ to execute the query.
5. The ″RRS Remove Interest Confirmation″ panel will open. The RM name and UR identifier fields are pre-filled. Press ″enter″ to confirm the removal of this interest.

From this point onward, any subordinate nodes that restart and ask this server about this UR will be unable to obtain this information. If you restart the server containing these nodes, they may be assigned an outcome different from the outcome of the transaction. You must manually resolve these nodes before you bring up the servers and start the server for which you just released the UR.

You know you are done when RRS marks the subordinate server as ″restart anywhere.″ Determine this by choosing option 1 under ″Browse and RRS log stream″ and then choosing suboption 4 under ″RRS Resource Manager Data log.″

Steps for resolving indoubts if you receive message BBOU0764W: Before you begin: This message indicates that the transaction outcome has been determined and communicated to the subordinate, but the subordinate resource has not yet been ″forgotten.″ When this message is displayed, there has already been a heuristic outcome.

Note: The subordinate will not let the UR move to the ″in-forget″ state until it is told to forget it. In other words, the subordinate will still be involved in a UR, so RRS will not mark the subordinate server as ″restart anywhere.″ See "When peer restart and recovery doesn't work" on page 19 for more information on ″restart anywhere.″

See *z/OS MVS Programming: Resource Recovery*, SA22-7616 for more information on how to use the RRS panels and what administrative access (RACF access to the facility class, for example) is needed to resolve URs and remove interests.

Perform the following steps to remove an expression of interest in this UR.

1. Select option 3, ″Display/Update RRS Unit of Recovery information″ on the main RRS panel.

2. To view the details of this URID, enter the URID in the ″URID Pattern″ field on the query panel. Press ″enter″ to execute the query.

3. The query results should display the UR. In the column labeled ″s″, enter ″v″ to display the details for this UR.

4. The ″RRS Unit of Recovery Details″ panel will open. Near the bottom of the panel will be a heading titled ″Expressions of Interest″ followed by one or more rows below it. These rows represent each individual expression of interest in this UR. Complete these steps to remove the OTS interest:
   a. Find the row that represents the ″OTS″ interest.

      Note: This row will have an RM name in the form of ″BSS00.xxx.yyy.IBM″, where ′xxx′ is the system to which the server was configured and ′yyy′ is the specific server name.
   b. Type ″r″ in the column labeled ″S″ to indicate that you want to remove this interest.
   c. Press ″enter″ to execute the query.

5. The ″RRS Remove Interest Confirmation″ panel will open. The RM name and UR identifier fields are pre-filled. Press ″enter″ to confirm the removal of this interest.

You know you are done when RRS marks the subordinate server as ″restart anywhere.″ Determine this by choosing option 1 under ″Browse and RRS log stream″ and then choosing suboption 4 under ″RRS Resource Manager Data log.″

Stop operations

Stop vs. immediate stop (cancel)

Stop means that all currently-running transactions are carried out before the application controller is taken down, while or **immediate stop** (cancel) means that the application controller is immediately taken down without waiting for the active transactions to complete.

Example: When you request that a server **stop**, the current work is finished before the server is stopped, and when you request an **immediate stop** (cancel), the server stops immediately and ignores any current or pending tasks.

Stopping the WebSphere for z/OS environment

This section is the reverse of the startup procedures. However, it describes what has to be canceled first before shutting down the WebSphere for z/OS environment.

Steps for stopping the WebSphere for z/OS environment

Before you begin: When you stop the daemon on a base application server, it brings down the application servers on that system.When you stop the location service daemon on one system, it doesn't bring down the servers on the other system(s).

Perform the following steps to shutdown the WebSphere for z/OS environment:

1. To stop a cluster from the administrative console, click **Servers** > **Clusters** in the left-hand navigation tree.

   Note: As opposed to servers, clusters can be stopped only from the administrative console. (Actually, clusters can also be stopped from the console, but it is much more work because you need to stop all servers that belong to that cluster by hand, and they might be located in a Sysplex across systems.)

2. Place a checkmark in the checkbox next to the cluster you want to stop and click either the **Stop** or the **Immediate Stop** button near the top of the Server Cluster panel.

   All application servers in the sysplex associated with this cluster will stop. In addition, a **stop** can be issued against each individual server from the MVS console.

   To shut down the WebSphere for z/OS environment on a system, stop that system's Daemon (see below). It will bring down all other server instances on the system. To bring WebSphere for z/OS down on all systems, stop the Daemons on all systems.

Note: When you stop the Daemon, you may get an `A03` abend and dump in your Daemon address space, but this does not impair the stop.

Stopping servers and servants from the administrative console

You can stop the individual servers that make up a cluster. When you stop an application server, the servants associated with that server are also stopped

For additional information, see the Admnistration section of the WebSphere Application Server for z/OS V5.0 InfoCenter, access to which can be obtained through the WebSphere for z/OS library Web site http://www.ibm.com/software/webservers/appserv/zos_os390/library.html.

Perform the following steps to stop a server:

1. To stop a cluster from the administrative console, click **Servers** > **Application Servers** in the left-hand navigation tree.

   ---

2. Place a checkmark in the checkbox next to each server you want to stop and click either the **Stop** or **Immediate Stop** button near the top of the Application Server panel.

   All application servers you have checked and their associated servants will stop. In addition, a **stop** can be issued against each individual server from the MVS console.

Steps for stopping the location service daemon from the MVS console

Before you begin: If you cancel the location service daemon, it cancels all other applications on that system.

Issue **one** of the following commands to stop the location service daemon:

- `STOP DAEMON01`
- `STOPCANCEL DAEMON01,ARMRESTART`

Note: The command containing ″`ARMRESTART`″ must be used if ARM is active and you want it to restart the location service daemon

Cancel operations

Note: See “Stop vs. immediate stop (cancel)” on page 25 for a description of the difference between stopping and cancelling an application control region.

Canceling the location service daemon

Note: Use caution with the cancel command.

This section describes how to cancel the location service daemon. Canceling the location service daemon brings down all other application servers on this system.

Steps for canceling the location service daemon from the MVS console

Before you begin: If you cancel the location service daemon, it cancels all other applications on that system.

Issue **one** of the following commands to cancel the location service daemon:

- `CANCEL DAEMON01`
- `CANCEL DAEMON01,ARMRESTART`

Note: The command containing ″`ARMRESTART`″ must be used if ARM is active and you want it to restart the location service daemon

Display operations

Displaying the status of ARM-registered address spaces including WebSphere for z/OS servers and server instances

WebSphere for z/OS ships with all control regions issuing automatic restart management (ARM) registration commands. If your installation enables ARM, you should read this section.

This section describes how to use ARM to display the status of all ARM-registered address spaces (including the address spaces of server instances) in the WebSphere for z/OS environment. ARM is used to restart all address spaces that go down, if they are registered with ARM. This does not apply if the address spaces are canceled.

Each WebSphere for z/OS control region registers with ARM. If a control region terminates abnormally or the system fails, ARM will try to restart the failing address spaces. In doing this, ARM will ensure that dependent address spaces are grouped together and will start in the appropriate order. In general, the default ARM policy will restart WebSphere for z/OS in place. If using a sysplex, see the *WebSphere Application Server for z/OS V5.0: Installation and Customization*, GA22-7909, for setup guidelines to ensure that no cross-system restarts are performed.

Steps for displaying the status of ARM-registered address spaces

Perform the following steps to use ARM to display the status of ARM registered address spaces (including the address spaces of server instances) in the WebSphere for z/OS environment:

1. Initialize all servers.

2. To display all registered address spaces (including the address spaces of server instances), issue the command:

```
d xcf,armstatus,detail
```

Displaying the status of a cluster or server

This section describes how to use ARM to display the status of a specific server or server instance in the WebSphere for z/OS environment.

Steps for displaying the status of a cluster or server:

Notes on automatic restart management and run-time: At run-time, be aware of the following regarding automatic restart management:

1. Once your server instances have initialized, you can display their status with regard to automatic restart management. To display all registered address spaces (including the address spaces of server instances), issue:

```
d  xcf,armstatus,detail
```

   To display the status of a particular server instance, use the `DISPLAY` command and identify the job name. For example, to display the status of the Daemon server instance (job BBODMN), issue:

```
d  xcf,armstatus,jobname=bbodmn,detail
```

2. If you issue STOP, CANCEL, or MODIFY commands against server instances, be aware of how automatic restart management behaves regarding WebSphere for z/OS server instances:

Table 2. Behavior of automatic restart management regarding WebSphere for z/OS server instances

If you issue . . .	Automatic restart management will . . .
STOP *address_space*	not restart the address space
CANCEL *address_space*	not restart the address space
CANCEL *address_space*, ARMRESTART	restart the address space
MODIFY *address_space*,CANCEL	not restart the address space
MODIFY *address_space*,CANCEL,ARMRESTART	restart the address space

Displaying active address spaces

This command is useful for displaying active address spaces (for example, when you want to know if DB2 is up).

Issue the following command to display (list) all active address spaces:

```
d a,l
```

Displaying active replies

Displaying active replies from the MVS console allows you to observe system activity and determine if the system requires an operator response.

Issue the following command to display (list) all active replies:

```
d r,r
```

Displaying units of work (threads) for DB2

This section describes how to display units of work (threads) for DB2.

Issue the following command to display units of work (active threads) for DB2:

```
–db2 dis thread(*)
```

Displaying indoubt units of work (threads) for DB2

This section describes how to display indoubt units of work (threads) for DB2.

An indoubt unit of work (thread) is a unit of recovery (UR or set of changes that are to be made, or not made, as a unit) in an indoubt state when a resource manager is coordinating the processing and RRS is waiting for the coordinator to tell it whether to resolve the UR by a commit or a backout. See *z/OS MVS Programming: Resource Recovery*, SA22-7616, for more information.

Issue the following command to display units of work (active threads) for DB2:

```
–db2 dis thread(*) type(indoubt)
```

You know that no unresolved threads exist if you see the message:

```
DB2 No Indoubt Threads Found
```

Displaying units of work for CICS

See *CICS Operations and Utilities Guide*, SC34-5991, for information on how to display units of work for CICS.

Displaying units of work (transactions) for IMS

This section describes how to display units of work (transactions) for IMS.

Steps for displaying units of work (transactions) for IMS

Perform the following steps to display units of work (transactions) for IMS:

1. To display the status of a specific transaction, issue the command:
   ```
   /dis tran trans-name
   ```
2. To display the status of a specific program, issue the command:
   ```
   /dis prog program-name
   ```
3. To display the number of Message Processing Regions (MPRs) that are currently active, issue the command:
   ```
   /display active region
   ```

   For more information about IMS commands, see *IMS/ESA Summary of Operator Commands*, SC26-8766.

Displaying units of work for RRS

See *z/OS MVS Programming: Resource Recovery*, SA22-7616, to for information on how to display units of work for RRS.

Modify operations

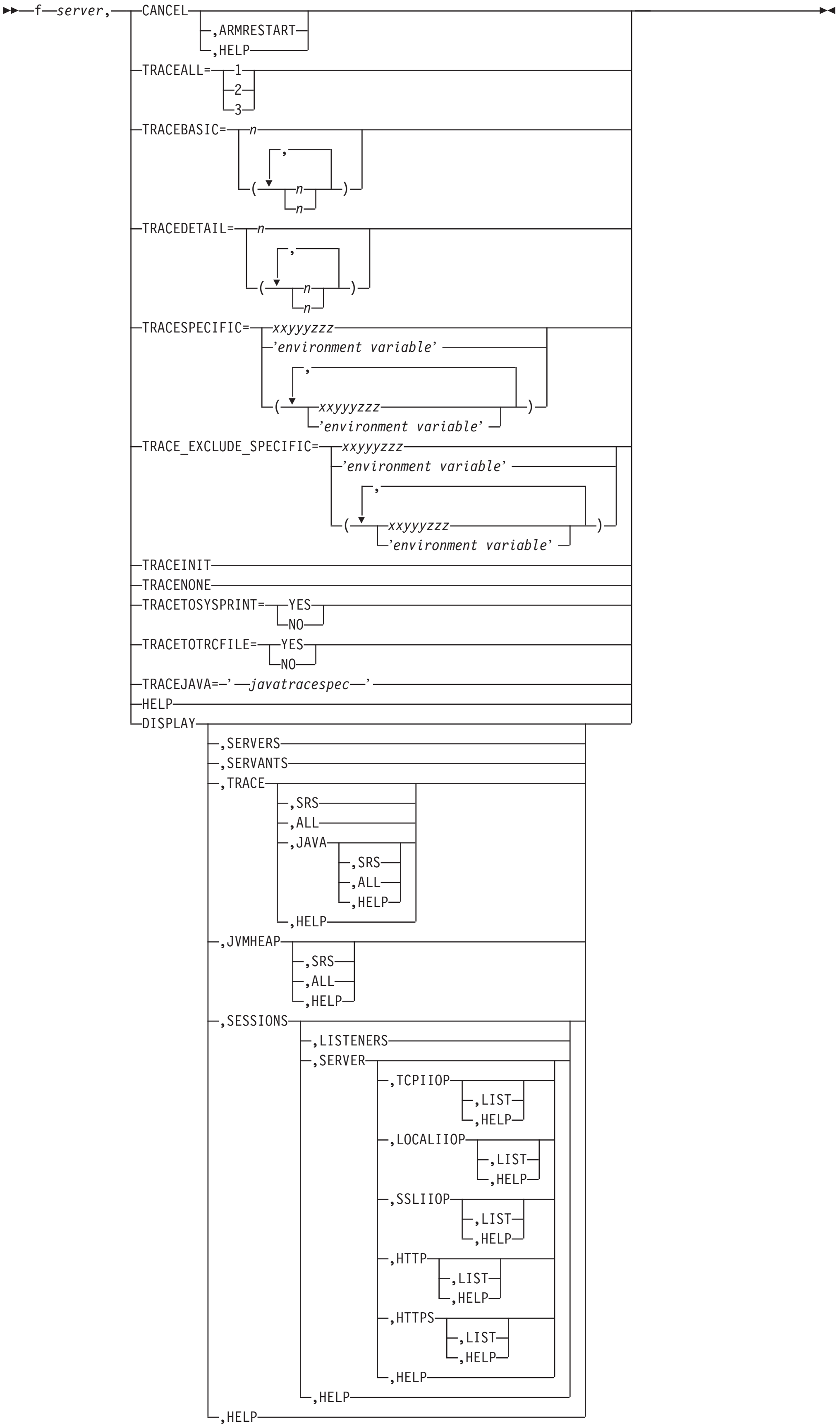
Modify command syntax

Purpose

To use the MVS console to dynamically modify WebSphere for z/OS operations.

Format

MODIFY Command


f server,
CANCEL
,ARMRESTART
,HELP
TRACEALL=
1
2
3
TRACEBASIC=
n
,
(
n
n
)
TRACEDETAIL=
n
,
(
n
n
)
TRACESPECIFIC=
xxyyyzzz
'environment variable'
,
(
xxyyyzzz
'environment variable'
)
TRACE_EXCLUDE_SPECIFIC=
xxyyyzzz
'environment variable'
,
(
xxyyyzzz
'environment variable'
)
TRACEINIT
TRACENONE
TRACETOSYSPRINT=
YES
NO
TRACETOTRCFILE=
YES
NO
TRACEJAVA=' javatracespec '
HELP
DISPLAY
,SERVERS
,SERVANTS
,TRACE
,SRS
,ALL
,JAVA
,SRS
,ALL
,HELP
,HELP
,JVMHEAP
,SRS
,ALL
,HELP
,SESSIONS
,LISTENERS
,SERVER
,TCPIIOP
,LIST
,HELP
,LOCALIIOP
,LIST
,HELP
,SSLIIOP
,LIST
,HELP
,HTTP
,LIST
,HELP
,HTTPS
,LIST
,HELP
,HELP
,HELP
,HELP

Parameters

CANCEL

Used to cancel a server. *server* refers to the server short name.

You can specify the following options:

ARMRESTART

Specify if you are using ARM and want ARM to restart the server after it terminates. If you don’t specify the ARMRESTART option on the CANCEL parameter, ARM will not restart the server.

HELP

Get help for the CANCEL syntax.

Note: You cannot use the CANCEL parameter to cancel a cluster from the MVS console. Instead, you must cancel each of the servers that make up the cluster.

TRACEALL=n

Use TRACEALL to establish a general trace level for the server.

Valid trace levels are 0 (none), 1 (exception), 2 (basic), and 3 (detailed tracing). Under normal conditions and in production, use 1 (exception).

Note: Be careful when using a level of 3 (detailed for all components) because it can potentially yield more data than can be handled reasonably.

TRACEBASIC=n

Specify the WebSphere for z/OS components for which you want to switch on a basic level of tracing.

This command has the ability to override a different tracing level established by `TRACEALL` for those components.

Note: Do not change this variable unless directed by IBM service personnel.

You can specify one or more of the following options for either TRACEBASIC or TRACEDETAIL:

0	RAS
1	Common Utilities
3	COMM
4	ORB
6	OTS
7	Shasta
9	OS/390 Wrappers
A	Daemon
E	Security
F	Externalization
J	JRAS (internal tracing—via direction from IBM support)
L	J2EE

TRACEDETAIL=n

Specify the WebSphere for z/OS components for which you want to switch on a detailed level of tracing

This command activates the most detailed tracing for the specified WebSphere for z/OS components and overrides different settings in TRACEALL. The selected components *(n,...)* are identified by their component-ID (valid values are the same as for TRACEBASIC above). Subcomponents, specified by numbers, receive detailed traces. Other parts of WebSphere for z/OS receive tracing as specified on the TRACEALL environment variable.

Note: Do not change this variable unless directed by IBM service personnel.

TRACESPECIFIC=xxyyyzzz

Specifies tracing overrides for specific WebSphere for z/OS trace points. Trace points are specified by 8-digit, hexadecimal numbers. To specify more than one trace point, use parentheses and separate the numbers with commas. You can also specify an environment variable name by enclosing the name in single quotes. The value of the environment variable will be handled as if you had specified that value on TRACESPECIFIC.

Note: Do not use TRACESPECIFIC unless directed by IBM service personnel.

TRACE_EXCLUDE_SPECIFIC=xxyyyzzz

Specifies trace points to exclude. Trace points to exclude are specified by 8-digit, hexadecimal numbers. To specify more than one trace point, use parentheses and separate the numbers with commas. You can also specify an environment variable name by enclosing the name in single quotes. The value of the environment variable will be handled as if you had specified that value on TRACE_EXCLUDE_SPECIFIC.

You can use TRACE_EXCLUDE_SPECIFIC as a mask to turn off otherwise-on traces. For example, use the TRACESPECIFIC command to turn on tracing for a whole part and then use TRACE_EXCLUDE_SPECIFIC to turn off one trace within that part.

Note: Do not use TRACE_EXCLUDE_SPECIFIC unless directed by IBM service personnel.

TRACEINIT

Reset to the initial trace settings.

TRACENONE

Turns off all trace settings.

TRACETOSYSPRINT={YES|NO}

Select whether or not to send the trace to sysprint.

YES specifies to send the trace to sysprint and NO stops sending the trace to sysprint.

TRACETOTRCFILE={YES|NO}

Select whether or not to direct the trace to the TRCFILE DD card.

YES specifies to send the trace to the TRCFILE DD card and NO stops sending the trace to the TRCFILE DD card.

TRACEJAVA

Modify the Java trace string. The java trace specification is used to control java tracing and conforms to the java trace specification rules. <Need reference to the InfoCenter here> *=all=enabled means to enable all types of tracing for all registered trace components.

HELP

Display a list of all the keywords you may use with the modify command. You can also use the HELP parameter after the CANCEL, and DISPLAY parameters to display lists of all the keywords you can use with either of these parameters.

DISPLAY
Displays the server's name, the system name where it is running, and the current code level.

You can specify the following options:

SERVERS
This command is directed to a server and displays the name, system name, and code level for each active server in the sysplex that is in the same cell.

SERVANTS
Displays a list of ASIDs of servants attached to the server against which you issued the display command.

TRACE
Display trace information for a server controller.

You can further modify this command with one of the following options:

SRS
Display trace information for all servants, one at a time.

ALL
Display trace information for the controller and all servants one at a time.

JAVA
Display the Java trace string settings for a server controller.

You can further modify this command with one of the following options:

SRS
Display Java trace information for all servants, one at a time.

ALL
Display Java trace information for the controller and all servants one at a time.

HELP
Display a list of all the keywords you may use with the modify display trace java command.

HELP
Display a list of all the keywords you may use with the modify display trace command.

JVMHEAP
Display the JVM heap information for a server controller.

You can further modify this command with one of the following options:

SRS
Display the JVM heap information for all servants, one at a time.

ALL
Display the JVM heap information for the controller and all servants, one at a time.

HELP
Display a list of all the keywords you may use with the modify display javaheap command.

SESSIONS
Display session information for the server.

You can further modify this command with one of the following options:

LISTENERS
Display the listening port numbers for each protocol. This is actually the default, so the `f server,sessions` and `f server,sessions,listeners` commands would have the same outcome.

SERVER
Display the number of sessions in use for each protocol on the server.

You can further modify this command with one of the following options:

TCPIIOP
Display the number of TCP/IP IIOP sessions active on the server.

You can further modify this command with one of the following options:

LIST
List the server session information for the TCP/IP IIOP protocol.

HELP
Display a list of all the keywords you may use with the modify display session server tcpipiiop command.

Note: The LIST and HELP parameters are also available with the other session protocol display commands.

LOCALIIOP
Display the number of LOCALIIOP sessions active on the server.

SSLIIOP
Display the number of SSLIIOP sessions active on the server.

HTTP
Display the number of HTTP sessions active on the server.

HTTPS
Display the number of HTTPS sessions active on the server.

HELP
Display a list of all the keywords you may use with the modify display session server command.

HELP
Display help for the modify display sessions command.

Modify command examples

This section gives examples of how to use the modify command and some sample output.

Getting help for the modify command

You can get syntax help for the various levels of the modify command.

Example 1: The following command will display a list of all the keywords you may use with the modify command:

```
f bbo5acr,help
```

Here is a sample display:

```
F BBO5ACR,HELP
BBOO0178I THE COMMAND MODIFY  MAY BE FOLLOWED BY ONE OF THE FOLLOWING KEYWORDS:
BBOO0179I CANCEL - CANCEL THIS CONTROL REGION
BBOO0179I TRACEALL - SET OVERALL TRACE LEVEL
BBOO0179I TRACEBASIC - SET BASIC TRACE COMPONENTS
BBOO0179I TRACEDETAIL - SET DETAILED TRACE COMPONENTS
BBOO0179I TRACESPECIFIC - SET SPECIFIC TRACE POINTS
BBOO0179I TRACEINIT - RESET TO INITIAL TRACE SETTINGS
BBOO0179I TRACENONE - TURN OFF ALL TRACING
BBOO0179I TRACETOSYSPRINT - SEND TRACE OUTPUT TO SYSPRINT (YES/NO)
BBOO0179I TRACETOTRCFILE - SEND TRACE OUTPUT TO TRCFILE DD CARD (YES/NO)
BBOO0179I DISPLAY - DISPLAY STATUS
BBOO0179I TRACE_EXCLUDE_SPECIFIC - EXCLUDE SPECIFIC TRACE POINTS
BBOO0179I TRACEJAVA - SET JAVA TRACE OPTIONS
```

Example 2: The following command will display a list of all the keywords you may use with the modify display command:

```
f bbo5acr,display,help
```

Here is a sample display:

```
F BBO5ACR,DISPLAY,HELP
BBOO0178I THE COMMAND DISPLAY, MAY BE FOLLOWED BY ONE OF THE FOLLOWING KEYWORDS:
BBOO0179I SERVERS - DISPLAY ACTIVE CONTROL REGIONS
BBOO0179I SERVANTS - DISPLAY SERVANT PROCESSES OWNED BY THIS CONTROL PROCESS
BBOO0179I SESSIONS - DISPLAY INFORMATION ABOUT COMMUNICATIONS SESSIONS
BBOO0179I TRACE - DISPLAY INFORMATION ABOUT TRACE SETTINGS
BBOO0179I JVMHEAP - DISPLAY JVM HEAP STATISTICS
BBOO0188I END OF OUTPUT FOR COMMAND DISPLAY,HELP
```

Example 3: The following command will display a list of all the keywords you may use with the modify display sessions command:

```
f bbo5acr,display,sessions,help
```

Here is a sample display:

```
F BBO5ACR,DISPLAY,SESSIONS,HELP
BBOO0178I THE COMMAND DISPLAY,SESSIONS, MAY BE FOLLOWED BY ONE OF THE
FOLLOWING KEYWORDS:
BBOO0179I LISTENERS - DISPLAY LISTENER INFORMATION
BBOO0179I SERVER - DISPLAY SERVER SESSION INFORMATION
BBOO0188I END OF OUTPUT FOR COMMAND DISPLAY,SESSIONS,HELP
```

Canceling application clusters and servers with the modify command

Before you begin: You cannot cancel a cluster from the MVS console. Instead, you must cancel each of the servers that make up the cluster.

Example 1: The following command will cancel the bbo5acr server:

```
f bbo5acr,cancel
```

Example 2: The following command will cancel the bbo5acr server and instruct ARM to restart it after it terminates:

```
f bbo5acr,cancel,armrestart
```

Modifying trace variables

You can use the modify command to dynamically alter trace variables.

Establishing a general level of trace: To establish a general trace level for the server, use the following command :

```
f server,traceall=n
```

Valid trace levels are 0 (none), 1 (exception), 2 (basic), and 3 (detailed tracing). Under normal conditions and in production, use 1 (exception).

Example: The following command will turn on exception level tracing for the bbo5acr server:

```
f bbo5acr,traceall=1
```

Here is a sample display:

```
F BBO5ACR,TRACEALL=1
BBOO0211I MODIFY COMMAND TRACEALL=1 COMPLETED SUCCESSFULLY
```

Setting basic and detailed trace levels for WebSphere for z/OS components: To specify the WebSphere for z/OS components for which you want to switch on a basic level of tracing, use the following command:

```
f server,tracebasic=(n,...)
```

Example 1: The following command will turn on a basic level of tracing for the Daemon component on server bbo5acr:

```
f bbo5acr,tracebasic=a
```

Here is a sample display:

```
F BBO5ACR,TRACEBASIC=A
BBOO0211I MODIFY COMMAND TRACEBASIC=A COMPLETED SUCCESSFULLY
```

To specify the WebSphere for z/OS components for which you want to switch on a detailed level of tracing use the following command:

```
f server,tracedetail=(n,...)
```

Example 2: The following command will turn on a detailed level of tracing for security on server bbo5acr:

```
f bbo5acr,tracedetail=e
```

Here is a sample display:

```
F BBO5ACR,TRACEDETAIL=E
BBOO0211I MODIFY COMMAND TRACEDETAIL=E COMPLETED SUCCESSFULLY
```

Setting specific trace points: To set specific trace points, use the following command:

```
f server,tracespecific=n | (n,...)
```

Example 1: The following command will turn on the specific trace point 04006001:

```
f bbo5acr,tracespecific=04006001
```

Example 2: The following command will turn on the specific trace points 04006001 and 04006027:

```
f bbo5acr,tracespecific=(04006001,04006027)
```

Example 3: The following command will turn on the specific trace points set in the environment variable 'tracepoints':

```
f bbo5acr,tracespecific='tracepoints'
```

Excluding specific trace points: To exclude specific trace points, use the following command:

```
f server,trace_exclude_specific=n | (n,...)
```

Example: The tracespecific command below turns on tracing for a whole part and then the trace_exclude_specific turns off tracing for the point 04006031 within that part.:

```
f bbo5acr,tracespecific=04006000
```

```
f bbo5acr,trace_exclude_specific=04006031
```

Resetting to the initial trace settings: To reset to the initial trace settings use the following command:

```
f server,traceinit
```

Turning off tracing: To turn off tracing, use the following command:

```
f server,tracenone
```

Sending the trace to sysprint: To send the trace to sysprint, use the following command:

```
f server,tracetosysprint=yes
```

To stop sending the trace to sysprint, use the following command:

```
f server,tracetosysprint=no
```

Modifying the Java trace string

To modify the Java trace string, use the following command:

```
f server,tracejava='trace specification'
```

Example: The following command enables all types of tracing for all registered trace components:

```
f bbo5acr,tracejava='*=enabled=all'
```

Note: The single quotes are a required part of the syntax.

Using the modify command to display information

You can use the modify command to display status of various server components and activities, including:

- Active controllers
- Trace settings
- Servants
- Sessions
- JVM Heap
- Java trace

Displaying status of a server: To display the status of a server, use the following command:

```
f server,display
```

Example: The following command will display status for the bbo5acr server:

```
f bbo5acr,display
```

Here is a sample display:

```
F BBO5ACR,DISPLAY
BBOO0173I SERVER BBO5SR4/BBOASR4A ACTIVE ON SY1 AT LEVEL wd5xo03.
BBOO0188I END OF OUTPUT FOR COMMAND DISPLAY
```

Displaying status of clusters: To display the status of clusters in the active controller, use the following command:

```
f server,display,servers
```

Example: The following command will display status for the bbo5acr server:

```
f bbo5acr,display,servers
```

Here is a sample display:

```
F BBO5ACR,DISPLAY,SERVERS
BBOO0182I SERVER             ASID   SYSTEM    LEVEL
BBOO0183I CBDAEMON/DAEMON01    31x  SY1       wd5xo03
BBOO0183I BBO5SR4 /BBOASR4A   1F6x  SY1       wd5xo03
BBOO0188I END OF OUTPUT FOR COMMAND DISPLAY,SERVERS
```

Displaying servants: To display servants, use the following command:

```
f server,display,servants
```

Example: The following command will display servants for the bbo5acr server:

```
f bbo5acr,display,servants
```

Here is a sample display:

```
F BBO5ACR,DISPLAY,SERVANTS
BBOO0185I SERVER BBO5SR4/BBOASR4A HAS 1 SERVANT PROCESS (ASID: 0038x)
BBOO0188I END OF OUTPUT FOR COMMAND DISPLAY,SERVANTS
```

Displaying trace settings and Java string trace settings: To display the trace settings for a server instance, use the following command:

```
f server,display,trace
```

Example 1: The following command will display trace settings for the bbo5acr server:

```
f bbo5acr,display,trace
```

Here is a sample display:

```
F BBO5ACR,DISPLAY,TRACE
BBOO0224I TRACE INFORMATION FOR SERVER BBO5SR4/BBOASR4A/STC00047
BBOO0197I LOCATION = SYSPRINT BUFFER
BBOO0197I AGGREGATE TRACE LEVEL = 1
BBOO0197I EXCEPTION TRACING = RAS(0), Common Utilities(1), COMM(3),
ORB(4), OTS(6), Shasta(7), OS/390 Wrappers(9), Daemon(A), Security(E),
Externalization(F), JRAS(J), J2EE(L)
BBOO0197I BASIC TRACING =
BBOO0197I DETAILED TRACING =
BBOO0197I TRACE SPECIFIC = NONE SPECIFIED
```

```
BBOO0197I TRACE EXCLUDE SPECIFIC = NONE SPECIFIED
BBOO0225I TRACE INFORMATION FOR SERVER BBO5SR4/BBOASR4A/STC00047 COMPLETE
BBOO0188I END OF OUTPUT FOR COMMAND DISPLAY,TRACE
```

To display the Java trace string settings for a server, use the following command:

```
f server,display,trace,java
```

Example 2: The following command will display Java trace settings for the bbo5acr server:

```
f bbo5acr,display,trace,java
```

Here is a sample display:

```
F BBO5ACR,DISPLAY,TRACE,JAVA
BBOO0196I TRACE INFORMATION FOR SERVER BBO5SR4/BBOASR4A
BBOJ0050I CTL(STC00047):*=all=disabled
BBOO0188I END OF OUTPUT FOR COMMAND DISPLAY,TRACE,JAVA
```

To display the Java trace information for all servants, one at a time.

```
f server,display,trace,java.srs
```

Example 3: The following command will display Java trace settings for the bbo5acr server:

```
f bbo5acr,display,trace,java,srs
```

Here is a sample display:

```
F BBO5ACR,DISPLAY,TRACE,JAVA,SRS
BBOO0196I TRACE INFORMATION FOR SERVER BBO5SR4/BBOASR4A
BBOJ0050I SR(STC00048):*=all=disabled
BBOO0188I END OF OUTPUT FOR COMMAND DISPLAY,TRACE,JAVA,SRS
```

To display the Java trace information for the controller and all servants, one at a time.

```
f server,display,trace,java.all
```

Example 4: The following command will display Java trace settings for the controller and all servants:

```
f bbo5acr,display,trace,java,all
```

Here is a sample display:

```
F BBO5ACR,DISPLAY,TRACE,JAVA,ALL
BBOO0196I TRACE INFORMATION FOR SERVER BBO5SR4/BBOASR4A
BBOJ0050I CTL(STC00047):*=all=disabled
BBOJ0050I SR(STC00048):*=all=disabled
BBOO0188I END OF OUTPUT FOR COMMAND DISPLAY,TRACE,JAVA,ALL
```

Displaying JVM heap information: To display the JVM heap information for a server controller, use the following command:

```
f server,display,jvmheap
```

Example 1: The following command will display JVM heap information for the bbo5acr server:

```
f bbo5acr,display,jvmheap
```

Here is a sample display:

```
F BBO5ACR,DISPLAY,JVMHEAP
BBOO0201I JVM HEAP INFORMATION FOR SERVER BBO5SR4/BBOASR4A/STC00047
BBOO0202I (STC00047) HEAP(MIDDLEWARE), COUNT(00000000), FREE STORAGE(
396FA70), TOTAL STORAGE( 7FFFA00)
BBOO0204I JVM HEAP INFORMATION FOR SERVER BBO5SR4/BBOASR4A/STC00047 COMPLETE
BBOO0188I END OF OUTPUT FOR COMMAND DISPLAY,JVMHEAP
```

To display the JVM heap information for all servants one at a time, use the following command:

```
f server,display,jvmheap,srs
```

Example 2: The following command will display JVM heap information for all servants in the bbo5acr server:

```
f bbo5acr,display,jvmheap,srs
```

Here is a sample display:

```
F BBO5ACR,DISPLAY,JVMHEAP,SRS
+BBOO0201I JVM HEAP INFORMATION FOR SERVER BBO5SR4/BBOASR4A/STC00048
+BBOO0202I (STC00048) HEAP(MIDDLEWARE), COUNT(00000001), FREE STORAGE(
 25F4030), TOTAL STORAGE( 7FFFA00)
+BBOO0204I JVM HEAP INFORMATION FOR SERVER BBO5SR4/BBOASR4A/STC00048 COMPLETE
BBOO0188I END OF OUTPUT FOR COMMAND DISPLAY,JVMHEAP,SRS
```

To display the JVM heap information for the controller and all servants of a server one at a time, use the following command:

```
f server,display,jvmheap,all
```

Example 3: The following command will display JVM heap information for the controller and all servants in the bbo5acr server:

```
f bbo5acr,display,jvmheap,all
```

Here is a sample display:

```
F BBO5ACR,DISPLAY,JVMHEAP,ALL
BBOO0201I JVM HEAP INFORMATION FOR SERVER BBO5SR4/BBOASR4A/STC00047
BBOO0202I (STC00047) HEAP(MIDDLEWARE), COUNT(00000000), FREE STORAGE(
396FA70), TOTAL STORAGE( 7FFFA00)
BBOO0204I JVM HEAP INFORMATION FOR SERVER BBO5SR4/BBOASR4A/STC00047 COMPLETE
+BBOO0201I JVM HEAP INFORMATION FOR SERVER BBO5SR4/BBOASR4A/STC00048
+BBOO0202I (STC00048) HEAP(MIDDLEWARE), COUNT(00000001), FREE STORAGE(
 25F4030), TOTAL STORAGE( 7FFFA00)
+BBOO0204I JVM HEAP INFORMATION FOR SERVER BBO5SR4/BBOASR4A/STC00048 COMPLETE
BBOO0188I END OF OUTPUT FOR COMMAND DISPLAY,JVMHEAP,ALL
```

Displaying sessions: To display sessions, use the following command:

```
f server,display,sessions
```

Example 1: Since "listeners" is the default, either of the following command will display session listeners for the bbo5acr server:

```
f bbo5acr,display,sessions
```

or

```
f bbo5acr,display,sessions,listeners
```

Here is a sample display:

```
F BBO5ACR,DISPLAY,SESSIONS,LISTENERS
BBOO0189I PROTOCOLS AND LISTENING PORTS FOR SERVER BBO5SR4/BBOASR4A
BBOO0190I TCPIIOP     2809
BBOO0190I SSLIIOP     6000
```

```
BBOO0190I HTTP       9088
BBOO0190I HTTPS      443
BBOO0188I END OF OUTPUT FOR COMMAND DISPLAY,SESSIONS,LISTENERS
```

To display server sessions counts, use the following command:

```
f server,display,sessions,server
```

Example 2: The following command will display the server session counts for all protocols used by the bbo5acr server:

```
f bbo5acr,display,sessions,server
```

Here is a sample display:

```
F BBO5ACR,DISPLAY,SESSIONS,SERVER
BBOO0191I SERVER SESSION COUNTS FOR PROTOCOLS USED BY SERVER
BBO5SR4/BBOASR4A
BBOO0192I TCPIIOP    2
BBOO0192I LOCALIIOP  0
BBOO0192I SSLIIOP    0
BBOO0192I HTTP       0
BBOO0192I HTTPS      0
BBOO0188I END OF OUTPUT FOR COMMAND DISPLAY,SESSIONS,SERVER
```

You can further refine the modify command to display, or list, the number of sessions used by a specific protocol. The available options are:

- tcpiiop
- localiiop
- ssliiop
- http
- https

In addition, you can also specify to display the results or list the results by leaving off the list modifer or including the list modifier, respectively.

Example 1: The following command will display the number of sessions using the TCP/IP IIOP protocol on the bbo5acr server:

```
f bbo5acr,display,sessions,server,tcpiiop
```

Here is a sample display:

```
F BBO5ACR,DISPLAY,SESSIONS,SERVER,TCPIIOP
BBOO0193I SERVER BBO5SR4/BBOASR4A HAS 2 TCPIIOP   SERVER SESSIONS
BBOO0188I END OF OUTPUT FOR COMMAND DISPLAY,SESSIONS,SERVER,TCPIIOP
```

Other WebSphere for z/OS operations

ARM and restart

There are some things to remember when using ARM to restart your servers:

1. If you are ARM-enabled and you cancel or stop a server, it will restart in place using the "armrestart" command described previously.
2. If you start the Daemon on a system that already has a deamon, it will terminate.
3. Every other server will come up on a dynamic port unless the configuration has a fixed port. Therefore, the fixed ports must be unique in a sysplex.

Setting up error log streams for different servers and server instances

See the *WebSphere Application Server for z/OS V5.0: Installation and Customization*, GA22-7909, for instructions on how to set up and read the error log stream for different servers.

When you encounter errors, the error log can offer lots of useful information. For instructions on how to access the error log, see the Troubleshooting section of the WebSphere Application Server for z/OS V5.0 InfoCenter, access to which can be obtained through the WebSphere for z/OS library Web site http://www.ibm.com/software/webservers/appserv/zos_os390/library.html.

Using Workload Management for WebSphere for z/OS operations

This section describes general Workload Management (WLM) tasks you may use to operate WebSphere for z/OS.

WLM application environments are classified as either ″static″ or ″dynamic.″ The environments exist independently of each other, which allows you to define one of each environment with the same name on the same system. Because there are separate classifications, you need to put certain options in your WLM commands to target the correct one. Here is a quick breakdown of each.

Static options for WLM commands
WLM commands with the static option affect only static application environments, which are defined by iwmarin0. Commands with the static option, like their static application environment counterparts, are sysplex in scope. This means that if you quiesce an application environment, it is on all systems in the sysplex. However, statically defined commands do not affect dynamically defined application environments over a sysplex-wide scope.

Dynamic options for WLM commands
WLM commands with the dynamic option affect only dynamic application environments, which are defined by an address space at runtime using WLM procedure calls. Commands with the dynamic option, like their dynamic application environment counterparts, have only a system scope.

Displaying the status of a WLM application environment

This section shows you how to display the status of a static or dynamic WLM application environment.

Note: The WLM application environment name is the same as the generic server name.

See *z/OS MVS System Commands Summary*, SA22-7628, for more information about the display command.

Displaying the status of a static application environment: Display the status of a static application environment and its state with the `d wlm,applenv=[name|*]` command:

d wlm,applenv=[*name*|*]
Displays the status for static application environments. Insert a specific application name or ″*″ to display them all.

Example: Here is an example that displays all static application environments:

```
- SY1     d wlm,applenv=*
   SY1     IWM029I 19.02.29 WLM DISPLAY 341
  APPLICATION ENVIRONMENT NAME   STATE     STATE DATA
  BR14                           AVAILABLE
  FRESCA                         AVAILABLE
  PAYAT2                         AVAILABLE
  PAYAT5                         AVAILABLE
  PAYROLL                        AVAILABLE
  PAYTEST                        AVAILABLE
  PAYTEST2                       AVAILABLE
```

Example: Here is an example that displays the specific static application environment db2pay:

```
0- SY1     d wlm,applenv=db2pay
   SY1     IWM029I 18.59.22 WLM DISPLAY 319
  APPLICATION ENVIRONMENT NAME   STATE      STATE DATA
  DB2PAY                         AVAILABLE
  ATTRIBUTES:PROC=BR14      SUBSYSTEM TYPE:CB
```

The `PROC` in the above example is the JCL PROC that WLM uses to start the server region.

Displaying the status of a dynamic application environment: Display the status of a dynamic application environment and its state with the `d wlm,dynappl=[`*name*`|*],[`*options*`]` command:

d wlm,dynappl=[*name***|*],[***options***]**

Displays the status for dynamic application environments. Insert a specific application name or "*" to display them all.

The optional parameters, which allow you to further qualify the dynamic application environment of interest, are as follows:

STYPE

The subsystem type

SNAME

The subsystem name

SNODE

The node name

Example: Here is an example that displays all dynamic application environments:

```
- SY1     d wlm,dynappl=*
   SY1     IWM029I 19.02.29 WLM DISPLAY 341
  DYNAMIC APPL. ENVIRON. NAME    STATE     STATE DATA
  BBOASR4                        AVAILABLE
  BBOSYSM                        AVAILABLE
```

Example: Here is an example that displays the specific dynamic application environment bboasr4:

```
0- SY1     d wlm,dynappl=bboasr4
   SY1     IWM029I 18.59.22 WLM DISPLAY 319
  DYNAMIC APPL. ENVIRON. NAME    STATE     STATE DATA
  BBOASR4                        AVAILABLE
  ATTRIBUTES:PROC=BR14        SUBSYSTEM TYPE:CB
  SUBSYSTEM NAME:WEBA         NODENAME:LOTUS
```

. . .

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:

```
BBOO0075E 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 of the `VARY WLM` command.

Note: The application environment is sysplex in scope. This means that when WLM stops the application environment with a static command, it is stopped across the entire sysplex. When you resume, it is resumed on every system in the sysplex. This does not apply if you use a dynamic command, in which case your changes are only system-wide.

WLM shuts down the application environment because server regions are failing. The system cannot determine why, so it shuts down the environment and requests help from the operator.

Note: If it does not do this, it continues in a failure loop (known as a *storm drain*). The failing system appears like it is performing well because the transactions are ending quickly, but they are actually failing.

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 one of the following commands, depending on your application environment type:
   - Static application environment:
     ```
     d wlm,applenv=*
     ```
   - Dynamic application environment:
     ```
     d wlm,dynappl=*,options
     ```

   Note: See "Displaying the status of a dynamic application environment" on page 44 for a description of the options.

2. To start the application environment, issue one of the following commands, depending on your application environment type:
   - Static application environment:
     ```
     v wlm,applenv=environment_name,resume
     ```
   - Dynamic application environment:
     ```
     v wlm,dynappl=environment_name,resume,options
     ```

   Note: See "Displaying the status of a dynamic application environment" on page 44 for a description of the options.

   where `environment_name` is the application environment name.

You know you are done when a message is issued to the console stating that the application environment was resumed.

Note: IBM does not support the following WLM command:

```
V WLM,APPLENV=environment_name,REFRESH
```

Getting out of the stopped state and back to the available state

If a server region goes down in your sysplex and you get into the stopped state, it is important to understand that the stopped state is sysplex-wide for an application environment. The state is at the server level, not the instance level. If this occurs, WLM won't be able to start additional server regions to finish the work in progress. WLM can't tell if it is a runtime error or an environment problem. Since all systems in your sysplex are identical, if you have this problem on one system, it may happen on the other system, even if you haven't seen it yet.

Steps for getting out of the stopped state and back to the available state: Perform the following steps to get a system back to the available state from the stopped state.

1. Determine why the system is in the stopped state. This could be due to any of the following:
   - A JCL error in the server region proc
   - A WebSphere for z/OS runtime bug
   - An application bug
   - Other environmental problems
   - Other MVS-related problems.

   If you are unable to determine the cause of the problem, contact the IBM Support Center.

   ____
2. Resolve the problem.

   ____
3. Resume operations by issuing the resume command.

   The WLM resume command will resume WLM, in turn starting the server regions. However, if you haven't fixed the problem, the application environment will most likely return back to the stopped state. Resume will get you back to the available state only if you fixed the problem.

   ```
   v wlm,applenv=applenvname,resume
   ```

   ____
4. Issue the following command:

   ```
   d wlm,applenv=applenv
   ```

   to see if the application environment is active.

Chapter 4. Operational considerations for z/OS or OS/390 subsystems

This chapter describes helpful hints for when customizing z/OS or OS/390 subsystems that are required or recommended for WebSphere for z/OS.

Note: These are considerations for the System Programmers that are responsible for these subsystems, not instructions for installing or configuring these subsystems.

Operational considerations for z/OS or OS/390 subsystems

DB2 Universal Database for z/OS and OS/390 operations

Guidelines for DB2 Universal Database for z/OS and OS/390 operations: This section provides guidelines and tips for DB2 operations. See the *WebSphere Application Server for z/OS V5.0: Installation and Customization* for details on how WebSphere for z/OS uses DB2.

- When you create and commit conversations, you may see a lot of DB2 activity as DB2 is offloading data. In some cases you may need to add another log volume or clean up the logs. Check the size of your DB2 logs. If the DB2 log runs out of space, the program will stop and you will need to add additional log data sets.
- When your configurations get larger, you may need to increase your default buffers. See *WebSphere Application Server for z/OS V5.0: Installation and Customization*, for recommendations for increasing the buffer pools. Check the 32K temporary work space for DB2. The work space should have been allocated during the WebSphere for z/OS installation. If it is not large enough, however, you may get an SQLCODE -904 return code when bringing up the LDAP server, the System Administration Server, or the Naming Server.
- When displaying DB2 threads, the correlation ID is equal to the MVS user ID of the requesting unit of work.

IMS operations

This section provides guidelines and tips for IMS operations as they relate to use with the WebSphere for z/OS-supported IMS connectors. Also see *IMS/ESA Operations Guide*, SC26-8741, for IMS operations guidelines as they relate to WebSphere for z/OS.

- When using the IMS Connector for Java to take advantage of two-phase commit processing, or using IMS PAA support, you need to set up a significant number of message processing regions to handle the total number of IMS transactions that might be issued in a WebSphere for z/OS transaction. One WebSphere for z/OS transaction could drive three or more transactions to IMS. To successfully process these transactions, IMS may need additional message processing regions available to handle the request. In general, IMS needs the same number of started message processing regions as the number of generated IMS transactions. See *WebSphere Application Server for z/OS V5.0: Installation and Customization* for more information.
- You may also need to set the IMS PARMLIM option if it wasn't set at installation time:

  ```
  assign parlim 0 tran tranname
  ```

Assigning a parallel limit value of "0" indicates that there is no limit on the number of transactions you can run at one time. This can also be specified at IMS generation during configuration time using the TRANSACT statement. This value allows transactions to be scheduled in multiplicity (or parallel) so that more transactions can run at the same time. You need to set this value in addition to setting up multiple message processing regions to make IMS work properly. See *WebSphere Application Server for z/OS V5.0: Installation and Customization*, for more information.

- **IMS OTMA support**: The IMS control region has to be started with the OTMA interfaces active. The IMS procedure parameter should specify OTMA=YES for WebSphere for z/OS to make the connection to the right IMS. You also need to define the XCF group name for the OTMA interfaces that IBM supports. See the instructions in *WebSphere Application Server for z/OS V5.0: Installation and Customization*, for details about setting up the IMS-OTMA Procedural Application Adapter.
- **IMS OTMA support runtime tip:** If you have to re-IPL your system, you must redefine the IMS OTMA SVC (supervisor call). You can do this by running the exec statement PGM=DFSYSVI0, which dynamically installs the SVC needed. If you don't redefine it, when you try to run IMS OTMA support, you will get an F92 abend and the server will come down.

RRS operations

See Table 6 on page 55 and *z/OS MVS Programming: Resource Recovery*, SA22-7616, for RRS operations guidelines.

Tips for RRS operations:

- If you have configured your logstreams to the coupling facility, then monitor your log streams to ensure offload is not occurring. RRS will perform better if its recovery logs do not offload.

  Note: Proper sizing of the RRS logs is important. Too small and you get reduced throughput since logger is offloading the logs too frequently. Too large and you could overflow your coupling facility.
- Keep the main and delayed (only contains active or live data) logs in your coupling facility. Make sure the CF definitions don't overflow.

  Note: A commit cannot occur until the log record is written.
- Until you stabilize your workloads, it is a good idea to use the archive log. If you have an archive log configured, RRS will unconditionally use it. However, there is a performance penalty for using it.

Workload Management (WLM) operations

See *z/OS MVS Planning: Workload Management*, SA22-7602 and *z/OS MVS Programming: Workload Management Services*, SA22-7619, for WLM operations guidelines.

Chapter 5. WebSphere for z/OS backup guidelines and procedures

This chapter describes WebSphere for z/OS backup guidelines and procedures.

Guidelines for backup of the WebSphere for z/OS system

Use the following guidelines to back up parts of your WebSphere for z/OS system:

1. Be sure to back up the RMDATA log for RRS. Otherwise, a failure could force you to do a cold start of RRS.
2. Set the ARCHIVE log retention period to one day.
3. Incorporate the following in your normal backup procedures:
   - WebSphere for z/OS proclibs
   - WebSphere for z/OS loadlibs
   - The directory where WebSphere for z/OS run-time information is written (the default is `/WebSphere/V5R0M0`).
4. Back up your own application executables, databases, and bindings.
5. If you wish to back up a single server, you can use the export/import function in the Administrative Console. For details on how to do this, see the assembling applications information in the z/OS view of the WebSphere Application Server InfoCenter, which you can access via the WebSphere for z/OS library Web site.

Chapter 6. Monitoring and recovering WebSphere for z/OS and dependent subsystems

Reviewers Note:

No changes have been made to this chapter. I will need to review it with someone who understands this for V5.

This chapter describes how to monitor and recover WebSphere for z/OS and its dependent subsystems.

Startup order for WebSphere for z/OS and dependent subsystems

The following table shows the order in which you need to bring up the dependent subsystems for WebSphere for z/OS. See *WebSphere Application Server for z/OS V5.0: Installation and Customization*, GA22-7909, for more information about the order in which you need to start dependent z/OS or OS/390 subsystems.

Note: When subsystems are shown on the same line, it indicates that they can be started at the same time or in any order.

Startup order	**Subsystems to start**		
1	Workload Manager (WLM) *automatically started*		
2	RACF *automatically started*		
3	System Logger *automatically started*		
4	RRS	Unix System Services (doesn't complete until TCP/IP is done, but can be started before TCP/IP)	
5	VTAM (optional)	TCP/IP	
6	APPC (optional)	TSO	
7	DB2 (optional)	CICS and CICS Transaction Gateway	IMS and IMS Connect
8	LDAP (optional)	NFS (only required in a SYSPLEX) **Note:** NFS is not required at all if using a shared HFS.	

Startup order	Subsystems to start		
9	WebSphere for z/OS • JVM: Started inside WebSphere for z/OS and WebServer • LE: Started inside WebSphere for z/OS and WebServer • Base Application Server: MVS operator starts the controller, which in turn starts the daemon, and WLM starts the servant regions • Network Deployed configuration: MVS operator starts the Node Agent which starts the Daemon. The MVS operator then starts the Deployment manager which can be then used to start the Application server controller regions through the administrative console.	WebServer (Servlet)	

Automation and recovery scenarios and guidelines

The following section provides information on how to monitor and recover WebSphere for z/OS and the subsystems it uses. It provides startup, shutdown, and recovery procedures and scenarios. It also tells you how to determine if the subsystems are up or down, and tells you where to find more information.

APPC automation and recovery scenarios

Table 3. APPC automation and recovery scenarios

Task	APPC automation and recovery scenarios
Startup	APPC should be started before WebSphere for z/OS. In theory, WebSphere for z/OS *could* be started before APPC, but only as long as no objects get dispatched in containers that have an IMS APPC LRMI associated with them. If APPC is not up before WebSphere for z/OS, and you want to use an APPC connector to talk to IMS, then you will have no connectivity. APPC/MVS does not have to be up for CICS. APPC does not have to be started after VTAM.
Shutdown	Reverse the startup procedure. Shutdown WebSphere for z/OS, APPC, then VTAM.

Table 3. APPC automation and recovery scenarios (continued)

Task	APPC automation and recovery scenarios
Handling in-flight or indoubt transactions if there is a failure	If you are using APPC for communications and it fails, do the following: 1. Shutdown all servers with APPC connectivity. 2. Restart APPC (if it totally failed). 3. Restart the WebSphere for z/OS server. **Note:** APPC resyncs itself. If your transaction is indoubt, IMS waits until you restart APPC. IMS relies on RRS for recovery. RRS will resolve indoubts by handshaking with every subsystem it was communicating with before it went down. If you are using CICS, note that CICS has its own coordinator.
How to determine if APPC is running	Issue the `DISPLAY APPC,LU,ALL` command. If APPC is not active, it will say so. In addition, the status of the LUs used by WebSphere for z/OS and/or IMS should be active or no APPC work will be successful.
What happens to WebSphere for z/OS if APPC goes down?	Any objects attempting to use the IMS APPC PAA will not work. The server region running on behalf of the container attempting to use APPC will likely get a `C9C24C05` error, indicating that an APPC ALLOCATE request was attempted and failed. Additional APPC error diagnostic information that helps to pinpoint the APPC problem is contained in the logs associated with this region.
What happens to other subsystems if APPC goes down?	Not applicable
Where to find more information	• *z/OS MVS Planning: Operations*, SA22-7601 • *z/OS MVS Planning: APPC/MVS Management*, SA22-7599 • *z/OS MVS Programming: Resource Recovery*, SA22-7616

WLM automation and recovery scenarios

Table 4. Workload Manager (WLM) automation and recovery scenarios

Task	WLM automation and recovery scenarios
Startup	WLM is automatically started by z/OS or OS/390 when you IPL your system. You don’t have to start it.
Shutdown	You cannot shutdown WLM.
Handling in-flight and indoubt transactions if there is a failure	Not applicable
How to determine if WLM is running	Not applicable
What happens to WebSphere for z/OS if WLM goes down?	Not applicable
What happens to other subsystems if WLM goes down?	Not applicable

Table 4. Workload Manager (WLM) automation and recovery scenarios (continued)

Task	WLM automation and recovery scenarios
How to handle a catostrophic failure of the WebSphere for z/OS server regions	Following a catastrophic failure of the WebSphere for z/OS server regions, you can use one of the following resume commands, depending on your application environment type: • Static application environment: `v wlm,applenv=`*`applenvname`*`,resume` • Dynamic application environment: `v wlm,applenv=`*`applenvname`*`,resume,`*`options`* **Note:** See “Displaying the status of a dynamic application environment” on page 44 for a description of the options.
Where to find more information	• *z/OS MVS Planning: Workload Management*, SA22-7602 • *z/OS MVS Programming: Workload Management Services*, SA22-7619

RACF automation and recovery scenarios

Table 5. RACF automation and recovery scenarios

Task	RACF automation and recovery scenarios
Startup	If it is installed, RACF is started as a part of IPL.
Shutdown	RACF is not shutdown.
Handling in-flight and indoubt transactions if there is a failure	Not applicable
How to determine if RACF is running	Use the `RACF SETROPTS` command to display the status of RACF.
What happens to WebSphere for z/OS if RACF goes down?	RACF goes into fail safe mode. This means that for every resource that is accessed, the operator is asked to verify if it is okay. In general, the system is IPLed if this occurs.
What happens to other subsystems if RACF goes down?	It depends on what subsystem and how RACF fails.
Where to find more information	• *z/OS Security Server RACF System Programmer's Guide*, SA22-7681 • *z/OS Security Server RACF Security Administrator's Guide*, SA22-7683

RRS automation and recovery scenarios

Table 6. RRS automation and recovery scenarios

Task	RRS automation and recovery scenarios
Startup	Ensure System Logger has been started before RRS. **Note:** RRS will display error messages indicating that System Logger must be started first if you try to start RRS without starting System Logger.Ensure RRS is started before WebSphere for z/OS. RRS does not start by itself. RRS will start automatically only if it was registered with the Automatic Restart Manager (ARM) and if ARM is running. To start RRS, issue the `start` command: `start atrrrs,sub=master` **Note:** RRS doesn't restart itself if you issue the cancel command, so you need to restart it manually if it was canceled or if ARM isn't running.
Shutdown	Shutdown RRS in the reverse order that you started RRS. Shutdown WebSphere for z/OS, then RRS, followed by System Logger. There is no controlled way to bring down RRS. The best approach is: 1. Quiesce WebSphere for z/OS. 2. Shutdown WebSphere for z/OS. 3. Cancel RRS. **Note:** You may want to bring down the DB2 you are using for WebSphere for z/OS before canceling RRS. To cancel RRS, issue the command: `setrrs cancel`
Handling in-flight and indoubt transactions if there is a failure	Refer to the RRS system management panels to display in-flight and resolve indoubt transactions. You can display the resource managers on the RM panels in RRS, display all units of recovery (UR), filter the URs, and then resolve the indoubts. You cannot resolve in-flights. You can display all RRS-managed transactions. If you are using the IMS Connector for Java, this process applies only if IMS Connector for Java, IMS Connect, and the IMS subsystem have been configured locally on the same z/OS or OS/390 system image on which the WebSphere for z/OS J2EE server runs. The local configuration is the only configuration in which IMS Connector for Java runs as an RRS-transactional connector.
How to determine if RRS is running	Use the display command: `d a,atrrs` `atrrs` is the name of the default RRS proc shipped with WebSphere for z/OS. Use the procname that you use to start RRS. The address space comes from the proc.
What happens to WebSphere for z/OS if RRS goes down?	RRS is a required subsystem, so WebSphere for z/OS can't run without it. If RRS goes down, WebSphere for z/OS will get fatal errors. You need to get RRS started, then restart WebSphere for z/OS.
What happens to other subsystems if RRS goes down?	RRS is the z/OS or OS/390 transaction monitor. If you cancel RRS, you will have problems with any subsystems using it (for example, WebSphere for z/OS, DB2, IMS). Ensure you understand the implications before you cancel RRS.

Table 6. RRS automation and recovery scenarios (continued)

Task	RRS automation and recovery scenarios
Where to find more information	*z/OS MVS Programming: Resource Recovery*, SA22-7616

Unix System Services automation and recovery scenarios

Table 7. UNIX System Services automation and recovery scenarios

Task	Unix System Services automation and recovery scenarios
Startup	Unix System Services is a permanent component of MVS and is started automatically at IPL time.
Shutdown	Unix System Services does not support a shutdown capability, so it is always available.
Handling in-flight or indoubt transactions if there is a failure	The only data that could be considered transactional in nature is data stored in the HFS.
How to determine if Unix System Services is running	Unix System Services is always available as long as the system is up and running.
What happens to WebSphere for z/OS if Unix System Services goes down?	If Unix System Services fails, the system must be re-IPLed. WebSphere for z/OS will get errors and terminate.
What happens to other subsystems if Unix System Services goes down?	If Unix System Services fails, the system must be re-IPLed.
Where to find more information	*z/OS UNIX System Services Planning*, GA22-7800

TCP/IP automation and recovery scenarios

Table 8. TCP/IP automation and recovery scenarios

Task	TCP/IP automation and recovery scenarios
Startup	TCP/IP must be up before starting WebSphere for z/OS.
Shutdown	Shutdown WebSphere for z/OS before shutting down TCP/IP.
Handling in-flight or indoubt transactions if there is a failure	Methods in flight will have their transactions rolled back when the attempt to send a response to the method fails. Other transactions will wait for a timeout.
How to determine if TCP/IP is running	Use the display command looking for the TCP/IP proc.
What happens to WebSphere for z/OS if TCP/IP goes down?	If TCP/IP goes down, then WebSphere for z/OS on the system must be restarted. You will get an SVC dump because the socket layer was destroyed.
What happens to other subsystems if TCP/IP goes down?	If TCP/IP goes down, sessions break and transactions react as described above. **Note:** WebSphere for z/OS, which cannot recognize when TCP/IP comes back up, must be restarted. **Note:** If TCP/IP goes down, you should recycle LDAP before restarting WebSphere for z/OS.

DB2 automation and recovery scenarios

Table 9. DB2 automation and recovery scenarios

Task	**DB2 automation and recovery scenarios**
Startup	DB2 is started after RRS but before LDAP, NFS, and WebSphere for z/OS.
Shutdown	Reverse of startup sequence.

Handling in-flight or indoubt transactions if there is a failure	Use the RRS panels to resolve. See *z/OS MVS Programming: Resource Recovery*, SA22-7616. The RRS panels are the preferred way to resolve DB2 indoubts because they allow you to view all resource managers that have an interest in the transaction. However, you can also use DB2 to resolve indoubts. You can issue the command:
	`DISPLAY THREAD(*) TYPE(INDOUBT)`
	to display DB2 information about the indoubt threads it knows about (if there are too many, you can go into S.LOG to view the information). This display will give you a DB2 identifier called a ″nid″. Copy the nid and paste it into this command:
	`-RECOVER INDOUBT (RRSAF) ACTION(COMMIT)`
	` NID(B1D379D17ED6CF900000009401010000)`
	where the `nid` is the one that you cut from the display command. You can issue this command to roll back the transaction:
	`-RECOVER INDOUBT (RRSAF) ACTION(ABORT)`
	` NID(B1D379D17ED6CF900000009401010000)`

How to determine if DB2 is running	Use the display command to display the DB2 address space.
What happens to WebSphere for z/OS if DB2 goes down?	WebSphere for z/OS needs DB2 to run, so it abends if DB2 goes down. You need to restert the LDAP server, then restart DB2, and then restart WebSphere for z/OS.
What happens to other subsystems if DB2 goes down?	Not applicable
Where to find more information	See the DB2 books under “Where to find related information, tools, and supplements” on page x.

CICS automation and recovery scenarios

Table 10. CICS automation and recovery scenarios

Task	**CICS automation and recovery scenarios**
Startup	CICS and any required CICS products, such as CICS Transaction Gateway, need to be properly installed, initialized, and started before any workflows to a CICS-enabled WebSphere for z/OS application control server region are run.
Shutdown	Shutdown the WebSphere for z/OS application control region that uses CICS as a backing store, then shutdown the CICS service.

Table 10. CICS automation and recovery scenarios (continued)

Task	CICS automation and recovery scenarios
Handling in-flight or indoubt transactions if there is a failure	If there is an error during processing, both CICS and WebSphere for z/OS rely on the underlying RRS subsystem to handle all rollback notifications to the registered interests. In the case of in-flight transactions, RRS will notify all participants that a rollback is required, and normal rollback processing will occur in each registered party. In the case of indoubt transactions, it may be necessary to recycle the WebSphere for z/OS Application Control/Server region to release any pending transaction in CICS.
How to determine if CICS is running	This is installation dependent.
What happens to CICS if WebSphere for z/OS goes down?	Should WebSphere for z/OS happen to go down, one of two situations could occur: 1. If WebSphere for z/OS and CICS are currently engaged in a unit of work, then RRS processing as described above would occur and it may be necessary to recycle the application control server regions to release pending transactional work in CICS. 2. If WebSphere for z/OS and CICS are not currently engaged in a unit of work, CICS is not affected.
What happens to other subsystems if CICS goes down?	Not applicable
Where to find more information	*CICS Operations and Utilities Guide*, SC34-5991

IMS automation and recovery scenarios

Table 11. IMS automation and recovery scenarios

Task	IMS automation and recovery scenarios
Startup	IMS and any required IMS products, such as IMS Connect, need to be properly installed, initialized, and started before any workflows to an IMS-enabled WebSphere for z/OS application control server region are run.
Shutdown	Shutdown the WebSphere for z/OS application Control Region which uses IMS as a backing store, then shutdown the IMS service
Handling in-flight or indoubt transactions if there is a failure	If there is an error during processing, both IMS and WebSphere for z/OS rely on the underlying RRS subsystem to handle all rollback notifications to the registered interests. In the case of in-flight transactions, RRS will notify all participants that a rollback is required and normal rollback processing will occur in each registered party. In the case of indoubt transactions, it may be necessary to recycle the WebSphere for z/OS Application Control/Server region to release any pending transaction in the IMS MPRs.
How to determine if IMS is running	This is installation-dependent.

Table 11. IMS automation and recovery scenarios (continued)

Task	IMS automation and recovery scenarios
What happens to IMS if WebSphere for z/OS goes down?	Should WebSphere for z/OS happen to go down, one of two situations could occur: 1. If WebSphere for z/OS and IMS are currently engaged in a unit of work, then RRS processing as described above would occur and it may be necessary to recycle the application control server regions to release pending transactional work in the IMS MPR. 2. If WebSphere for z/OS and IMS are not currently engaged in a unit of work, IMS is not affected.
What happens to other subsystems if IMS goes down?	Not applicable
Where to find more information	*IMS/ESA Operator's Reference*, SC26-8742

LDAP automation and recovery scenarios

Table 12. LDAP automation and recovery scenarios

Task	LDAP automation and recovery scenarios
Startup	LDAP, as used by WebSphere for z/OS, is completely run within the WebSphere for z/OS address spaces using something called ″the local backend.″ This support takes the front side of the LDAP client APIs and the backside database implementation and runs them completely inside the WebSphere for z/OS Naming Server and Interface Repository. For Naming and IR, OMVS and DB2 must be up *before* Naming and IR. To run the LDAP server, TCPIP, OMVS, and DB2 must all be up before the LDAP server. **Note:** There are two LDAP modes supported: 1. Local LDAP backend. 2. Remote LDAP Server: the WebSphere for z/OS environment has to be set up correspondingly, and DB2, TCP/IP, and the remote server have to be up and running before WebSphere for z/OS is started.
Shutdown	Shutdown Naming and IR, then OMVS and DB2. For the LDAP server, shutdown the LDAP server, then TCPIP and DB2, and then OMVS.
Handling in-flight or indoubt transactions if there is a failure	If there is a failure during processing, Naming and IR rely on RRS to issue a rollback directly to DB2 and, as a result, any work done by the LDAP code is rolled back along with it. For the LDAP server, AUTOCOMMIT is set to NO, causing any error to ROLLBACK for that transaction. This ensures the atomicity characteristic of LDAP operations.
How to determine if LDAP is running	In the case of WebSphere for z/OS, if Naming and IR are operating, then LDAP is operating. In the case of the LDAP server, and a started task is used for the LDAP server, use the SDSF to see if the started task is running. Examine the output log for the started task to see if any error messages were displayed. Alternatively, the `LDAPSRCH` command (from TSO), or `LDAPSEARCH` command (from Unix System Services shell) can be used to perform a simple search to verify that the LDAP server is running.

Table 12. LDAP automation and recovery scenarios (continued)

Task	LDAP automation and recovery scenarios
What happens to WebSphere for z/OS if LDAP goes down?	• In J2EE server regions, the LDAP server **must** be active since it is a separate server that no longer runs inside the WebSphere for z/OS region. Recycle the JNDI, then restart the J2EE application servers.
What happens to other subsystems if LDAP goes down?	Most z/OS or OS/390 subsystems do not depend on LDAP, but this may change in the future. In the case of accessing LDAP through the LDAP server, there is a way to configure the LDAP server to operate in a sysplex environment such that (using sysplex-enabled DNS) LDAP requests will be sent to the LDAP server in the sysplex that is operating (assuming that there is one). As an alternative, subsystems that want to use LDAP could configure a backup LDAP server to be contacted in case the primary server is not accessible. In this case, the application would assume that it could retrieve all of the same data that it could get from the backup on the primary which would be handled by some replication mechanism. The LDAP server currently supports a master/slave replication mechanism, but you could also try duplicating the sysplex server using DB2 data sharing.
Where to find more information	*z/OS Security Server LDAP Server Administration and Use*, SC24-5923

NFS automation and recovery scenarios

Note: NFS is used as a shared file system in OS/390 R8. Shared HFS is used in OS/390 R9 and above. The following comments relate to the runtime usage of NFS, not the application development time usage.

Table 13. NFS automation and recovery scenarios

Task	NFS automation and recovery scenarios
Startup	During Unix System Services filesystem initialization, NFS Client is started and run in an NFS colony address space. The FILESYSTYPE parmlib statement for NFS Client must be present in the SYS1.PARMLIB(BPXPRMxx) member.
Shutdown	To stop the NFS Client gracefully, the system operator could issue the modify operator command `STOP` with the OS/390 NFS Client address space name. If the `STOP` command fails to gracefully shut down the NFS Client, the operator could force an abnormal termination by issuing the `CANCEL` command.
Handling in-flight and indoubt transactions if there is a failure	Not applicable
How to determine if NSF is running	Run the nfsstat utility in directory `/usr/lpp/NFS`.
What happens to WebSphere for z/OS if NSF goes down?	New server starts will fail. An attempt to access an environment variable will fail.
What happens to other subsystems if NSF goes down?	Other subsystems should continue to work fine.

Table 13. NFS automation and recovery scenarios (continued)

Task	NFS automation and recovery scenarios
Where to find more information	• *z/OS Network File System User's Guide*, SC26-7419 • *z/OS Network File System Customization and Operation*, SC26-7417

WebSphere for z/OS (Daemon) automation and recovery scenarios

Table 14. WebSphere for z/OS automation and recovery scenarios

Task	WebSphere for z/OS (Daemon) automation and recovery scenarios
Startup	Refer to "Starting the WebSphere for z/OS environment" on page 12.
Shutdown	Refer to "Stopping the WebSphere for z/OS environment" on page 25.
Handling in-flight or indoubt transactions if there is a failure	The Daemon is a location agent. If the Daemon fails during the course of a transaction, locate requests to the Daemon will fail. These request failures will be surfaced by the client ORB. If the client is a WebSphere for z/OS client running in a sysplex, the locate request will be routed to another available Daemon in the sysplex, if present.
How to determine if the Daemon is running	Use the MVS display command described in "Determining if the location service daemon is up" on page 13.
What happens to WebSphere for z/OS if the Daemon goes down?	If the Daemon goes down, all WebSphere for z/OS servers started on the same system as the terminating Daemon will also be terminated.
What happens to other subsystems if the Daemon goes down?	Other subsystems will continue to work fine. As a general rule, if the Daemon goes down and there is another one in the sysplex, clients won't be affected.

WebServer automation and recovery scenarios

Table 15. WebServer (Servlet) automation and recovery scenarios

Task	WebServer automation and recovery scenarios
Startup	The WebSphere for z/OS product and infrastructure do not require a WebServer or WebSphere for z/OS. WebServers and the WebSphere for z/OS Standard Edition runtime have a relationship with WebSphere for z/OS only in the sense that a client application program that is written to use WebSphere for z/OS facilities may be written as a servlet. Any implications for ordering of startup will be introduced by the applications. You probably want to have the WebSphere for z/OS Object servers up and ready before starting the client application Web server hosting.
Shutdown	There are no dependencies from the product code. Similar to most applications, you may want to quiesce the clients prior to taking down the target WebSphere for z/OS servers. Shut down the WebServer to stop the port of entry.

Table 15. WebServer (Servlet) automation and recovery scenarios (continued)

Task	WebServer automation and recovery scenarios
Handling in-flight or indoubt transactions if there is a failure	Since a WebServer is stateless, there are no in-flight or indoubt transactions.
How to determine if WebServer is running	Use display commands and viewer tools (SDDF) to monitor the Application Server.
What happens to WebSphere for z/OS if WebServer goes down?	WebSphere for z/OS requires an IBM HTTP J2EE server WebServer in order to provide full function servlets if WebServer is not there. So, if the WebServer these applications that require a port of entry (like servlets and SOAP) can't run.
What happens to other subsystems if WebServer goes down?	HTTP only provides info to other subsystems, so they are uneffected if WebServer goes down.
Where to find more information	*z/OS HTTP Server Planning, Installing, and Using*, SC34-4826

Chapter 7. WebSphere for z/OS administration procedures

Reviewers Note:
No changes have been made to this chapter. I will need to review it with someone who understands this for V5.

This chapter describes WebSphere for z/OS administration tasks and guidelines.

For further information, please see:

- *WebSphere Application Server for z/OS V5.0: Installation and Customization*, GA22-7909 to:
  - Set up RACF system security and userids
  - Update DNS definitions
  - Set up your TCP/IP network
  - Update your hosts file as you expand your sysplex.
- See ″Application Programming″ section in the WebSphere Application Server for z/OS V5.0 InfoCenter, access to which can be obtained through the WebSphere forz/OS library Web site http://www.ibm.com/software/webservers/appserv/zos_os390/library.html. to:
  - Set up userids
  - Deploy applications in a Parallel sysplex Environment
  - Install WebSphere for z/OS applications (sysplex-wide).

Logging messages and trace data for Java server applications

By using the WebSphere for z/OS support for logging application messages and trace data, you can improve the reliability, availability, and serviceability of any Java application that runs in a WebSphere for z/OS server. Through this support, your Java application's messages can appear on the MVS master console, in the error log stream, or in the component trace (CTRACE) data set for WebSphere for z/OS. Your application's trace entries can appear in the same CTRACE data set.

Determining where to issue the messages

You might want to issue messages to the MVS master console to report serious error conditions for mission-critical applications. Through the master console, an operator can receive and, if necessary, take action in response to a message that indicates the status of an application. In addition, by directing messages to the master console, you can trigger automation packages to take action for specific conditions or events related to your application's processing.

Any messages that your application issues to the console also appear in either the error log stream or the CTRACE data set for WebSphere for z/OS, depending on the message type. Logging the messages in these system resources can help you more easily diagnose errors related to your application's processing. Similarly, issuing requests to log trace data in the CTRACE data set is another method of recording error conditions or collecting application data for diagnostic purposes.

System performance when logging messages and trace data

You can select the amount and types of trace data to be collected, which provides you with the ability to either run your application with minimal tracing when performance is a priority, or run your application with detailed tracing when you need to recreate a problem and collect additional diagnostic information.

The error log stream, the CTRACE data set for WebSphere for z/OS, and the master console are primarily intended for monitoring or recording diagnostic data for system components and critical applications. Depending on your installation's configuration, directing application messages and data to these resources might have an adverse affect on system performance. For example, if you send application data to the CTRACE data set, trace entries in that data set might wrap more quickly, which means you might lose some critical diagnostic data because the system writes new entries over existing ones when wrapping occurs. Use this logging support judiciously.

Note: You can use WebSphere for z/OS support for logging messages and trace data only for Java applications, not for Java applets. See See ″Application Programming″ section in the WebSphere Application Server for z/OS V5.0 InfoCenter, access to which can be obtained through the WebSphere forz/OS library Web site http://www.ibm.com/software/webservers/appserv/zos_os390/library.html for more information about logging messages and trace data for Java server applications.

Issuing application messages to the MVS master console

With the WebSphere for z/OS reliability, availability, and servicability support for Java (JRAS), you can issue messages from your Java application to the MVS master console. You might want to issue messages to the master console to report serious error conditions for mission-critical applications, or to trigger automation packages.

The messages your application issues also appear in either the error log stream or the component trace (CTRACE) data set that WebSphere for z/OS uses.

Logging the messages is another method of recording error conditions or collecting application data for diagnostic purposes.

Using a message logger

WebSphere for z/OS provides code that creates and manages a message logger, which processes your application's messages. WebSphere for z/OS creates only one message logger for each unique organization, product, or component, so that you can more easily identify the messages recorded in the error log stream or CTRACE data set for a specific application. The message logger runs in the Java Virtual Machine (JVM) for the WebSphere for z/OS server in which your Java application will run.

To use a message logger, all you need to do in your Java application is:

1. Define the message logger.
2. Drive the method to instruct WebSphere for z/OS to create the message logger.
3. Code messages at appropriate points in your application.

See:

- ″Application Programming″ section in the WebSphere Application Server for z/OS V5.0 InfoCenter, access to which can be obtained through the WebSphere forz/OS library Web site http://www.ibm.com/software/webservers/appserv/zos_os390/library.html, for more general information.

- *WebSphere Application Server for z/OS V5.0: Installation and Customization*, GA22-7909, for information on how to set up the error log stream.

Chapter 8. Security administration

This chapter describes general WebSphere for z/OS security procedures.

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 Security Server RACF Security Administrator's Guide*, SA22-7683
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"

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.

Chapter 9. WebSphere for z/OS performance tuning guidelines

This chapter describes general WebSphere for z/OS tuning guidelines for J2EE applications. For steps in diagnosing performance problems, refer to Chapter 10, “WebSphere for z/OS performance monitoring procedures”, on page 101.

Introduction

One of the goals of the WebSphere for z/OS programming model and runtime is to significantly simplify the work required for application developers to write and deploy applications. Sometimes we say that WebSphere for z/OS relieves the application programmer of many of the plumbing tasks involved in developing applications.

For example, application code in WebSphere for z/OS does not concern itself directly with remote communication–it locates objects which may be local or remote and drives methods. Therefore, you won’t see any direct use of socket calls or TCP/IP programming in a WebSphere for z/OS application.

This separation of what you want to do from where you do it is one aspect of removing the application programmers from plumbing tasks. Other considerations are not having to deal with data calls for some types of beans, potentially user authentication, and threading. There are generally no calls from the application code to touch sockets, RACF calls, or management of threading. Removing this from the application programmer doesn’t mean this work won’t get done. Rather, it means that there may be more work for the DBA, the network administrator, the security administrator, and the performance analyst.

This chapter will focus on the performance tuning aspects of WebSphere for z/OS. This becomes a complex exercise because the nature of the runtime involves many different components of the operating system and middleware.

Recommendation: Before you read a description of WebSphere for z/OS tuning guidelines, it is important to note that, no matter how well the middleware is tuned, it cannot make up for poorly designed and coded applications. Focusing on the application code can help improve performance. Often, poorly written or designed application code changes will make the most dramatic improvements to overall performance.

For more information on tuning specific applications, see the ″Application Programming″ section in the WebSphere Application Server for z/OS V5.0 InfoCenter, access to which can be obtained through the WebSphere for z/OS library Web site http://www.ibm.com/software/webservers/appserv/zos_os390/library.html

Recommended hardware configuration

The tuning guidelines presented in this chapter will have the most benefit on the following recommended configuration:

- IBM S/390 G5 Model 9672-Rx6, or later

  Note: IEEE floating point, which is commonly used in Java, is emulated on earlier machines.

- Storage

  Storage requirements are higher than for traditional workloads

  - Virtual storage default should be about 175 MB per servant, which includes a 64 MB default heap size and a default initial LE heap size of 80 MB.
  - Real storage minimum is 512 MB per LPAR for a light load such as the IVP. For most real-world applications, we recommend 2 GB or higher.
- DASD

  To maximize your performance, we recommend a fast DASD subsystem (for example, IBM Shark), running with a high cache read/write hit rate.
- Networking

  For high bandwidth applications, we recomend at least a 1 Gb Ethernet connection. If your applications have extremely high bandwidth requirements, you may need additional Ethernet connections.
- Cryptography

  For applications that make heavy use of cryptography, we recommend the zSeries or S/390 cryptographic hardware and the Integrated Cryptographic Service Facility.

Organization of tuning

There are four layers of tuning that need to be addressed:

- The operating system
- Subsystems
- WebSphere for z/OS
- Specific applications

We deal with the first three in separate sections of this chapter and briefly touch on the fourth. For more information on tuning applications, refer to the ″Application Programming″ section in the WebSphere Application Server for z/OS V5.0 InfoCenter, access to which can be obtained through the WebSphere forz/OS library Web site http://www.ibm.com/software/webservers/appserv/zos_os390/library.html.

Refer to Table 16 for a list of tuning tips and quick links to each tip.

Table 16. Links to tuning tips

System to be tuned	Comments	Cross Reference
Operating System		
Virtual Storage	Ensure adequate virtual storage	“Virtual storage” on page 72
Real Storage	A small configuration requires at least 512MB of real storage.	“Real storage” on page 72
Java virtual machine (JVM) heap size	Specifying a sufficient JVM heap size is important to Java performance.	“Garbage collection and `JVM Heap Size`” on page 73
Servant recycling	Changing servant recycling can improve the performance of applications with storage leaks.	“Servant recycling” on page 74

Table 16. Links to tuning tips (continued)

System to be tuned	Comments	Cross Reference
Operating System		
z/OS or OS/390	Tuning tips for various components of z/OS or OS/390	"Tuning tips for z/OS or OS/390" on page 74
Resource Recovery Services (RRS)		"Tuning tips for the Resource Recovery Services (RRS)" on page 75
Language Environment (LE)		"Tuning tips for LE" on page 77
LE HEAP		"Fine tuning the LE Heap" on page 78
z/OS UNIX System Services		"z/OS UNIX System Services" on page 79
HFS Performance		"HFS performance considerations" on page 80
Workload Manager (WLM)		"Tuning tips for WLM" on page 80
Subsystems		
DB2		"Tuning tips for DB2" on page 82
RACF		"Tuning tips for RACF" on page 84
TCP/IP		"Tuning tips for TCP/IP" on page 85
WebSphere MQ and JMS		"Tuning tips for Using External WebSphere MQ V5.3 as the Messaging Provider" on page 87
GRS		"Tuning tips for GRS" on page 91
JVM		"Tuning tips for JVM" on page 92
CICS		"Tuning tips for CICS" on page 92
WebSphere for z/OS		
WebSphere for z/OS runtime environment		"Tuning the WebSphere for z/OS runtime" on page 93
Server configuration		on page 0
Locating executable programs		
Security		"Security" on page 94
Servlet/EJB integrated runtime		"Servlet / EJB integrated runtime" on page 95
J2EE Applications		
Topology considerations	Describes some of the topology considerations you should be aware of.	"Topology" on page 96

Table 16. Links to tuning tips (continued)

System to be tuned	Comments	Cross Reference
Operating System		
J2EE containers and applications	Selection of EJBs and transaction policies affects performance.	"J2EE container and applications" on page 97
J2EE applications	General programming tips on JavaServer pages (JSPs) and Java Database Connectivity (JDBC)	"J2EE application programming tips" on page 99
SOAP tuning	The Simple Object Access Protocol provides a mechanism to exchanging information between peers in a decentralized, distributed environment.	

Tuning the operating system

Storage

Virtual storage

Ensure that you don't underestimate the amount of virtual storage applied to the WebSphere for z/OS servers. Generally, they use significantly more virtual memory than traditional application servers on z/OS or OS/390. The setting of REGION on the JCL for the proc should be large (at least 200MB to run), and much larger if high throughput is required. You can get an idea of the virtual storage usage through RMF or other performance monitors. It would not be unreasonable for the servant procs to specify `REGION=0M`, which tells the operating system to give all the available region (close to 2GB).

Note: For more information on `REGION=0M` and IEFUSI, please see "Chapter 2. Preparing the base OS/390 or z/OS environment" (specifically, the section "Recommendations for using memory") in *WebSphere Application Server for z/OS V5.0: Installation and Customization*, GA22-7909.

If you choose to not put most of the runtime in LPA, as described in the program locations section, be sure to specify more region (as high as 512MB). Also, in conjunction with the increase in storage usage, you may have to define more paging space or auxiliary storage to back up the additional virtual storage used.

Real storage

From reading "Virtual storage" on page 72, you can see that the servers use a lot of virtual storage. Since real storage is needed to back the virtual storage, its usage is also high. Expect a requirement of at least 512MB of real storage for a small configuration. For controllers and servants, real storage utilizations depends on the size of the JVM heapsize.

Recommendation: As we have said, WebSphere for z/OS puts much higher demands on virtual memory than a traditional workload. It can be the case that in a heavy use environment 2G of central storage is not enough to handle the real storage demands of a high volume Java application. In this case, we recommend that you configure with 64-bit real storage, which will give you the ability to dedicate more central storage to the LPAR. Installations with a zSeries processor (z800, z900) have the ability to run OS/390 Rel. 10 in 64-bit mode. z/OS on a zSeries will always run in 64-bit mode. Running in 64-bit mode gives you the ability to define more than 2 GB of central storage When you configure for 64-bit real all of the storage is defined as central storage.For non-zSeries processors or 31-bit mode, you can minimize paging by defining more expanded storage.

Garbage collection and `JVM Heap Size`

Specifying a sufficient `JVM Heap Size` is important to Java performance. The JVM has thresholds it uses to manage the JVM's storage. When the thresholds are reached, the garbage collector (GC) gets invoked to free up unused storage. GC can cause significant degradation of Java performance.

How to view or set: Use the WebSphere adminstrative console:

1. Click **Servers** > **Application Servers** > **`server_name`** > **Process Definition** > **Java Virtual Machine**
2. On the Configuration Tab you can specify the Initial Heap Size and the Maximum Heap Size.

Tips: Use the following guidelines to specify JVM heap size:

- In order to get it to run less frequently, you can give the JVM more memory. This is done by specifying a larger value for `Initial Heap Size`. The default of 64M is a good starting point but may need to be raised for larger applications. When specifying either a larger or smaller JVM heap size value, IBM recommends that you code **both** the initial and maximum values that you desire.
- It is good for the Initial Heap Size to equal the Maximum Heap Size because it allows the allocated storage to be completely filled before GC kicks in. Otherwise, GC will run more frequently than necessary, potentially impacting performance.
- Make sure the region is large enough to hold the specified JVM heap.
- Beware of making the `Initial Heap Size` *too* large. While it initially improves performance by delaying garbage collection, it ultimately affects response time when garbage collection eventually kicks in (because it runs for a longer time).
- Paging activity on your system must also be considered when you set your JVM heap size. If your system is already paging heavily, increasing the JVM heapsize might make performance worse rather than better.
- To determine if you are being affected by garbage collection, you can enable Verbose Garbage Collection on the JVM Configuration tab. The default is unenabled. This will write a report to the output stream each time the garbage collector runs. This is not a very human-friendly report, but you will get an idea of what is going on with Java GC.

  Example: This is an example of a verboseGC report.

  ```
  ...
  <AF[21]: Allocation Failure. need 32784 bytes, 32225 ms since last AF>
  <AF[21]: managing allocation failure, action=1 (84320/131004928)
                                                  (3145728/3145728)>
  <GC(21): GC cycle started Wed Feb 27 22:46:11 2002
  <GC(21): freed 99587928 bytes, 76% free (102817976/134150656), in 118 ms>
  <GC(21): mark: 103 ms, sweep: 15 ms, compact: 0 ms>
  <GC(21): refs: soft 0 (age >= 32), weak 0, final 878, phantom 0>
  <AF[21]: completed in 118 ms>
  ...
  ```

  Key things to look for in a verboseGC report are:

  – Time spent in garbage collection.

Ideally, you want to be spending less than 5% of the time in GC. To determine percentage of time spent in GC, divide the time it took to complete the collection by the time since the last AF and multiply the result by 100. In the example on page 73,

```
118/32225 * 100 = 0.366%
```

If you are spending more than 2% of your time in GC and if GC is occurring frequently, you may need to increase your Java heap size.

– Growth in the allocated heap.

  To determine this, look at the %free. You want to make sure the number is not continuing to decline. If the %free continues to decline you are experiencing a gradual growth in allocated heap from GC to GC which could indicate that your application has a memory leak.

You can also use the MVS console command, *modify display, jvmheap* to display JVM heap information. See "Modify operations" on page 29 for details. In addition, you can check the server activity and interval SMF records. See "SMF record type 120 (WebSphere for z/OS)" on page 121 for details.

Servant recycling

The default specification for servant recycling is 0, meaning that it is turned off.

Note: If your application exploits one or more of the following, then you must leave the servant recycle function off , or set to 0 (zero):

- HTTP Session Affinity (i.e., Session-in-memory using cookies over the HTTP internal transport).
- Stateful Session EJBs that are configured, via WebSphere for z/OS extended deployment descriptors, to store data in memory.
- Stateful Session EJBs that are configured to ″activate once passivate never.″

Unless you have a specific need to turn recycling on, we recommend that you leave it off.

To activate servant recycling, set this value to a number higher than 0. For example, if you set the value to 50,000, after 50,000 transactions the servant will no longer pick up any new work. A new servant is started to pick up the new work while the old servant finishes up the work it already has. When it finishes, it terminates, and the process continues with the new servant.

Tuning tips for z/OS or OS/390

This section provides tuning tips for various components of z/OS or OS/390.

CTRACE Tips:

- The first place to review is your CTRACE configuration. Ensure that all components are either set to `MIN` or `OFF`. To display the CTRACE options for all components on your system, issue the following command from the operator console:

  ```
  D TRACE,COMP=ALL
  ```

  To change the setting for an individual component to its minimum tracing value, use the following command (where 'xxx' is the component):

  ```
  TRACE CT,OFF,COMP=xxxx
  ```

This will eliminate any unnecessary overhead of collecting trace information that is not being used. Often during debug, CTRACE is turned on for a component and not shut off when the problem is resolved.

SMF Tips:

- Ensure that you are not collecting more SMF data than you need. Review the SMFPRMxx to ensure that only the minimum number of records are being collected.

  Use SMF 92 or 120 only for diagnostics.

  – SMF Type 92

    SMF Type 92 records are created each time an HFS file is opened, closed, deleted, and so forth. Almost every web server request references HFS files, so thousands of SMF Type 92 records are created. Unless you specifically need this information, turn off SMF Type 92 records. In the following example, we have disabled the collection of SMF type 92 records:

    Example:

    ```
    ACTIVE,
    DSNAME(SYS1.&.SYSNAME..SMF.MAN1;SYS1.&SYSNAME..SMF.MAN2;),
    NOPROMPT,
    REC(PERM),
    MAXDORM(3000),
    STATUS(010000),
    JWT(0510),
    SID(&SYSNAME;(1:4)),
    LISTDSN,
    SYS(NOTYPE(19,40,92)),
    INTVAL(30),
    SYNCVAL(00),
    SYS(DETAIL,INTERVAL(SMF,SYNC)),
    SYS(EXITS(IEFACTRT,IEFUJI,IEFU29,IEFU83,IEFU84,IEFU85,IEFUJV,IEFUSI))
    ```

  – SMF Type 120

    You may find that running with SMF 120 records in production is appropriate, since these records give information specific to WebSphere applications such as response time for J2EE artifiacts, bytes transferred, and so forth. If you do choose to run with SMF 120 records enabled, we recommend that you use server interval SMF records and container interval SMF records rather than server activity records and container activity records. For details of the SMF 120 record, refer to "SMF record type 120 (WebSphere for z/OS)" on page 121.

    Controlling collection of SMF 120 records involves two steps.

    1. Use the Administration application to tell WebSphere for z/OS to collect SMF records and the types of records (server interval SMF records and container interval SMF records).
    2. Use the SMFPRMxx to tell z/OS to collect type 120 records.

You might also want to review your DB2 records and the standard RMF written SMF records, and ensure that the SMF datasets are allocated optimally. Information on ensuring a high performance SMF recording environment can be found in the chapter on Customizing SMF in *z/OS MVS System Management Facilities (SMF)*, SA22-7630.

Tuning tips for the Resource Recovery Services (RRS)

- Throughput will benefit from moving the RRS logs in logger to a coupling facility (CF) logstream. Doing so will help transactions complete quickly and not require any DASD I/O.

- The number of log records per transaction depends on your operating system level. Before OS/390 R10 (which had DB2 6.1), you could write as many as four log records to a logger in a single operation. The archive log was not required, so you could get up to three records per transaction. DB2 7.1 allowed you get up to two log writes per transaction with a PTF, OS/390 R10 with DB2 7.1 PTF allowed you to get up to one log record per transaction, and OS/390 R10 with RRS APAR OW50074 can get to zero log records per transaction in the logger.
- Ensure that your logger configuration is optimal by using SMF 88 records. Seeing the tuning section of *z/OS MVS Setting Up a Sysplex*, SA22-7625 or the chapter on System Logger Accounting in *z/OS MVS System Management Facilities (SMF)*, SA22-7630for details.
- In any case, you should monitor the logger to ensure that there is a sufficient size in the CF and that offloading is not impacting the overall throughput. The transaction logs are one of the only shared I/O intensive resources in the mainline and can affect throughput dramatically if they are mistuned.
- DASD logger can limit your throughput because it is I/O-sensitive. The CF logger has much more throughput (in one measurement, the CF logger was six times faster than the DASD logger). If it's not possible to use CF logs, use well performing DASD and make sure the logs are allocated with large CI sizes.
- The default values of LOGR policy may have an impact on performance. We recommend the default settings in Table 17.

Table 17. Recommended default setting for LOGR

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

- If you are using XA Resource Managers and you have chosen to put the logs in the logger, you may have to review the log sizes. As of this writing, we cannot give specific recommendations.
- If you don't need the archive log, we recommend that you eliminate it since it can introduce extra DASD I/Os. The archive log contains the results of completed transactions. Normally, the archive log is not needed. Following is an example of disabling archive logging.

  Example:

```
//STEP1 EXEC PGM=IXCMIAPU
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
    DATA TYPE(LOGR)
    DELETE LOGSTREAM NAME(ATR.WITPLEX.ARCHIVE)
    DELETE LOGSTREAM NAME(ATR.WITPLEX.MAIN.UR)
    DELETE LOGSTREAM NAME(ATR.WITPLEX.RESTART)
    DELETE LOGSTREAM NAME(ATR.WITPLEX.RM.DATA)
    DELETE LOGSTREAM NAME(ATR.WITPLEX.DELAYED.UR)
    DELETE STRUCTURE NAME(RRSSTRUCT1)
/*
//STEP2 EXEC PGM=IXCMIAPU
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
    DATA TYPE(LOGR)
    DEFINE STRUCTURE NAME(RRSSTRUCT1)
           LOGSNUM(9)
    DEFINE LOGSTREAM NAME(ATR.WITPLEX.MAIN.UR)
           STRUCTNAME(RRSSTRUCT1)
           STG_DUPLEX(YES)
           DUPLEXMODE(UNCOND)
           LS_DATACLAS(SYSPLEX)
           LS_STORCLAS(LOGGER)
           HLQ(IXGLOGR)
           AUTODELETE(YES)
           RETPD(3)
    DEFINE LOGSTREAM NAME(ATR.WITPLEX.RESTART)
           STRUCTNAME(RRSSTRUCT1)
           STG_DUPLEX(YES)
           DUPLEXMODE(UNCOND)
           LS_DATACLAS(SYSPLEX)
           LS_STORCLAS(LOGGER)
           HLQ(IXGLOGR)
           AUTODELETE(YES)
           RETPD(3)
    DEFINE LOGSTREAM NAME(ATR.WITPLEX.RM.DATA)
           STRUCTNAME(RRSSTRUCT1)
           STG_DUPLEX(YES)
           DUPLEXMODE(UNCOND)
           LS_DATACLAS(SYSPLEX)
           LS_STORCLAS(LOGGER)
           HLQ(IXGLOGR)
           AUTODELETE(YES)
           RETPD(3)
    DEFINE LOGSTREAM NAME(ATR.WITPLEX.DELAYED.UR)
           STRUCTNAME(RRSSTRUCT1)
           STG_DUPLEX(YES)
           DUPLEXMODE(UNCOND)
           LS_DATACLAS(SYSPLEX)
           LS_STORCLAS(LOGGER)
           HLQ(IXGLOGR)
           AUTODELETE(YES)
           RETPD(3)
/*

//* DEFINE LOGSTREAM NAME(ATR.WITPLEX.ARCHIVE)
           //* STRUCTNAME(RRSSTRUCT1)
           //* STG_DUPLEX(YES)
           //* DUPLEXMODE(UNCOND)
           //* LS_DATACLAS(SYSPLEX)
           //* LS_STORCLAS(LOGGER)
           //* HLQ(IXGLOGR)
           //* AUTODELETE(YES)
           //* RETPD(3)
```

Tuning tips for LE

- For best performance, use the LPALSTxx parmlib member to ensure that LE and C++ runtimes are loaded into LPA, as shown in the following example:

  Example: sys1.parmlib(LPALSTxx):

```
********************************* Top of Data ********************
USER.LPALIB,
ISF.SISFLPA,                            SDSF
CEE.SCEELPA,                            LANGUAGE ENVIRONMENT
CBC.SCLBDLL,                            C++ RUNTIME

.
.
.
******************************** Bottom of Data ******************
```

- Options you should **NOT** use during production:
  - `RPTSTG(ON)`
  - `RPTOPTS(ON)`
  - `HEAPCHK(ON)`
- If you are running a client on z/OS, setting the following: `SET LEPARM='HEAPP(ON)'` in a shell script, turns on LE heappools, which should improve the performance of the client.
- Fine-tuning LE Heap options

  Refer to “Fine tuning the LE Heap” for details.

Fine tuning the LE Heap

The LE Heap is an area of storage management to be concerned with. For servers, IBM has compiled default values for `HEAP` and `HEAPPOOL` into the server main programs. These are good starting points for simple applications.**Tips:** To fine tune the LE Heap settings, use the following procedure:

1. Use the LE function `RPTSTG(ON)` on the `PARM=` in JCL to get a report on storage utilization for your application servers.

   ```
   SET LEPARM='RPTOPTS(ON),RPTSTG(ON)'
   ```

   Results appear in servant joblog. To bring the server down cleanly, use the following command:

   ```
   VARY WLM,APPLENV=xxxx,QUIESCE
   ```

   The following example shows the servant SYSPRINT DD output from the `RPTSTG(ON)` option.

   Example:

   ```
   .   .   .
   0    HEAP statistics:
          Initial size:                                   83886080
          Increment size:                                  5242880
          Total heap storage used (sugg. initial size):  184809328
          Successful Get Heap requests:                     426551
          Successful Free Heap requests:                    424262
          Number of segments allocated:                          1
          Number of segments freed:                              0
      .   .   .

     Suggested Percentages for current Cell Sizes:
       HEAPP(ON,8,6,16,4,80,42,808,45,960,5,2048,20)
     Suggested Cell Sizes:
       HEAPP(ON,32,,80,,192,,520,,1232,,2048,)
    . . .
   ```

2. Take the heap values from the ″Suggested Cell Sizes″ line in the example in 1 and use them in `RPTSTG(ON)` function to get another report on storage utilization.

   ```
   SET LEPARM='RPTOPTS(ON),RPTSTG(ON,32,,80,,192,,520,,1232,,2048,)'
   ```

   The following example in shows the servant joblog output from the `RPTOPTS(ON),RPTSTG(ON,32,,80,,192,,520,,1232,,2048,)` option.

   Example:

   ```
   .   .   .
   0    HEAP statistics:
          Initial size:                                   83886080
          Increment size:                                  5242880
          Total heap storage used (sugg. initial size):  195803218
   ```

```
       Successful Get Heap requests:                       426551
       Successful Free Heap requests:                      424262
       Number of segments allocated:                            1
       Number of segments freed:                                0
   .   .   .

  Suggested Percentages for current Cell Sizes:
    HEAPP(ON,32,8,80,43,192,48,520,20,1232,5,2048,20)
  Suggested Cell Sizes:
    HEAPP(ON,32,,80,,192,,520,,1232,,2048,)
 . . .
```

3. From the second storage utilization report, shown in 2 on page 78, take the heap values from the ″Suggested Percentages for current Cell Sizes″ line use them in RPTSTG(ON) function to get a third report on storage utilization.

   ```
   SET LEPARM='RPTOPTS(ON),RPTSTG(ON,32,8,80,43,192,48,520,20,1232,5,2048,20)'
   ```

   The following example shows the servant joblog output from the RPTOPTS(ON),RPTSTG(ON,32,8,80,43,192,48,520,20,1232,5,2048,20) option.

   Example:

   ```
   .   .   .
   0    HEAP statistics:
          Initial size:                                     83886080
          Increment size:                                    5242880
          Total heap storage used (sugg. initial size):    198372130
          Successful Get Heap requests:                       426551
          Successful Free Heap requests:                      424262
          Number of segments allocated:                            1
          Number of segments freed:                                0
      .   .   .

     Suggested Percentages for current Cell Sizes:
       HEAPP(ON,32,8,80,43,192,48,520,20,1232,5,2048,20)
     Suggested Cell Sizes:
       HEAPP(ON,32,,80,,192,,520,,1232,,2048,)
    . . .
   ```

4. On the third storage utilization report, look for the ″Total heap storage used (sugg. initial size):″ line and use this value for your initial LE heap setting. For example, in the report in 3 this value is 198372130.
5. Make sure that you remove RPTSTG since it does incur a small performance penalty to collect the storage use information.
6. For your client programs that run on z/OS or OS/390, we recommend that you at least specify HEAPP(ON) on the proc of your client to get the default LE heappools.

For more information, please see “Tuning tips for LE” on page 77.

Note: Ensure that if you use LE HEAPCHECK, you turn it off once you have ensured that your code doesn’t include any uninitialized storage. HEAPCHECK can be very expensive.

z/OS UNIX System Services

WebSphere Application Server for z/OS V5 no longer requires or recommends shared HFS for the configuration files. WebSphere for z/OS maintains its own mechanism for managing this data in a cluster. If you chose to use shared HFS for your application, the following section may give you some information.

For basic z/OS UNIX System Services performance information, refer to the following web site:

```
http://www.ibm.com/servers/eserver/zseries/ebusiness/perform.html
```

HFS performance considerations

- HFS R/O mount

  Special consideration needs to be made to file system access when you run in a sysplex. If you mount the file system R/W in a shared HFS environment, only one system will have local access to the files. All other systems have remote access to the files which negatively affects performance. You may choose to put all of the files for WebSphere in their own mountable file system and mount it R/O to improve performance. However, to change your current application or install new applications, the file system must be mounted R/W. You will need to put operational procedures in place to ensure that the file system is mounted R/W when updating or installing applications.
- HFS files caching
  - HFS Files Caching Read/Write files are cached in kernel dataspaces. In order to determine what files would be good candidates for file caching you can use SMF 92 records.
  - Initial cache size is defined in BPXPRMxx
- zFS
  - New UNIX file system for z/OS
  - You may benefit from using the zFS for your Unix file system. See *z/OS UNIX System Services Planning*, GA22-7800 for more information.
- Using the filecache command
  - High activity, read-only files can be cached in the USS kernel using the filecache command.
  - Access to files in the filecache can be much more efficient than access to files in the HFS, even if the HFS files are cached in dataspaces.
  - GRS latch contention, which sometimes is an issue for frequently accessed HFS files, will not affect files in the filecache.
  - To filecache important files at startup, you can add filecache command to your /etc/rc file.
  - Unfortunately, files which are modified after being added to the filecache may not be eligible for caching until the file system is unmounted and remounted, or until the system is re-IPLed.
  - Refer to *z/OS UNIX System Services Command Reference*, SA22-7802 for more information about the filecache command.
  - Example of using filecache command:

    ```
    /usr/sbin/filecache -a /usr/lpp/WebSphere/V5R0M0/MQSeries/java/samples/base/de_DE/mqsample.html
    ```

- - - - - - . ---

Tuning tips for WLM

If you are running z/OS 1.2 or higher, you may choose to use the dynamic application environment. In that case, use the Administrative console to turn on the dynamic application environment. Use the Administrative console to provide the JCL proc name for the servant and the JCL Parm for the servant.

Note: Please see the ″Servers″ section in the WebSphere Application Server for z/OS V5.0 InfoCenter, access to which can be obtained through the WebSphere for z/OS library Web site http://www.ibm.com/software/webservers/appserv/zos_os390/library.html for more information on ″Defining application server processes.″

Whether or not you use the dynamic application environment, you will need to follow the instructions below to set the WLM goals.

Setting the WLM goals properly can have a very significant effect on application throughput. The WebSphere for z/OS system address spaces should be given a fairly high priority. As work comes into the system, the work classification of the enclaves should be based on your business goals.

Tips:

- Location service daemons and controllers
  - Classify as SYSSTC or high Velocity
- Application Servers
  - Classify with velocity goal using STC classification rules.

    -- Java garbage collection runs under this classification. Java GC is a CPU and storage intensive process, so if you set the velocity goal too high GC could consume more of the system resources than desired. On the other hand, if your Java heap is correctly tuned, GC for each server region should run no more than 5% of the time. Also, providing proper priority to GC processing is necessary since other work in the server region is stopped during much of the time GC is running. --

    JSP compiles run under this classification. If your system is configured to do these compiles at run-time, setting the velocity goal too low could result in longer delays waiting for JSP compiles to complete.

    Application work is actually classified under the work manager.
- Application Environment for work running under servants
  - Subsystem type = CB
  - Classify based on Application Server name and Userid
  - Percentage response time goal is recommended

    It is usually a good idea to make the goals achievable. For example, a goal that 80% of the work will complete in .25 seconds is a typical goal. Velocity goals for application work are not meaningful and should be avoided.
  - Provide a high velocity default service class for CB transactions (Default is SYSOTHER)
- Set your Application environment to ″No Limit″
  - Required if you need more than one servant per system.
  - Under WLM, you can control how many servants can be started for each server. If you need more than one servant in a server make sure that ″No Limit″ is selected for the application environment associated with your server. For information about setting up WLM performance goals, see *z/OS MVS Planning: Workload Management*, SA22-7602.

    Example:

    ```
    Application-Environment  Notes  Options  Help
    --------------------------------------------------------------------------
                        Modify an Application Environment
    Command ===> ______________________________________________________________

    Application Environment Name . : BBOASR2
    Description  . . . . . . . . . . WAS.V40.WB02 Application server
    Subsystem Type . . . . . . . . . CB
    Procedure Name . . . . . . . . . BBOASR2S
    ```

```
Start Parameters . . . . . . . . IWMSSNM=&IWMSSNM
                                 ________________________________________
                                 ________________________________________

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

Note: When the WLM configuration is set to no limit, you can control the maximum and minimum number of server regions by specifying the environment variables `MAX_SRS=x` and `MIN_SRS=y` in the server's current.env file. **Caution:** If you specify MAX_SRS you must ensure that you specify a MAX_SRS value that is greater than or equal to MIN_SRS times the number of service classes you have defined for this application environment. Failure to do so can result in timeouts due to an insufficient number of server regions.

- Results reported in RMF monitor I workload activity report:
  - Transactions per second (not always the same as client tran rate)
  - Average response times (and distrubution of response times)
  - CPU time used

Tuning for subsystems

Tuning tips for DB2

Performance tuning for DB2 is usually critical to the overall performance of a WebSphere for z/OS application. DB2 is often the preferred datastore for a session or EJB. There are many books that cover DB2 tuning–we can't possibly provide as thorough a treatment of DB2 here as we would like. Listed here are some basic guidelines for DB2 tuning as well as some guidelines for tuning DB2 for WebSphere. For more complete information on DB2 tuning refer to DB2 Universal Database for OS/390 and z/OS Administration Guide Version 7 Document Number SC26-9931-02. <Need to add citation here> For information on application programming for JDBC, refer to "JDBC topic" on page 99.

General DB2 tuning

Tips: Following are some general DB2 tuning tips:

- First, ensure that your DB2 logs are large enough and are allocated on the fastest volumes you have.
- Next ensure that you have tuned your bufferpools so that the most often-read data is in memory as much as possible. Use ESTOR and hyperpools.
- You many want to consider pre-formatting tables that are going to be heavily used. This avoids formatting at runtime.
- Ensure that you have optimal CI sizes for the DB2 logs.

DB2 tuning for WebSphere

Tips: Follow are DB2 tuning tips for the WebSphere environment:

- Turn off any JDBC tracing in db2sqljjdbc.properties.

  ```
  DB2SQLJSSID=DB2
  DB2SQLJATTACHTYPE=RRSAF
  DB2SQLJMULTICONTEXT=YES
  DB2SQLJDBRMLIB=MVSDSOM.DB2710.DBRMLIB.DATA
  #DB2SQLJ_TRACE_FILENAME=/tmp/mytrc
  ```

- We recommend that you ensure indexes defined on all your object primary keys. Failure to do so will result in costly tablespace scans.

- Ensure that, once your tables are sufficiently populated, you do a re-org to compact the tables. Executing RUNSTATS will ensure that the DB2 catalog statistics about table and column sizes and accesses are most current so that the best access patterns are chosen by the optimizer.
- You will have to define more connections called threads in DB2. WebSphere for z/OS uses a lot of threads. Sometimes this is the source of throughput bottlenecks since the server will wait at the create thread until one is available.
- Make sure you are current with JDBC maintenance. Many performance improvements have been made to JDBC recently. To determine the JDBC maintenance level, enter the following from the shell:

  ```
  java COM.ibm.db2os390.sqlj.util.DB2DriverInfo
  ```

  If this returns a class not found, either you are at a level of the driver that is older and doesn't support this command or you have not issued the command properly.
- We recommend that you enable dynamic statement caching in DB2. To do this, modify your ZPARMS to say `CACHEDYN(YES) MAXKEEPD(16K)`. Depending on the application, this can make a very significant improvement in DB2 performance. Specifically, it can help JDBC and LDAP query.
- Increase DB2 checkpoint interval settings to a large value. To do this, modify your ZPARMS to include `CHKFREQ=xxxxx`, where xxxxx is set at a high value when doing benchmarks. On production systems there are other valid reasons to keep checkpoint frequencies lower, however.

Example: This example identifies zparm values discussed in this section.

```
//DB2INSTE   JOB MSGCLASS=H,CLASS=A,NOTIFY=IBMUSER
/*JOBPARM SYSAFF=*
//******************************************************************
//* JOB NAME = DSNTIJUZ
//*
//* DESCRIPTIVE NAME = INSTALLATION JOB STREAM
//*
//*    LICENSED MATERIALS - PROPERTY OF IBM
//*    5675-DB2
//*    (C) COPYRIGHT 1982, 2000 IBM CORP.  ALL RIGHTS RESERVED.
//*
//*    STATUS = VERSION 7
//*
//* FUNCTION = DSNZPARM AND DSNHDECP UPDATES
//*
//* PSEUDOCODE =
//*   DSNTIZA  STEP  ASSEMBLE DSN6.... MACROS, CREATE DSNZPARM
//*   DSNTIZL  STEP  LINK EDIT DSNZPARM
//*   DSNTLOG  STEP  UPDATE PASSWORDS
//*   DSNTIZP  STEP  ASSEMBLE DSNHDECP DATA-ONLY LOAD MODULE
//*   DSNTIZQ  STEP  LINK EDIT DSNHDECP LOAD MODULE
//*   DSNTIMQ  STEP  SMP/E PROCESSING FOR DSNHDECP
//*
//* NOTES = STEP DSNTIMQ MUST BE CUSTOMIZED FOR SMP.  SEE THE NOTES
//*         NOTES PRECEDING STEP DSNTIMQ BEFORE RUNNING THIS JOB.
//*
//*  LOGLOAD=16000000,
//*********************************************************************/
//*
//DSNTIZA EXEC PGM=ASMA90,PARM='OBJECT,NODECK'
//STEPLIB DD DSN=ASM.SASMMOD1,DISP=SHR
//SYSLIB   DD  DISP=SHR,
//         DSN=DB2710.SDSNMACS
//         DD  DISP=SHR,
//         DSN=SYS1.MACLIB
```

```
//SYSLIN   DD  DSN=&LOADSET(DSNTILMP),DISP=(NEW,PASS),
//             UNIT=SYSALLDA,
//             SPACE=(800,(50,50,2)),DCB=(BLKSIZE=800)
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  UNIT=SYSALLDA,SPACE=(800,(50,50),,,ROUND)
//SYSUT2   DD  UNIT=SYSALLDA,SPACE=(800,(50,50),,,ROUND)
//SYSUT3   DD  UNIT=SYSALLDA,SPACE=(800,(50,50),,,ROUND)
//SYSIN    DD  *
    DSN6ENV    MVS=XA
    DSN6SPRM   RESTART,                 X
               .
               .
               .
               AUTH=YES,                                                X
               AUTHCACH=1024,                                           X
               BINDNV=BINDADD,                                          X
               BMPTOUT=4,                                               X
               CACHEDYN=YES,                                            X
               .
               .
               .
               MAXKEEPD=16000,                                          X
               .
               .
               .
   DSN6ARVP    ALCUNIT=CYL,                                             X
               .
               .
               .
   DSN6LOGP    DEALLCT=(0),                                             X
               .
               .
               .
   DSN6SYSP    AUDITST=NO,                                              X
               BACKODUR=5,                                              X
              CHKFREQ=16000000,                                        X
               CONDBAT=400,                                             X
               CTHREAD=1200,                                            X
               DBPROTCL=PRIVATE,                                        X
               DLDFREQ=5,                                               X
               DSSTIME=5,                                               X
               EXTRAREQ=100,                                            X
               EXTRASRV=100,                                            X
               EXTSEC=NO,                                               X
               IDBACK=1800,                                              X
               .
               .
               .
             //*
```

Tuning tips for RACF

Tips: Follow these guidelines for RACF tuning:

- As is always the case, don’t turn things on unless you need them. In general, the cost of security has been highly optimized. However, if you don’t need EJBROLEs, then don’t enable the class in RACF.
- You should ensure that you place into memory, by the RACLIST command, those items that will improve performance. Specifically, ensure that you RACLIST (if used):
  - ACEE
  - GTS
  - UID/GID

– CBIND

– EJBROLE

Example:

```
RACLIST (ACEE, GTS, UID/GID, CBIND, EJBROLE)
```

- Use of things like SSL come at a price. If you are a heavy SSL user, ensure that you have appropriate hardware, such as PCI crypto cards, to speed up the handshake process.
- Here's how you define the BPX.SAFFASTPATH facility class profile. This profile allows you to bypass SAF calls which can be used to audit successful HFS accesses.
  – Define the facility class profile to RACF.

    ```
    RDEFINE FACILITY BPX.SAFFASTPATH UACC(NONE)
    ```

  – Activate this change by doing one of the following:
    - re-IPL
    - invoke the SETOMVS or SET OMVS operator commands.

  Note: Do not use this option if you need to audit successful HFS accesses or if you use the IRRSXT00 exit to control HFS access.
- Use VLF caching of the UIDs and GIDs as shown in the example COFVLFxx parmlib member below:

  Example:sys1.parmlib(COFVLFxx):

```
********************************* Top of Data ********************
.
.
.
CLASS NAME(IRRGMAP) EMAJ(GMAP)
CLASS NAME(IRRUMAP) EMAJ(UMAP)
CLASS NAME(IRRGTS) EMAJ(GTS)
CLASS NAME(IRRACEE) EMAJ(ACEE)
.
.
.
******************************** Bottom of Data ******************
```

To avoid a costly scan of the RACF databases, make sure all HFS files have valid GIDs and UIDs.

Tuning tips for TCP/IP

TCP/IP can be the source of some significant remote method delays.**Tips:** Follow these tips to tune TCP/IP:

1. First, ensure that you have defined enough sockets to your system and that the default socket time-out of 180 seconds is not too high. To allow enough sockets, update the BPXPRMxx parmlib member:
   - Set MAXSOCKETS for the AF_INET filesystem high enough.
   - Set MAXFILEPROC high enough.

     We recommend setting MAXSOCKETS and MAXFILEPROC to at least 5000 for low-throughput, 10000 for medium-throughput, and 35000 for high-throughput WebSphere transaction environments. Setting high values for these parameters should not cause excessive use of resources unless the sockets or files are actually alllocated.

   Example:

```
/* Open/MVS Parmlib Member                                            */
/* CHANGE HISTORY:                                                    */
/*   01/31/02 AEK Increased MAXSOCKETS on AF_UNIX from 10000 to 50000*/
/*                per request from Michael Everett                    */
/*   10/02/01 JAB Set up shared HFS                                   */

/* KERNEL RESOURCES              DEFAULT             MIN MAX         */
/* ========================      =================== === =========== */
   .
   .
   .
   MAXFILEPROC(65535)            /* 64               3   65535       */
   .
   .
   .
    NETWORK DOMAINNAME(AF_INET) DOMAINNUMBER(2) MAXSOCKETS(30000)
   .
   .
   .
```

2. Next check the specification of the port in TCPIP profile dataset to ensure that NODELAYACKS is specified as follows:

   ```
   PORT 8082 TCP NODELAYACKS
   ```

   In your runs, changing this could improve throughput by as much as 50% (this is particularly useful when dealing with trivial workloads). This setting is important for good performance when running SSL.
3. You should ensure that your DNS configuration is optimized so that lookups for frequently-used servers and clients are being cached. Caching is sometimes related to the name server's Time To Live (TTL) value. On the one hand, setting the TTL high will ensure good cache hits. However, setting it high also means that, if the Daemon goes down, it will take a while for everyone in the network to be aware of it.
4. Increase the size of the TCPIP send and receive buffers from the default of 16K to at least 64K. This is the size of the buffers including control information beyond what is present in the data that you are sending in your application.To do this specify the following:

   ```
   TCPCONFIG TCPSENDBFRSIZE 65535
             TCPRCVBUFRSIZE 65535
   ```

   Note: It would not be unreasonable, in some cases, to specify 256K buffers.
5. Increase the default listen backlog. This is used to buffer spikes in new connections which come with a protocol like HTTP. The default listen backlog is 10 requests. We recommend that you increase this value to something larger.

   ```
   SOMAXCONN 100
   ```
6. Reduce the finwait2 time. In the most demanding benchmarks you may find that even defining 65K sockets and file descriptors does not give you enough 'free' sockets to run 100%. When a socket is closed (for example, no longer needed) it is not made available immediately. Instead it is placed into a state called finwait2 (this is what shows up in the netstat -s command). It waits there for a period of time before it is made available in the free pool. The default for this is 600 seconds.

   Note: Unless you have trouble using up sockets, we recommend that you leave this set to the default value.

   If you are using z/OS V1.2 or above, you can control the amount of time the socket stays in finwait2 state by specifying the following in the configuration file:

```
FINWAIT2TIME 60
```

Tuning tips for Using External WebSphere MQ V5.3 as the Messaging Provider

There are a number of WebSphere MQ parameters that you will probably need to modify for your specific environment. This section contains only a brief summary of WebSphere MQ Tuning concepts. For a more in-depth discussion, please consult http://www-3.ibm.com/software/ts/mqseries/txppacs/txpm4.html and download the MP16 Support Pac, Capacity planning and tuning for MQSeries for OS/390. You will also find Performance Reports for all versions of WebSphere MQ, MQSeries and WebSphere MQ Integrator (WMQI) at this web site.

Numerous tools are also available as Support Pacs. Another Support Pac of interest is MP1B, MQSeries for OS/390 V5.2 - Interpreting accounting and statistics data. You may need to collect SMF Accounting and Statistics data to analyze your MQSeries performance problems. If you wish to participate in MQSeries discussion groups, you can check the following web site http://www.mqseries.net/phpBB2/.

WebSphere MQ Tips: Follow these tips when using an external WebSphere MQ:

- Turn off MQ tracing in the MQ ZPARMS by specifying:.

  ```
  TRACSTR=NO,          TRACING AUTO START                X
  ```

  Example:

  ```
  //*
  //*            Assemble step for CSQ6LOGP
  //*
  //LOGP   EXEC PGM=ASMA90,PARM='DECK,NOOBJECT,LIST,XREF(SHORT)',
  //            REGION=4M
  //SYSLIB   DD DSN=MQSERIES.V5R3M0.SCSQMACS,DISP=SHR
  //         DD DSN=SYS1.MACLIB,DISP=SHR
  //SYSUT1   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
  //SYSPUNCH DD DSN=&&LOGP;,
  //           UNIT=SYSDA,DISP=(,PASS),
  //           SPACE=(400,(100,100,1))
  //SYSPRINT DD SYSOUT=*
  //SYSIN    DD *
           CSQ6LOGP INBUFF=60,         LOG INPUT BUFFER SIZE (KB)        X
                 MAXRTU=2,             MAX ALLOCATED ARCHIVE LOG UNITS   X
                 DEALLCT=0,            ARCHIVE LOG DEALLOCATE INTERVAL   X
                 MAXARCH=500,          MAX ARCHIVE LOG VOLUMES           X
                 OFFLOAD=YES,          ARCHIVING ACTIVE                  X
                 OUTBUFF=4000,         LOG OUTPUT BUFFER SIZE (KB)       X
                 TWOACTV=YES,          DUAL ACTIVE LOGGING               X
                 TWOARCH=YES,          DUAL ARCHIVE LOGGING              X
                 TWOBSDS=YES,          DUAL BSDS                         X
                 WRTHRSH=20            ACTIVE LOG BUFFERS
           END
  /*
  //*
  //*            Assemble step for CSQ6ARVP
  //*
  //ARVP   EXEC PGM=ASMA90,COND=(0,NE),
  //            PARM='DECK,NOOBJECT,LIST,XREF(SHORT)',
  //            REGION=4M
  //SYSLIB   DD DSN=MQSERIES.V5R3M0.SCSQMACS,DISP=SHR
  //         DD DSN=SYS1.MACLIB,DISP=SHR
  //SYSUT1   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
  //SYSPUNCH DD DSN=&&ARVP;,
  //           UNIT=SYSDA,DISP=(,PASS),
  //           SPACE=(400,(100,100,1))
  //SYSPRINT DD SYSOUT=*
  ```

```
//SYSIN   DD *
         CSQ6ARVP ALCUNIT=BLK,       UNITS FOR PRIQTY/SECQTY           X
               ARCPFX1=WIT.MQ1.WITA, DSN PREFIX FOR ARCHIVE LOG 1      X
               ARCPFX2=WIT.MQ2.WITA, DSN PREFIX FOR ARCHIVE LOG 2      X
               ARCRETN=9999,         ARCHIVE LOG RETENION TIME (DAYS)  X
               ARCWRTC=(1,3,4),      ARCHIVE WTO ROUTE CODE            X
               ARCWTOR=NO,           PROMPT BEFORE ARCHIVE LOG MOUNT   X
               BLKSIZE=24576,        ARCHIVE LOG BLOCKSIZE             X
               CATALOG=YES,          CATALOG ARCHIVE LOG DATA SETS     X
               COMPACT=NO,           ARCHIVE LOGS COMPACTED            X
               PRIQTY=4320,          PRIMARY SPACE ALLOCATION          X
               PROTECT=NO,           DISCRETE SECURITY PROFILES        X
               QUIESCE=5,            MAX QUIESCE TIME (SECS)           X
               SECQTY=540,           SECONDARY SPACE ALLOCATION        X
               TSTAMP=YES,           TIMESTAMP SUFFIX IN DSN           X
               UNIT=DASD,            ARCHIVE LOG DEVICE TYPE 1         X
               UNIT2=                ARCHIVE LOG DEVICE TYPE 2
         END
/*
//*
//*             Assemble step for CSQ6SYSP
//*
//SYSP   EXEC PGM=ASMA90,COND=(0,NE),
//             PARM='DECK,NOOBJECT,LIST,XREF(SHORT)',
//             REGION=4M
//SYSLIB   DD DSN=MQSERIES.V5R3M0.SCSQMACS,DISP=SHR
//         DD DSN=SYS1.MACLIB,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSPUNCH DD DSN=&&SYSP;,
//            UNIT=SYSDA,DISP=(,PASS),
//            SPACE=(400,(100,100,1))
//SYSPRINT DD SYSOUT=*
//SYSIN   DD *
         CSQ6SYSP CTHREAD=600,       TOTAL NUMBER OF CONNECTIONS        X
               CMDUSER=CSQOPR,       DEFAULT USERID FOR COMMANDS       X
               EXITLIM=30,           EXIT TIMEOUT (SEC)                X
               EXITTCB=8,            NUMBER OF EXIT SERVER TCBS        X
               IDBACK=500,            NUMBER OF NON-TSO CONNECTIONS     X
               IDFORE=100,           NUMBER OF TSO CONNECTIONS         X
               LOGLOAD=900000,       LOG RECORD CHECKPOINT NUMBER      X
               OTMACON=(,,DFSYDRU0,2147483647,CSQ),  OTMA PARAMETERS   X
               QMCCSID=0,            QMGR CCSID                        X
               QSGDATA=(,,,),        QUEUE-SHARING GROUP DATA          X
               RESAUDIT=YES,         RESLEVEL AUDITING                 X
               ROUTCDE=1,            DEFAULT WTO ROUTE CODE            X
               SMFACCT=NO,           GATHER SMF ACCOUNTING             X
               SMFSTAT=NO,           GATHER SMF STATS                  X
               STATIME=30,           STATISTICS RECORD INTERVAL (MIN)  X
               TRACSTR=NO,          TRACING AUTO START                X
               TRACTBL=99,           GLOBAL TRACE TABLE SIZE X4K       X
               WLMTIME=30,           WLM QUEUE SCAN INTERVAL (SEC)     X
               SERVICE=0             IBM SERVICE USE ONLY
         END
/*
//*
//*  LINKEDIT CSQARVP, CSQLOGP and CSQSYSP into a
//*  system parameter module.
//*
//LKED   EXEC PGM=IEWL,COND=(0,NE),
//      PARM='SIZE=(900K,124K),RENT,NCAL,LIST,AMODE=31,RMODE=ANY'
//*
//*   OUPUT AUTHORIZED APF LIBRARY FOR THE NEW SYSTEM
//*   PARAMETER MODULE.
//*
//SYSLMOD  DD DSN=SYS1.WITA.LINKLIB,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,DCB=BLKSIZE=1024,
//            SPACE=(1024,(200,20))
```

```
//SYSPRINT DD SYSOUT=*
//ARVP     DD DSN=&&ARVP;,DISP=(OLD,DELETE)
//LOGP     DD DSN=&&LOGP;,DISP=(OLD,DELETE)
//SYSP     DD DSN=&&SYSP;,DISP=(OLD,DELETE)
//*
//*   LOAD LIBRARY containing the default system
//*   parameter module (CSQZPARM).
//*
//OLDLOAD  DD DSN=MQSERIES.V520.SCSQAUTH,DISP=SHR
//SYSLIN   DD *
   INCLUDE SYSP
   INCLUDE ARVP
   INCLUDE LOGP
   INCLUDE OLDLOAD(CSQZPARM)
 ENTRY CSQZMSTR
 NAME CSQZPARM(R)                    Your system parameter module name
/*
```

- Similarly, you are advised to turn off tracing in the channel initiator. Unlike base MQ tracing, this parameter cannot be enabled dynamically. To turn tracing back on for debug purposes, you will need to reassemble your MQ XPARMS. Enabling channel initiator tracing can degrade your system by 5-10%.

  The variable to set is as follows:

  ```
  TRAXSTR=NO,       START TRACE AUTOMATICALLY  YES|NO
  ```

- For improved performance in the laboratory, the following WebSphere MQ tuning parameters required modification from their default settings.The difference in most cases was significant.
  - Optimize the number of concurrent queue manager connections

    ```
    CTHREAD parameter of CSQ6SYSP (Maximum number of  concurrent
                                   connections to MQ)
    IDBACK parameter of CSQ6SYSP (Maximum number of background concurrent
                                  threads connected to MQ)
    ```

    CTHREAD is the maximum number of simultaneous connections to the queue manager. It should be greater than, or equal to, the sum of IDFORE and IDBACK. IDFORE is the number of concurrent TSO connections to WebSphere MQ, and IDBACK is the number of concurrent background connections, which includes jms threads. If any or all of these parameters are too low, applications will be unable to connect to the queue manager. These parameters are found in the CSQ6SYSP section of the MQ ZPARMS. You can update the MQ ZPARMS at any time; the updates will take effect the next time you restart the queue manager.
  - Adjust the WebSphere MQ checkpoint interval and active log buffers

    ```
    LOGLOAD parameter of CSQ6SYSP (number of log records written
                                   before a checkpoint)
    WRTHRSH parameter of CSQ6LOGP (Number of active log buffers)
    ```

    When using persistent messages, it is particularly important to pay attention to logging characteristics. MQ logs should always be placed on high performance volumes with DASD fast write enabled. MQ logs are often the single most significant bottleneck when using persistent messages. The LOGLOAD parameter controls the number of log records written before a checkpoint (assuming, of course, that the log is large enough to hold this number of records). Checkpoints generally should occur no more frequently than every 5 or 6 minutes. If your MQ checkpoints are more frequent, you may need to increase either the size of the logs, the value of LOGLOAD, or both. In the laboratory, we use a LOGLOAD value of 900000 because we execute high throughput jms workloads that are very write-intensive. When the space on the log is exhausted, a log switch occurs which disrupts performance. You can avoid frequent log switches by increasing the size of the log(s).

WRTHRSH is the number of active log buffers, and determines how much data is held in memory before a log write occurs. If you have a high I/O rate to your log volume(s), you may wish to increase this parameter. In the laboratory, we use a value of 200.

Generally speaking, your message rate for persistent messages cannot exceed the bandwidth capacity of your slowest log volume. For example, if your were sending messages of 5KB at a throughput rate of 300 per second, you would be writing at least 1.9 MB of data per second to the log (this is roughly 1.3KB plus the user message size for each logged message).

– Specify the size of the archive logs

```
SECQTY parameter of CSQ6ARVP  (archive log space allocation)
PRIQTY parameter of CSQ6ARVP  (archive log space allocation)
ALCUNIT parameter of CSQ6ARVP  (archive Log allocation unit)
TWOARCH parameter of CSQ6LOGP  (dual archive logs)
```

The PRIQTY and SECQTY parameters control the size of the archive logs. Generally it is best to allocate them (ALCUNIT) in cylinders in lieu of blocks. Depending upon your data integrity requirements, you may or may not choose to have dual archive logging.

– Specify the number of buffers

```
BUFFERS  (number of buffers on the DEFINE BUFFERPOOL statement)
```

Use the DEFINE BUFFERPOOL statement to specify the number of buffers. It is important to insure the number of buffers is large enough to hold at least an entire message (and its headers). Otherwise, WebSphere MQ will be forced to write to the Page Data sets for every message. For example, a 100MB message requires at least 26000 pages in the bufferpool. MQ Buffer Manager statistics can be used to determine the number of times a buffer was unavailable. See Support Pac MP1B, MQSeries for OS/390 V5.2 - Interpreting accounting and statistics data.

– Specify queue definitions

```
INDXTYPE(NONE)  (index specification for queue definitions) or
INDXTYPE(CORRELID)
DEFPSIST(NO)
```

Unless your applications retrieve messages by other than correlation ID (which is the case for jms publish/subscribe) or message ID, it is normally best not to specify message selectors on queue definitions. You should, however, make sure you have specified INDXTYPE(CORRELID) on the SYSTEM.JMS.ND/D queues, and/or on the SYSTEM.JMS.ND.CC / .D.CC queues, and/or on any shared message queues for publish/subscribe. Specify DEFPSIST(NO) unless you want messages on a particular queue to default to persistent. Since DEFPSIST(YES) will affect performance, make sure you really want persistent messages.

– Specify channel definitions

```
BATCHSZ parameter for queue definitions (Number of messages
                                          sent as a batch)
```

We did not modify the BATCHSZ parameter in the laboratory.

JMS Tips: Unlike WebSphere MQ, there are no global jms tuning parameters that you can enable to optimize jms performance. Tuning recommendations for jms (as differentiated from WebSphere MQ tuning recommendations) involve the design of the jms application. The following are recommended for improved performance:

- Create long-lived queue manager connections rather than creating and destroying connections on every message.
- Use bindings mode if your queue manager and client both reside on the same zOS image. Bindings mode is a cross-memory interface that eliminates the need for MQ to call TCP/IP. It can be implemented by not setting the TransportType on the connection.
- Client acknowledgements required that an ack be received from the client before another message is sent. Auto acknowledgements are a better choice and reduce delays.
- Performance will be best if the client and queue manager have the same CCSID so the queue manager does not need to translate message headers.
- Small messages are best. Use of system resources and throughput will be proportional to the size of MQ messages. However, if you must send large amounts of data, one larger message is preferable to multiple small ones. If using very large messages (for example, over 1MB) see ″WebSphere MQ Tips″ on page 87.
- Persistent, transacted messages perform better than persisted non-transacted messages because multiple MQ commits can be delayed until the end of the transaction.
- Express (nonpersistent) messages perform most optimally, so use them if your application does not require persistence.
- MQ Application Server Framework (ASF) generally adds more overhead then non-ASF messaging.
- MQ local queues defined as `DEFSOPT(SHARED)` with the `SHARE` option, and shared by multiple threads or processes generally perform better than non-shared queues and use fewer resources.

Tuning tips for GRS

WebSphere for z/OS uses GRS to communicate information between servers in a sysplex. When there are multiple servers defined in a system or a sysplex, a request may end up on the wrong server. To determine where the transaction is running we use GRS. Therefore, if you are using global transactions, we will issue an enqueue for that transaction at the start of the transaction and hold on to that enqueue until the transaction ends. WebSphere for z/OS uses GRS enqueues for the following:

- Two-phase commit transactions involving more than one server
- HTTP sessions in memory
- Stateful EJBs
- ″Sticky″ transactions to keep track of pseudo-conversational states.

Tips:

- If you **are not** in a sysplex, you should configure `GRS=NONE`.
- If you **are** in a sysplex, we strongly recommend `GRS=STAR`.

This requires configuring GRS to use the coupling facility. See the GRS documentation for details on setting this up.

Tuning tips for JVM

Before you begin:

1. Ensure that you have the most recent version of JVM that is supported by WebSphere for z/OS. As of this writing, the JVM level for WebSphere Application Server for z/OS V5 is 1.3.1 and PTF 17.
2. Have the most recent PTFs, since almost every PTF level has improved performance of the JVM.
3. Have sufficient `JVM Heap Size`. Refer to "Garbage collection and `JVM Heap Size`" on page 73 for a discussion of this setting.

How to view or set: Use the WebSphere Administrative console :

1. Click **Servers > Application Servers > `server_name` > Process Definition > Java Virtual Machine**
2. Select the options listed in the Tips section belong on the Configuration Tab.

Tips:

- Run with the JIT (Just In Time) compiler active.

  In the General Properties section of the Configuration Tab, ensure that Disable JIT is not selected. The default is JIT support enabled.
- Do not specify the debug version of the JVM `libjava_g` in your libpath. The debug version will not perform as well as the non-debug version.
- To speed up JVM initialization and improve server startup time, specify the following command line arguments in the General JVM Arguments field in the General Properties section of the Configuration Tab.

  ```
  -Xquickstart
  -Xverify:none
  ```

  This can reduce servant startup by as much as 40% at a cost of about 8% reduction in runtime.
- Sometimes poor performance is caused by a missing class. The class loader will look in it's tables of already loaded classes and if the class is not found to be already loaded it will search for it. This search process can cause a high amount of I/O activity to the HFS volumes. To determine if this is the problem you can collect CTRACE records from the file system. Once you determine which class is not being found you can repair the problem by providing the class or by removing the need for it.

  Note: Please see the ″Applications″ section in the WebSphere Application Server for z/OS V5.0 InfoCenter, access to which can be obtained through the WebSphere forz/OS library Web site http://www.ibm.com/software/webservers/appserv/zos_os390/library.html for more information on ″Application client troubleshooting tips.″

Notes:

1. For more information about JVM performance on z/OS and OS/390, see `http://www.s390.ibm.com/java/perform.html`.
2. For information about the garbage collector, `JVM heap size` and enabling verbose garbage collection, see "Garbage collection and `JVM Heap Size`" on page 73.

Tuning tips for CICS

These recommendations only apply to WebSphere aplications that access CICS.

The LGDFINT system initialization parameter specifies the log defer interval used by CICS log manager when determining how long to delay a forced journal write request before invoking the MVS system logger. The value is specified in milliseconds. Performance evaluations of typical CICS transaction workloads have shown that the default setting of 5 milliseconds gives the best balance between response time and central processor cost. Be aware that CICS performance can be adversely affected by a change to the log defer interval value. Too high a value will delay CICS transaction throughput due to the additional wait before invoking the MVS system logger. An example of a scenario where a reduction in the log defer interval might be beneficial to CICS transaction throughout would be where many forced log writes are being issued, and little concurrent task activity is occurring. Such tasks will spend considerable amounts of their elapsed time waiting for the log defer period to expire. In such a situation, there is limited advantage in delaying a call to the MVS system logger to write out a log buffer, since few other log records will be added to the buffer during the delay period.

Tips:

- Set the `LGDFINT` system initialization parameter to 5.

  While CICS is running, you can use the `CEMT SET SYSTEM[LOGDEFER(value)]` command to alter the `LGDFINT` setting dynamically.
- Set the CICS `RECEIVECOUNT` value high enough to handle all concurrent EXCI pipes on the system.

  The default value is 4. You set this value in the EXCI sessions resource definition.

For more detailed information on CICS, refer to the *CICS Performance Guide*, SC34-6009.

Tuning the WebSphere for z/OS runtime

The first thing to do is review the WebSphere for z/OS configuration. One simple way to do this is to look in your application control and server regions in SDSF. When each server starts, the runtime prints out the current configuration data in the joblog. Refer to for an example of a joblog.

Tips for internal tracing:

1. To start, you should ensure that you are not collecting more diagnostic data than you need. You should check your WebSphere for z/OS tracing options to ensure that `ras_trace_defaultTracingLevel=0` or `1`, and that `ras_trace_basic` and `ras_trace_detail` are not set.

   How to view or set: Use the WebSphere adminstrative console:

   a. Click **Environment>Manage WebSphere Variables**.

   b. On the Configuration Tab check for any of these variables in the name field and observe the variable setting in the value field.

   c. To change or set a variable, specify the variable in the name field and specify the setting in the value field. You can also describe the setting in the description field on this tab.

   Note: Please see the ″Troubleshooting″ section in the WebSphere Application Server for z/OS V5.0 InfoCenter, access to which can be obtained through the WebSphere forz/OS library Web site http://www.ibm.com/software/webservers/appserv/zos_os390/library.html for more information on trace options and settings.

2. If you use any level of tracing, including `ras_trace_defaultTracingLevel=1`, ensure that you set `ras_trace_outputLocation` to `BUFFER`. `ras_trace_defaultTracingLevel=1` will write exceptions to the trace log as well as to the ERROR log.
3. To reduce memory requirements, you can set the `ras_trace_BufferCount=4` and `ras_trace_BufferSize=128`, which will get 512KB of storage for the trace buffers (the minimum allowed).
4. If you are tracing to sysprint with `ras_trace_defaultTracingLevel=3`, you may experience an almost 100% throughput degradation. If you are tracing to CTRACE, however, you may only experience a 15% degradation in throughput. For this reason, it is best to trace to CTRACE.

Security

As a general rule, two things happen when you increase security: the cost per transaction increases and throughput decreases.

By default, WebSphere for z/OS runs with security on. The runtime will always incur a small price to collect and carry the security credential information for users and the server. Since all the security authorization checks are done with SAF (RACF or equivalent), you can choose to enable and disable SAF classes to control security. A disabled class will cost a negligible amount of overhead.

When a class is active, the number of profiles in a class will affect the overall performance of the check. Placing these profiles in a (RACLISTed) memory table will improve the performance of the access checks. Audit controls on access checks also affect performance. Usually, you audit failures and not successes. Audit events are logged to DASD and will increase the overhead of the access check.

If you are using EJBROLEs, specifying more roles on a method will lead to more access checks that need to be executed and a slower overall method dispatch. If you are not using EJBROLEs, do not activate the class.

Java 2 Security

If you do not need Java 2 security, you should disable it.

Disabling Java 2 security for a cell: How to view or set: Use the WebSphere adminstrative console:

1. Click **Security>Global Security** in the navigation tree.
2. On the Configuration Tab ensure that the checkbox next to ″Enforce Java 2 Security″ is not checked.
3. Click OK or Apply on the Global Security page.
4. Save the changes and make sure a file sync is performed before restarting the servers.
5. For the changes to take effect, restart all the servers, which include the Network Deployment Manager, all Node Agents, and all application servers.

Disabling Java 2 security for an application server: How to view or set: Use the WebSphere adminstrative console:

1. Click **Server>Application Servers** in the navigation tree. The Application Servers page appears.
2. Click the **application server name** in the **Name** column of the Application Server collection table. The configuration panel of the application server selected appears.

3. On the configuration panel, click **Server Security** in the Additional Properties section. The Server Security panel of the application server appears.
4. Click **Server Level Security** in the Additional Properties section. The Server Level Security panel of the application server appears.
5. On the Configuration Tab ensure that the checkbox next to ″Enforce Java 2 Security″ is not checked.
6. Click OK or Apply on the Server Levell Security page.
7. Save the changes and make sure a file sync is performed before restarting the servers.
8. For the changes to take effect, restart the application server.

Authentication

You have several options when dealing with authentication:

- **Local authentication:** Local authentication is the fastest type because it is highly optimized.
- **UserID and password authentication:** Authentication that utilizes a userID and password has a high first-call cost and a lower cost with each subsequent call.
- **Kerberos security authentication:** We have not adequately characterized the cost of kerberos security yet.
- **SSL security authentication:** SSL security is notorious in the industry for its performance overhead. Luckily, there is a lot of assists available from hardware to make this reasonable on z/OS. Refer to

Note: For more information on security, see *WebSphere Application Server for z/OS V5.0: Installation and Customization*, GA22-7909.

Servlet / EJB integrated runtime

The Servlet / EJB integrated runtime is when you are running the servlet engine in WebSphere for z/OS rather than in DGW. The engine is normally set up to run in DGW (and pass information back and forth to WebSphere for z/OS) because it is faster than the alternative of running straight in WebSphere for z/OS. The latter setup is more powerful, however, and the time difference will be greatly noticeable only if performing trivial functions like the ping method.

If you are running just a servlet, the integrated runtime may not initially show an improvement in performance. However, when a servlet is calling on an EJB, it will benefit greatly from the integrated runtime. Essentially, the integrated runtime will convert the remote method calls to local in-process EJB invocations (which are much faster).

The Servlet / EJB integrated runtime provides functions such as access to transactional resource managers and it is the only way to get true J2EE compliance with WebSphere for z/OS.

HTTP internal transport

The HTTP internal transport is an option in the 4.0.1 runtime. This allows HTTP work to come directly to the 4.0.1 runtime rather than through another server. There are significant performance benefits to running with the HTTP internal transport. For more information, see the ″Application Programming″ section in the WebSphere Application Server for z/OS V5.0 InfoCenter, access to which can be obtained through the WebSphere forz/OS library Web site
http://www.ibm.com/software/webservers/appserv/zos_os390/library.html.

IBM HTTP server (IHS) tuning tips

For information on tuning IBM HTTP server (IHS) refer to the following Web site: http://www-3.ibm.com/software/webservers/httpservers/doc/v51/2tabcont.htm

Tuning for J2EE applications

In this section on tuning J2EE applications, we discuss the following points:

- Topology
- J2EE container and applications
- Programming tips

Topology

Topology can have a significant effect on WebSphere performance. This section describes some of the topology considerations you should be aware of when configuring and installing WebSphere for z/OS.

Single server or multiple servers?

WebSphere for z/OS gives you the ability to install your application either in a single server or spread it across multiple servers. There are many reasons for partitioning your application. However, for performance, placing your application all in the same server will always provide better performance than partitioning it. If you do choose to partition your application across servers, you will get better performance if there are at least replica servers on each system in the sysplex. See Figure 2. The WebSphere for z/OS runtime will try to keep calls local to the system if it can, which will, for example, use local interprocess calls rather than sockets.

Figure 2. Having applications on a single system avoids calls from one system to another

Local client or remote client?

On a local client, the client and the optimized communication are done on the same system. This has some additional client CPU costs but less communication cost. On a remote client, the client cost is replaced by the additional communication overhead of sockets. The CPU cost on either system is almost equivalent. Latency is better for a local client than for a remote client, meaning you will get better response time with a local client.

One copy of a server or many replicas?

You can define more than one copy of a server on a system. These copies are called replicas. We have found slight improvements in performance when running with a couple of replicas as opposed to just one (very large configuration). While there is some benefit, IBM does not recommend, at this time, the creation of replicated control regions for the sole purpose of improving performance. We do, however, recommend them for eliminating a single point of failure and for handling rolling upgrades without introducing an outage.

WAS 4.x Integrated Runtime

If the HTTP server makes calls to EJBs, you can provide a major performance enhancement by having the Java servlets and JSPs installed in the EJB container on the WAS 4.x runtime application server instance.

Using the HTTP internal transport

It is more efficient to have the HTTP internal transport installed on the WAS 5.0 runtime application server instance.

J2EE container and applications

Container configuration overview

In WebSphere for z/OS, there are several types of EJBs and several transaction policies supported. Selection of each type has performance implications. While we won't be able to give an exhaustive treatise on this yet, we will give some basic rules.

EJBs: There are two basic bean types in WebSphere for z/OS: session and entity.

Session beans: Within a session bean in WebSphere for z/OS, there are stateless and stateful session beans.

Stateless session bean
The lowest overhead type of bean. They are cheap to create, do very little automatically and, if not cleaned up by the application, will go away when the server terminates.

Stateful session bean
The default for a stateful session bean is to not harden its state to a backing store except in the case of a controlled server shut down. In this configuration, a stateful bean is slightly more overhead than a stateless bean. The configuration overhead of hardening the stateful bean state at the end of each transaction has yet to be quantified.

Entity beans: In WebSphere Application Server for z/OS V5, entity beans come in two flavors: bean managed persistence (BMP) and container managed persistence (CMP).

Since managing persistence is the responsibility of the bean in BMP, it really depends on the way the load and store is implemented whether a BMP is faster than a CMP. CMP beans manage persistence. The CMP bean implementation is highly optimized and will often produce better performance than a typical BMP bean. Additional improvements are expected which will add even more flexibility to CMP beans.

Note: Marking a bean method with the `readonly` extended deployment descriptor will cause the runtime to avoid writing the state of the bean back out. Specify this value on a method when you know that the method does not update any attributes of the bean.

Transaction policies: There are seven transaction policies in WebSphere for z/OS:

- `TRANSACTION_REQUIRES`
- `TRANSACTION_REQUIRES_NEW`
- `TRANSACTION_SUPPORTS`
- `TRANSACTION_NOT_SUPPORTED`
- `TRANSACTION_BEAN_MANAGED`
- `TRANSACTION_NEVER`
- `TRANSACTION_MANDATORY`

Within this specification, we also have local transaction and global transactions. Generally, local transactions are the fastest.

Container Managed Persistence (CMP) Connection and Prepared Statement Pooling

Two methods of CMP Connection Pooling: There are two ways to pool—within a transaction and across transactions:

″Within″ transaction pooling

Pooling within a transaction implies that the pool exists for the life of the transaction only.

Note: Each transaction comes with a new database connection and a new set of prepared statements.

″Across″ transaction pooling

Pooling across transactions involves a thread-level pool in which connections and their associated prepared statements are held from one transaction to the next. Pooling across transactions is the default and is recommended for best performance.

Note: Typically, there is one connection per database per thread and three threads per server region.

JVM properties related to CMP Connection Pooling: There are four JVM properties (set in the jvm.properties file) related to CMP Connection Pooling:

- com.ibm.websphere.cmp.connection.policy=*string*
- com.ibm.websphere.cmp.<Location-Name>.connection.timeout=*time*
- com.ibm.websphere.cmp.<Location-Name>.preparedStatement.poolsize=*size*
- com.ibm.websphere.cmp.<Location-Name>.connection.maxTrans=#*ofTransactions*

  By default, CMP will close and reopen the connection every 100 transactions to ensure that DB2 statistics are captured. If you set the value to 0, the connection is never closed which will give the best performance, however, DB2 statistics will not be captured. Setting higher values decreases the overhead, but also decreases the granularity of your DB2 data. We recommend that you set the value as high as your environment allows.

The variable text, <Location-Name>, correlates a property setting to a specific datasource definition from the Administration application. It matches the ″Location Name″ value specified on the datasource definition from the Administration application.

Note: See the ″Application Programming″ section in the WebSphere Application Server for z/OS V5.0 InfoCenter, access to which can be obtained through the WebSphere forz/OS library Web site http://www.ibm.com/software/webservers/appserv/zos_os390/library.html for more information on these and other JVM properties.

Database object characteristics: Both connections and prepared statements have certain characteristics when applied to CMP Connection Pooling.

Connections

- A connection to a remote database can time out whereas a connection to a local database can not.
- Accounting records are not generated until the connection is closed.
- A connection can be used by only a single transaction at a time.
- Connections are a scarce resource (about 600 per database).

Prepared statements

- A prepared statement does not time out, but its lifecycle corresponds to the lifecycle of its associated connection.
- A prepared statement is a scarce resource (100, by default, per database connection).

J2EE application programming tips

These programming tips relate to the following topics:

- JavaServer pages (JSPs)
- Java Database Connectivity (JDBC)

JSP topic

Tips:

1. Disable session state of JSPs.

   <%@ page language=″java″ contentType=″text/html″ session=″false″ %>
2. By default, JSPs will save session state

   <%@ page language=″java″ contentType=″text/html″ %>

JDBC topic

Tips:

1. Make sure you are at the current JDBC level. Refer to “DB2 tuning for WebSphere” on page 82.
2. Use prepared statements to allow dynamic statement cache of DB2 on z/OS.
3. Don’t include literals in the prepared statements, use a parameter marker ″?″ to allow dynamic statement cache of DB2 on z/OS.
4. Use the right getxxx method by each data type of DB2.
5. Turn auto commit off when just read-only operations are performed.
6. Use explicit connection context objects.
7. When coding an iterator, you have a choice of named or positioned. For performance, we recommend positioned iterators.
8. Close prepared statements before reusing the statement handle to prepare a different SQL statement within the same connection.
9. As a bean developer, you have the choice of JDBC or SQLJ. JDBC makes use of dynamic SQL whereas SQLJ generally is static and uses pre-prepared plans. SQLJ requires an extra step to create and bind the plan whereas JDBC does not. SQLJ, as a general rule, is faster than JDBC.
10. With JDBC and SQLJ, you are better off writing specific calls that retrieve just what you want rather than generic calls that retrieve the entire row. There is a high per-field cost.

Chapter 10. WebSphere for z/OS performance monitoring procedures

This chapter describes general WebSphere for z/OS performance monitoring procedures for J2EE applications.

The purpose of this chapter is to assist you in collecting the SMF/RMF data that IBM Service needs to help you debug performance issues with your configuration or application. This chapter is not intended to be a description of how to run your production SMF/RMF data. For information on collecting SMF/RMF data on a production system, refer to <need a reference here>.

Note: This chapter does NOT deal with DGW issues or how to cover the standalone servlet runtime. For more information on that, please go to this Web site: `http://www.ibm.com/s390/ebusiness/perform.html`

Performance information and accounting

WebSphere for z/OS relies on its use of WLM services to collect some of the accounting and performance data. This information gets presented back to the installation through RMF and RMF-written SMF records. In addition, WebSphere for z/OS has its own SMF records which collect additional domain-specific information for WebSphere for z/OS. First, unless you need the SMF records or RMF data, turn them off using the Administration application and the `SMFPRMxx` parmlib statement. You also use the `SMFPRMxx` parmlib statement to control the detail of the WebSphere for z/OS SMF records. If you do need SMF information, you should review the SMF parmlib to ensure you are collecting only the data that you need (both record types and detail).

Note: Refer to Chapter 11, “Systems Management Facility (SMF) recording and monitoring”, on page 111 for more information on SMF.

Setting up your workload manager goals and filtering criteria is probably beyond the scope of this section. However, you should recognize that you can classify work into service classes based on userid and server name. You should ensure that you classify the control regions as reasonably high-performing system tasks. Refer to “Tuning tips for WLM” on page 80.

Capturing WebSphere performance information

Through a combination of SMF recording and RMF, you can capture information about WebSphere performance and its resource utilization. .

- SMF record types 30, 70-79, 120, and others, plus various recording options such as the duration of recording intervals
- WLM workload classification and reporting classes
- RMF Workload Activity reports and RMF Monitor III facility

SMF record types

The following SMF record types are helpful in evaluating performance problems.

- 30 - Address Space Activity - these are usually captured by most installations and are used for monitoring or change-back of z/OS resource utilization for work that can be identified by one ore more address spaces. WebSphere transactions, however run in enclaves, and therefore are not counted in these records, but can be tracked in the type 72 records. SMF type 30 records include enclave (SMF30ENC) and task time (SMF30CPT = step CPU time = TCB + enclave + SRB time)

- 70-79 are used by RMF. The most interesting ones here are 70 (CPU) & 72 (Workload Activity) but the others in this range can be used to monitor use of other resources (DASD, real storage, and so forth).
- 120 - WebSphere server and container records contain information specific to WebSphere and WebSphere applications.
- Other record types for other subsystems such as System Logger (88), Unix System Services (34, 35, & 92), DB2 (100-102), CICS (110), and so forth.

WLM workload classification and reporting classes

You should classify your WebSphere work using the WLM CB-type classification criteria which includes the WebSphere server (and server instance), the Transaction Class, and the userid under which the work is running. This is done through a combination of the WLM Workload Classification rules in the WLM ISPF dialog panels. The Transaction Class can be assigned in the WebSphere Transaction Mapping file according to the HTTP virtual host, port number and URI (Universal Resource Identifier - encoded address for any resource on the Web).

Setting up WLM reporting classes

You should set up WLM Reporting Classes so you can monitor the resource utilization of your WebSphere workloads. These can be much more granular than service classes. This is important because you do not want to create too many service classes for WLM to manage, and each service class must run in its own servant region. Using RMF Workload Activity reports, you can see the CPU usage and other resource consuption according to report class in the RMF monitor 1 workload activity reports.

Here are some examples of different reporting classes which can be used to obtain system and workload characteristics.

- WebSphere J2EE applications, which will include any work done via JDBC to DB2 or other JDBC 2.0 resource providers.
- WebSphere system servers (node agents, daemons, and deployment managers).
- WebSphere controller region.
- WebSphere servant regions (this does not include the actual tranactions which are accounted for the CB-type work, but would account for housekeeping work such as java garbage collection and JSP compiles).
- CICS, IMS, MQSeries, or other resource managers called upon by WebSphere transactions.

Example: The following table shows classification rules for CB-type work in which the default service class is WSMED and has a reporting class of RWSDEFLT. Work run in the WSPROD WebSphere server is classified as WSMED with a reporting class of RWSPROD, unless it has a transaction class of TCLASS1, TCLASS2, or TCLASS2 assigned through the transaction class mapping file below.

```
  Qualifier  Qualifier  Start         Service  Report
# type       name       position      Class    Class
- ---------  --------   --------      -------- --------
                             Default: WSMED    RWSDEFLT
1 CN         WSPROD     1             WSMED    RWSPROD
2 . TC       . TCLASS1                WSFAST   RWSPRD1
2 . TC       . TCLASS2                WSMED    RWSPRD2
```

```
2 . TC       . TCLASS5                WSSLOW   RWSPRD5
1 CN         WSTEST     1             WSSLOW   RTSTEST
2 . UI       . USER1                  WSMED    RTSTSTU2
2 . TC       . TCLASS5                WSSLOW   RTSTST5
```

The following table shows how work can be assigned a transaction class based on its host name, port number, or URI. For example, a web request of http://ibm.com:80/Webap1/myservlet handled by the WSPROD server would be assigned a transaction class of TCLASS1, a service class of WSFAST, and a reporting class of RWSPRD1 by the classification rules shown above.

```
Transaction Class Mapping file format:
TransClassMap  <host>:<port>   <uritemplate>      <tclass>
TransClassMap  www.ibm.com:80  /Webap1/myservlet  TCLASS1
TransClassMap  www.ibm.com:*   /Webap1/myservlet  TCLASS2
TransClassMap  *:443           *                  TCLASS3
TransClassMap  *:*             /Webap1/myservlet  TCLASS4
TransClassMap  www.ibm.com:*   /Webap5/*          TCLASS5
TransClassMap  *               *                  TCLASS6
```

For more information on WLM classification rules refer to *z/OS MVS Planning: Workload Management*. For more information on the transaction class mapping facility, refer to the WebSphere Application Server for z/OS V5 InfoCenter, access to which can be obtained through the WebSphere for z/OS library Web site http://www.ibm.com/software/webservers/appserv/zos_os390/library.html.

RMF Workload Activity reports and RMF Monitor III

Performance metrics include transaction rates and response times. Resource utilization includes CPU, I/O (channel), and storage utilization.

Transactions/second
This is shown in the AVG, MPL, and AVG ENC fields which is equal to the average number of enclaves in the period in the RMF Monitor I report below.

Response times
The actual response times of the WLM transaction is shown in the TRANS.-TIME SS.TTT' column n the RMF Monitor I report below and are measured in milliseconds. (This also includes time waiting on WLM queue.)

Service Rates
Resource utilization in CPU service units, and Service Units per Sec. The APPL% field shows the number of processor engines (CPs) required to drive the work in the service (report) class.

Controller Regions Example: Here is the report for the WebSphere controller regions which were all assigned a particular reporting class according to the STC classification rules based on the started class jobname. There is only one transaction active in the system, there are no response time figures, and required 37.2% of a processor engine to support its work.

```
TRANSACTIONS    TRANS.-TIME  SS.TTT   ---SERVICE--   --SERVICE RATES--
AVG      1.00   ACTUAL            0   IOC        0   ABSRPTN     89615
MPL      1.00   EXECUTION         0   CPU   522567   TRX SERV    89615
ENDED       0   QUEUED            0   MSO   10159K   TCB          39.9
END/S    0.00   R/S AFFINITY      0   SRB    61728   SRB           4.7
#SWAPS      0   INELIGIBLE        0   TOT   10743K   RCT           0.0
EXCTD       0   CONVERSION        0   /SEC   89630   IIT           0.0
AVG ENC  0.00   STD DEV           0                  HST           0.0
REM ENC  0.00                                        APPL %       37.2
```

Servant Regions Example: Here is the report for the WebSphere servant regions which were all assigned a particular reporting class according to the STC classification rules based on the stared class jobname. There are only two transactions active in the system (meaning there were two servant regions), there are no response time figures, and they required 10% of a processor engine to support their work.

```
TRANSACTIONS    TRANS.-TIME  SS.TTT   ---SERVICE--   --SERVICE RATES--
AVG      2.00   ACTUAL            0   IOC        0   ABSRPTN    122075
MPL      2.00   EXECUTION         0   CPU   143957   TRX SERV   122075
ENDED       0   QUEUED            0   MSO   29113K   TCB          11.0
END/S    0.00   R/S AFFINITY      0   SRB    12460   SRB           1.0
#SWAPS      0   INELIGIBLE        0   TOT   29270K   RCT           0.0
EXCTD       0   CONVERSION        0   /SEC  244192   IIT           0.0
AVG ENC  0.00   STD DEV           0                  HST           0.0
REM ENC  0.00                                        APPL %       10.0
```

WebSphere Transactions (Enclaves) Example: Here is the report for the real WebSphere transaction work which runs as enclaves in this particular reporting class. The average number of these kind of transactions active in the system was 241.52, the average response time was 276 milliseconds, and required 2.129 processor engines to support this work.

```
TRANSACTIONS    TRANS.-TIME  SS.TTT   ---SERVICE--   --SERVICE RATES--
AVG    241.52   ACTUAL          276   IOC        0   ABSRPTN       115
MPL    241.52   EXECUTION       272   CPU    3343K   TRX SERV      115
ENDED  106717   QUEUED            4   MSO        0   TCB         255.5
END/S  890.32   R/S AFFINITY      0   SRB        0   SRB           0.0
#SWAPS      0   INELIGIBLE        0   TOT    3343K   RCT           0.0
EXCTD       0   CONVERSION        0   /SEC      17   IIT           0.0
AV ENC 241.52   STD DEV          66                  HST           0.0
REM ENC  0.00                                        APPL %      212.9
```

Delays Example: The QMPL field in the following report means that the server was waiting for a Server Region to select work off the WLM queue. Here is a queue delay report which shows that work was delayed 23.5% of the time waiting for the CPU and 12.6% of the time waiting for a servant region:

```
         EX   PERF   AVG   --USING%--  ----- EXECUTION DELAYS % -----
         VEL  INDX  ADRSP   CPU   I/O  TOTAL   CPU QMPL
GOAL    40.0%
ACTUALS 45.3%  .89   13.4   0.1   0.0   36.1  23.5 12.6
```

Started Class Classification Example: Here is an example of the STC classification rules/panel in WLM which can be used to classify the non-enclave time in the websphere server regions. WebSphere production controller region with a jobname of WSPRODC is assigned a service class of STCCR and reporting class of RCTLREG, production controller region with a jobname of WSPRODC is assigned a service class of STCSR and reporting class of RSRVREG, and testregions (servant and controllers) with a jobname starting with WSTare assigned a service class of STCWSTST and reporting class of RWSTST.

```
 Subsystem Type . . . . . . . : STC
 Description  . . . . . . . . . All started tasks
   -------Qualifier-------------            -------Class--------
   Type       Name     Start                Service     Report
   Type       Name     Start                Service     Report
                                  DEFAULTS: STCMED__    RSTCMED_
1  TNG        WSPRODC  ___                  STCCR___    RCTLREG_
1  TNG        WSPRODS  ___                  STCSR___    RSRVREG_
1  TNG        WST*     ___                  STCWSTST    RWSTST__
```

Steps for capturing performance data using RMF

Here is an example of running a short performance test in a sand-box environment.

Setting up SMF

Here is a sample SMFPRMxx member that will create interval records every 2 minutes, and record the following SMF record types:

- 30 - Address space
- 70-79 - RMF
- 82 - Crypto
- 88-90 - System Logger, Usage & System Data
- 101 - DB2
- 110 - CICS
- 120 - WebSphere

```
ACTIVE                      /*ACTIVE SMF RECORDING*/
    DSNAME(&SYSNAME..MAN1, &SYSNAME..MAN2)       /*TWO MAN DATASETS */
    LISTDSN                     /* LIST DATA SET STATUS AT IPL*/
    NOPROMPT                    /* DON'T PROMPT THE OPERATOR  */
    INTVAL(02)                  /* SMF GLOBAL RECORDING INTERVAL */
    SYNCVAL(00)                 /* GLOBAL SYNC VALUE            */
    MAXDORM(3000)               /* WRITE AN IDLE BUFFER AFTER 30 MIN*/
    STATUS(010000)              /* WRITE SMF STATS AFTER 1 HOUR*/
    SID(&SYSNAME(1:4))          /* USE SYSNAME AS SID     */
    SUBSYS(STC,INTERVAL(SMF,SYNC),
                          TYPE(0,30,70:79,88:90,101,110,120))
```

Set the SMF recording interval to 2 minutes by using the ’SET SMF=xx’ command to activate the SMFPRMxx member from SYSx.PARMLIB. Use the ’D SMF,O’ command to display the parameters in effect.

Use a tool like WSWS to simulate an application stress load.

While the transactions are running, switch to SDSF and RMF to observe the transactions.

SDSF

Use the SDSF DA panel to see how many application server address spaces are active, and observe at the CPU%, ECPU% and SIO rate. Use the ″ENC″ panel to see the enclaves running and what service classes they are running under.

RMF

RMF can usually be started with the simple ’S RMF’ command from the MVS console.

Here is a typical RMF procedure:

```
//RMF      PROC
//IEFPROC  EXEC PGM=ERBMFMFC,REGION=0M,PARM='MEMBER(XS)'
//IEFPARM  DD   DDNAME=IEFRDER
//IEFRDER  DD   DSN=SYS1.PARMLIB,DISP=SHR
```

The following is a copy of the PARMLIB member ERBRMFXS: (parmeters beginning with a /* are not used in this example but may be useful to you.

```
    CPU                       /* COLLECT CPU STATISTICS        */
         CHAN                      /* COLLECT CHANNEL STATISTICS    */
         CYCLE(1000)               /* SAMPLE AT 1 TIME / SECOND     */
         DEVICE(NOCHRDR)           /* NO CHARACTER RDR DEV STATS    */
         DEVICE(COMM)              /* ADDED COMM FOR 37X5           */
```

```
 DEVICE(DASD)               /* COLLECT DIRECT ACCESS DEVICE  */
 /* STATISTICS                    */
 DEVICE(NOGRAPH)            /* NO GRAPHICS DEVICE STATISTICS */
 DEVICE(NOTAPE)             /* NO TAPE DEVICE STATISTICS     */
 DEVICE(NOUNITR)            /* NO UNIT RECORD DEVICE STATS   */
 ENQ(SUMMARY)               /* ENQ REPORTING                 */
 INTERVAL(15M)              /* REPORT AT  15 MIN INTERVALS   */
 IOQ(DASD)                  /* I/O Q'ING FOR DEV IN LOG CU   */
 IOQ(COMM)                  /* I/O Q'ING FOR DEV IN LOG CU   */
 NOVSTOR                    /* NO RMF 3.2 AND LATER REL */
 OPTIONS                    /* OPERATOR MAY CHG RMF OPTIONS  */
 PAGING                     /* COLLECT PAGING STATISTICS     */
 PAGESP                     /* COLLECT PAGE/SWAP DATASET STAT*/
 RECORD                     /* RECORD INTO SMF DATASET       */
 NOSTOP                     /* STOP AFTER  90 MINUTES        */
 SYNC(SMF)                  /* INTERVAL SYNCED WITH SMF      */
 SYSOUT(H)                  /* SYSOUT CLASS OF OUTPUT REPORT */
 WKLD(PERIOD,SYSTEM)        /* COLLECT WORKLOAD MANAGER
                               STATISTICS AND REPORT AT THE
                               PERIOD LEVEL + TOTAL LINE     */
TRACE(CCVUTILP)
```

The Monitor III data gatherer can be started after RMF with the ′F RMF,S III′ modify command.

To use the RMF Monitor 3 display, go to the ″Sysplex Summary″ display as follows:

1. Type RMF on ISPF command line.
2. Type 3 to see Monitor III choices.
3. Type S to get Sysplex reports.
4. Type 1 to see the Sysplex Summary report.

You can scroll back and forth in time with PF10 and PF11, or over-type the time field.

Look at the transactions in the WebSphere service and reporting classs and note the Average Response time, Transactions per second, and Performance Index. You may also explore further in RMF Monitor III to see the ″System Information″ report.

WLM Delay Monitoring

WebSphere Application Server for z/OS V5 can use Workload Manager (WLM) services to report transaction begin-to-end response times and execution delay times. The WLM data collected by Resource Measurement Facility (RMF) is captured in two phases of the RMF report:

- BTE — the begin-to-end phase applies to requests handled by the controller
- EXE — the execution phase applies to requests handled by the servant

You can use this status information to determine where possible performance bottlenecks are occuring. This feature is available on z/OS R1V2 and above with WLM APAR OW51848 and RMF APAR OW52227.

When a new transaction enters the system, the WebSphere for z/OS application control region (ACR) starts the classify service. Delays associated with the WebSphere for z/OS ACR service class are counted separately for the BTE phase and the EXE phase. This support allows WLM to associate a performance block (PB) with an enclave to record delays that occur in the flow of a transaction. The state samples are collected on an ongoing basis and reported as a percentage of average transaction response time. Table 18 shows the states, their codes, the section of the RMF report where each is reported, the meaning, and suggested response. You can use this information in the RMF report to determine where some of your system's performance problems may be occurring.

Table 18. WLM delay monitoring states

State	Code	Report	Meaning	Response
ACTIVE	ACTIVE SUB	Both BTE and EXE	WebSphere is actively processing request	
ACTIVE_APPLIC	ACTIVE APPL	Both BTE and EXE	Application is running	Use application montoring tool to determine the cause of the delay.
WAITING TYPE1	TYP1	EXE	ejb collaborator delay	
WAITING TYPE2	TYP2	EXE	Resource manager delay	Called a J2C connector to perhaps DB2, CICS, IMS. Investigate other resource manager using their monitoring tools.
WAITING TYPE3	TYP3	EXE	Servant called to a different distributed object server using RMI/IIOP	1. Investigate the delay on the other server. The delay may point to session caches. 2. Look for any network problems. 3. Avoid outbound calls.
WAITING TYPE 4	TYP4	BTE	OTS call to RRS. Occurs only in controller when controller is trying to commit a distributed transaction.	1. Investigate the delay on the other server. 2. Look for any network problems. 3. Consider combining application into one server to avoid delay.
WAITING REGIST TO WORKTABLE	WORK	BTE	An indication of contention within the controller while trying to process concurrent requests.	If delay is excessive, consider adding another controller and splitting work off to it.
WAITING OTHER_PRODUCT	OTHER	BTE	Indicates a configuration problem in DNS or TCP/IP	Check to make sure all the DNS servers are running. You might want to look at OPING or ONSLOOKUP.

Table 18. WLM delay monitoring states (continued)

WAITING DISTRIB	DIST	BTE	Controller as a client went outbound waiting for a response.	1. Investigate the delay on the other server. 2. Look for any network problems. 3. Consider combining application into one server to avoid delay.
WAITING SESS_NETWORK	REMT	BTE	Time spent waiting for a TCP/IP session to be established on the network.	The two session delays should be observable in conjunction with TYP3 delays. Look at TCP/IP configuration.
WAITING SESS_SYSPLEX	SYSP	BTE	Time spent waiting for a TCP/IP session to be established on the sysplex.	The two session delays should be observable in conjunction with TYP3 delays. Look at TCP/IP configuration.
WAITING REGULAR_THREAD	REGT	BTE	Waiting for a thread in the controller. Work is bottlenecked in the controller because it is receiving more requests than it can process.	Split the controller.
WAITING SSL_THREAD	SSLT	BTE	Waiting for an SSL thread in the controller. Work is bottlenecked in the controller because it is receiving more requests for SSL handshakes than it can process.	Split controller in increase SSL threads. 1. Increase SSL threads. 2. Look at SSL configuration. 3. Split the controller to increase SSL threads.
WAITING SESS_LOCALMVS	LOCL	BTE	Time spent communicating with a different distributed object server using local optimized communication.	1. Investigate the delay on the other server. 2. Avoid outbound calls.

For information on how to interpret these reports, refer to *z/OS RMF Report Analysis*, SC33-7991 . These specific reports are described under the topic, WLMGL - Workload Activity Report - Goal Mode.

Steps for capturing a workload activity report

The following RMF1 job reads the data from RMF data buffers.

Run the RMF Monitor 1 post processor as a batch job which can read the RMF data buffers from memory and produce a report.

1. Here is a sample job to run the RMF Monitor 1 post processor: (You will need to change the time and date parameters.)

   ```
   //RMF1JOB  JOB 1,CLASS=A
   //RMFPP    EXEC PGM=ERBRMFPP,REGION=0M
   //MFPMSGDS DD   SYSOUT=*
   //SYSIN    DD   *
    SYSOUT(O)
    NOSUMMARY
    SYSRPTS(WLMGL(SCPER(WSHIGH,WSMED,WSLOW,SYSSTC,OPS_DEF)))
    DATE(04172003,04172003)        /* <== SET TO MEAS. DATE  MMDDYYYY  */
    RTOD(1430,1500)                /* <== SET TO MEAS. TIME OF DAY HHMM*/
    DINTV(0002)
   /*
   //
   ```

   This will take the raw SMF data, in this case the MANE data, and produce the workload activity report as an H output class in SDSF.

   ____

2. Using the Workload Activity Reports, observe the measurements in the service and reporting classes. After your measurement run is over, you should reset the SMF parameters to their standard settings with the 'SET SMF=xx' command.

   ____

Collecting additional performance diagnosis information

The following report options are listed here for information. IBM Service may request that you run one or more of these reports while assisting you with diagnosis. You do not need to collect this data unless it is requested by IBM Service.

- If you suspect that you are having throughput problems in a particular address space, for example by looking at some other real-time performance data, IBM Service may need to see a dump of one or more address spaces. This is done using the following parameters:

  ```
   JOBNAME=(<jobname list>)
   SDATA=(LSQA,PSA,SQA,SUM,SWA,TRT,WLM,CSA,RGN)
  ```

- If you suspect that the problem could be resulting from GRS latch or ENQ contention, check the RMF Enqueue Activity Report and enter the console command:

  ```
  D GRS,CONTENTION
  ```

  during a time period in which the performance problem is observed. SYS.BPX.A000.FSLIT.FILESYS.LSN represents HFS latches. Latch sets with a numeric suffix are file latches, specifically SYS.BPX.A000.FSLIT.FILESYS.LSN.01. If you detect file latch contention, the best way to determine the exact HFS file causing the problem is with an SVC dump, also collected during a time period in which contention occurred. You will need to dump one of the OMVS data spaces to get the file information.

```
DUMP COMM=(description of problem)
Reply to dump WTO, where serverproc is the name of your WebSphere Server
                   address space(s)
JOBNAME=(OMVS,Serverproc),DSPNAME=('OMVS'.SYSZBPX1,'OMVS'.SYSZBPX2),
SDATA=(CSA,GRSQ,LPA,NUC,PSA,RGN,SQA,TRT,SUM)
```

- Sometimes USS errors can cause performance problems. The USS Ctrace (SYSOMVS) MIN tracing option always records OMVS errors. You can take an SVC dump of the OMVS address space (as described in the previous bullet) and the data spaces and format the SYSOMVS CTRACE. Use IPCS options 7.2.1, suboption D, component SYSOMVS and the TALLY option (default is FULL). Look for trace events of errors in the TALLY report.
- To find delays in applications, collect application performance information
  - SMF 120 records.
  - Jinsight profile

Chapter 11. Systems Management Facility (SMF) recording and monitoring

This chapter, along with *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838, and *z/OS MVS System Management Facilities (SMF)*, SA22-7630, describes how to enable and use the System Management Facilities (SMF) to collect and record system and job-related information on the WebSphere for z/OS system. This information can be used to bill users, report system reliability, analyze your configuration, schedule work, identify system resource usage, and perform other performance-related tasks that your organization may require.

You can enable SMF recording for:

- **Capacity planning:**

  To determine:

  - How many transactions have run?
  - What is the average and maximum completion time for methods running on each server?
  - How many clients are attached to each server instance? Of these clients, how many are active?
- **Application profiling:**
  - To show an application broken down into its component parts.
  - To provide timing information on the application's component parts.
- **Error reporting:**
  - To detect and record soft failures (those that are generated through an exception or those that are performance-related).
  - To use this error information to trigger an event that will cause an action to occur once a threshold has been reached.

WebSphere for z/OS produces the appropriate SMF records that will allow your installation to perform these functions.

SMF record types

Two types of SMF records can be produced: *activity records* and *interval records*.

Activity records

Gathered as each activity within a server is completed. An activity is a logical unit of business function. It can be a server or user-initiated transaction.

Interval records

Consist of data gathered at installation-specified intervals and provide capacity planning and reliability information.

Six records can be produced: the *Server Activity record*, the *Server Interval record*, the *J2EE Container Activity Record*, the *J2EE Container Interval Record*, the *WebContainer Activity record*, and the *WebContainer Interval record*. Each record is described below. For more information about how to activate these records, see "Steps for using the WebSphere Application Server administrative console" on page 114.

Server activity record

The server activity SMF record is used to record activity that is running inside a WebSphere for z/OS Application Server. This record can be used to perform basic charge-back accounting and to profile your applications to determine, in detail, what is happening inside the WebSphere transaction server.

A single record is created for each activity that is run inside a server or server instance. If the activity runs in multiple servers, then a record is written for each server.

You can activate this record through the administrative console by setting server_SMF_server_activity_enabled=1.

Server interval record

The purpose of the server interval SMF record is to record activity that is running inside a WebSphere for z/OS application server. This record is produced at regular intervals and is an aggregate of the work that ran inside the server instance during the interval.

A single record is created for each server instance that has interval recording active during the interval. If a server has multiple server instances, then a record for each server instance is written and the records must be merged after processing to get a complete view of the work that ran inside the server.

You can activate this record through the administrative console by setting server_SMF_server_interval_enabled=1. You can specify an interval through the administrative console by setting server_SMF_interval_length=n, where n is the desired number of seconds.

J2EE container activity record

The purpose of the J2EE container activity SMF record is to record activity within a J2EE container that is located inside the WebSphere transaction server.

This record can be used to perform basic charge-back accounting, application profiling, problem determination, and capacity planning. A single record is created for each activity that is run within a J2EE container located inside a WebSphere transaction server.

You can activate this record through the administrative console by setting server_SMF_container_activity_enabled=1.

J2EE container interval record

The purpose of the J2EE container interval SMF record is to record activity within a J2EE container that is located inside the WebSphere transaction server.

This record is produced at regular intervals and is an aggregate of the activities running inside a J2EE container during the interval. This record can be used to perform application profiling, problem determination, and capacity planning.

A single record is created for each active J2EE container located in a WebSphere transaction server within the interval being recorded. If there is more than one server instance associated with a server, a record for the container will exist for each server instance. To get a common view of the work running in the J2EE container during the interval, you must merge the records after processing.

You can specify an interval through the WebSphere administrative console by setting server_SMF_interval_length=n, where n is the desired number of seconds.

WebContainer activity record

The purpose of the WebContainer activity SMF record is to record activity within a Web container running inside a WebSphere for z/OS transaction server.

The Web container is deployed within an EJB and runs within the EJB container. The Web container acts as a Web server handling HttpSessions and Servlets. The EJB container is not aware of the work the Web container does. Instead, the EJB container only records that the EJB has been dispatched. Meanwhile, the Web container gathers the detailed information, such as HttpSessions, Servlets, and their respective performance data. A single WebContainer Activity record is created for each activity that is run within a Web container.

WebContainer SMF recording is activated and deactivated along with the activation and deactivation of SMF recording for the J2EE container.

WebContainer interval record

The purpose of the WebContainer interval SMF record is to record activity within a Web container running inside a WebSphere for z/OS transaction server.

The Web container execution environment consists of an EJB that is deployed into the EJB container. The Web container acts as a Web Server handling HttpSessions and Servlets. The EJB container is not aware of the purpose of the WebContainer activity record and only records that the EJB has been dispatched, but does not gather any of the detailed information, such as HttpSessions, Servlets, and their respective performance data. A single WebContainer record is created for each Web container.

In addition to data that is associated with an individual activity, there are some cases of Web container work that are performed outside the scope of an individual request. For example, some instances of http session finalization and http session invalidation are performed asynchronously. In such a case a WebContainer interval record would record this data

WebContainer SMF recording is activated and deactivated along with the activation and deactivation of SMF recording for the J2EE container.

Functions of SMF recording

This section describes what you must do to enable SMF recording, format the output data set, view the output data set, and disable SMF recording for WebSphere Application Server.

Enabling SMF recording

To enable SMF recording for the WebSphere Application Server, there are two steps:

1. Use the WebSphere Application Server administrative console to enable properties for specific record types.
2. Edit the SMFPRMxx parmlib member.

Steps for using the WebSphere Application Server administrative console

Before you begin: Ensure that you have proper access to the administrative console.

How to view or set: Use the WebSphere adminstrative console:

1. Click **Server>Application Servers** in the navigation tree. The Application Servers page appears.

2. Click the **application server name** in the **Name** column of the Application Server collection table. The configuration panel of the application server selected appears.

3. On the configuration panel, under the Additional Properties section, click on **Process Definition**.

4. Click on **control** in the **process Type** column.

5. On the configuration panel, under the Additional Properties section, click on **Environment Entries**.

6. To enable specific record types, specify one or more of the following properties:
   - server_SMF_server_activity_enabled=1
   - server_SMF_server_interval_enabled=1
   - server_SMF_container_activity_enabled=1
   - server_SMF_container_interval_enabled=1
   - server_SMF_interval_length=n, where n is the length in seconds

7. Click OK or Apply on the Server Level Security page.

8. Save the changes and make sure a file sync is performed before restarting the servers.

9. For the changes to take effect, restart the application server.

The steps are completed when the record types are successfully activated.

For additional information on enabling SMF recording for WebSphere Application Server, see (Need a reference here).

Steps for editing the SMFPRMxx parmlib member

Before you begin: Make a working copy of the sample PARMLIB member SMFPRMYL.

Follow these steps to edit the SMFPRMxx parmlib member and enable SMF recording for WebSphere Application Server:

1. Insert an 'ACTIVE' statement to indicate SMF recording. See *z/OS MVS Initialization and Tuning Guide*, SA22-7591.

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 WebSphere Application Server type 120 records only.

   Recommendation: Keep the number of selected record types small to minimize the performance impact.

3. If an interval length was not defined on the server definition through the administrative console, then specify the interval in which you want the interval records created.

4. The server and container interval records will use either:
   - The value specified in the server/container definition as specified in the administrative console
   - The interval specified in the SMF parmlib member (from the SMF product settings) if you specify a length of 0.

You can specify the interval in which you want the Server and Container interval records created in the SMFPRMxx parmlib member (if no interval was specified in administrative console for the server or container definition). The default SMF recording interval is 30 minutes.

Step for writing records to DASD

Before you begin: Make sure you have your modified PARMLIB member SMFPRMxx.

Follow this step to start writing records to DASD:

- Issue the following command:

  ```
  t smf=xx
  ```

  where `xx` is the suffix of the SMF parmlib member (SMFPRMxx).

  Note: See *z/OS MVS System Management Facilities (SMF)*, SA22-7630, for more information.

Writing records to DASD has been completed successfully when the data is recorded in the data set which is specified in SMFPRMxx.

Formatting the output data set

Before you begin: Make sure SMF recording is running.

Perform the following steps to format the SMF recording output data set into a readable format for printing to the screen or other output device:

Note: For detailed information, see *z/OS MVS System Management Facilities (SMF)*, SA22-7630, the SMF Dump program. Below is a short summary of this information.

1. Switch the SMF data sets by entering "`i smf`" from the MVS console to switch the SMF data sets.

2. Run the SMF Dump program (IFASMFDP) to create a sequential data set from the raw dump. A sample JCL is shown in *z/OS MVS System Management Facilities (SMF)*, SA22-7630.

You have successfully formatted the output data set when SMFDUMP ends with return code 0.

Viewing the output data set

The data set should be viewed using a program that can display record type 120.

Steps for using the SMF ViewTool installation and invocation

The Java SMF Record Interpreter is provided in the form of a jar file named `bbomsmfv.jar`. To use it from the z/OS or OS/390 UNIX environment:

1. Verify that the `JAVA_HOME` environment variable refers to the current java installation.

   Example: `JAVA_HOME=../usr/bin/java/J1.3`

   Note: This should be at least Java 1.3 since this release is the first to implicitly contain the necessary record support needed by the interpreter.

2. Copy the file ″bbomsmfv.jar″ to your tools directory.

   Note: Be sure that any edits made to the file in the future are made to both copies of the file, or just execute from the installation directory in the first place.

3. To interpret SMF data from a cataloged z/OS or OS/390 sequential file named ″USER.SMFDATA″ (which was previously created using the IFASMFDP utility as described above), execute:

   ```
   java -cp bbomsmfv.jar com.ibm.ws390.sm.smfview.Interpreter "USER.SMFDATA"
   ```

   Note: It is implicit in the java command parameterization that your current working directory is the tools directory. If this is not the case, you will receive a `NoClassDefFoundError` on com.ibm.ws390.sm.smfview.Interpreter--Java doesn't generate a diagnostic when it doesn't find bbomsmfv.jar in the current directory.

The SMF ViewTool has been successfully installed and invoked when you do not receive any Java error messages after the invocation and the Browser output is shown on the screen.

SMF Browser

The SMF Browser available on the WebSphere for z/OS download site is able to display record type 120. To download the SMF Browser go to: `http://www6.software.ibm.com/dl/websphere20/zosos390-p`

For further information on the SMF Browser, download the browser package and read the associated documentation.

Example of SMF Browser output: This example shows sample output from the SMF Browser. The examples shown feature subtype 7 and subtype 8, in that order.

```
Record#: 14;
 Type: 120; Size: 820; Date: Fri Nov 23 04:54:17 EST 2001;
 SystemID: SY1; SubsystemID: WAS; Flag: 94;
 Subtype: 7 (WEB CONTAINER ACTIVITY);

 # Triplets: 4;
 Triplet #: 1; offset: 76; length: 32; count: 1;
 Triplet #: 2; offset: 108; length: 140; count: 1;
 Triplet #: 3; offset: 264; length: 556; count: 1;

 Triplet #: 1; Type: ProductSection;
   Version: 1; Codeset: Unicode; Endian: 1; TimeStampFormat: 1 (S390STCK64);
   IndexOfThisRecord: 1; Total # records: 1; Total # triplets: 4;

 Triplet #: 2; Type: WebContainerActivitySection;
   HostName          : PLEX1;
   ServerName        : BBOASR4;
   ServerInstanceName: BBOASR4A;
   WlmEnclaveToken         * 00000020 00000242 -------- --------
                               * á... * p1047
   ActivityID              * b6c7a7b7 14e9bc85 000000b0 00000007
                               * æGx╗Z┐e  Á   ⌂  Cp1047
                           * 0926306b -------- -------- --------
                                 *,............ *Cp1047
   ActivityStartTime       * b6c7a7b7 14e9bc85 40404040 40404040 *
   ActivityStopTime        * b6c7a7b7 53a8a645 40404040 40404040 *

 Triplet #: 3; Type: HttpSessionManagerActivitySection;
   # http sessions created: 0; # http sessions invalidated: 0;
   # http sessions active: 0;
Average session life time: 0 [sec*10**-3];

 Triplet #: 4; Type: WebApplicationActivitySection;
   Name: PolicyIVP-localhost_1;

   # Servlets: 1;
   Triplet #: 4.1; offset: 272; length: 284; count: 1;

   Triplet #: 4.1; Type: ServletActivitySection;
     Name: SimpleFileServlet;
     ResponseTime: 48 [sec*10**-3];
     # errors: 0;
     Loaded by this request: 0;
     Loaded since (raw): ea54948e0d;
     Loaded since: Thu Nov 22 10:02:49 EST 2001;
Record#: 72;
  Type: 120; Size: 1744; Date: Fri Nov 23 05:01:02 EST 2001;
  SystemID: SY1; SubsystemID: WAS; Flag: 94;
  Subtype: 8 (WEB CONTAINER INTERVAL);

  # Triplets: 4;
  Triplet #: 1; offset: 76; length: 32; count: 1;
  Triplet #: 2; offset: 108; length: 112; count: 1;
  Triplet #: 3; offset: 264; length: 1480; count: 1;

  Triplet #: 1; Type: ProductSection;
    Version: 1; Codeset: Unicode; Endian: 1; TimeStampFormat: 1 (S390STCK64);
    IndexOfThisRecord: 1; Total # records: 1; Total # triplets: 4;

  Triplet #: 2; Type: WebContainerIntervalSection;
    HostName          : PLEX1;
    ServerName        : BBOASR4;
    ServerInstanceName: BBOASR4A;
    SampleStartTime            * b6c7a6fd 655c0604 40404040 40404040 *
    SampleStopTime             * b6c7a939 9a0e614c 40404040 40404040 *

  Triplet #: 3; Type: HttpSessionManagerIntervalSection;
  http sessions #created: 1; #invalidated: 0;
  http sessions #active: 0; Min #active: 0; Max #active: 0;
  Average session life time: 0;
  Average session invalidate time: 0;
  http sessions #finalized: 0; #tracked: 0;
  http sessions #min live: 0; #max live: 0;

Triplet #: 4; Type: WebApplicationIntervalSection;
  Name: PolicyIVP-localhost_1;

  # Servlets loaded: 0;
  # Servlets: 4;
  Triplet #: 4.1; offset: 312; length: 292; count: 1;
  Triplet #: 4.2; offset: 604; length: 292; count: 1;
  Triplet #: 4.3; offset: 896; length: 292; count: 1;
  Triplet #: 4.4; offset: 1188; length: 292; count: 1;

  Triplet #: 4.1; Type: ServletIntervalSection;
    Name: SimpleFileServlet;
    # requests: 6;
    AverageResponseTime: 764 [sec*10**-3];
    MinimumResponseTime: 18 [sec*10**-3];
    MaximumResponseTime: 4133 [sec*10**-3];
    # errors: 0;
    Loaded since (raw): ea54948e0d;
    Loaded since: Thu Nov 22 10:02:49 EST 2001;

  Triplet #: 4.2; Type: ServletIntervalSection;
    Name: Was40Ivp;
    # requests: 4;
    AverageResponseTime: 4664 [sec*10**-3];
    MinimumResponseTime: 1584 [sec*10**-3];
    MaximumResponseTime: 12572 [sec*10**-3];
    # errors: 0;
    Loaded since (raw): ea58a1509e;
    Loaded since: Fri Nov 23 04:55:14 EST 2001;

  Triplet #: 4.3; Type: ServletIntervalSection;
    Name: /cebit.jsp;
    # requests: 1;
    AverageResponseTime: 204 [sec*10**-3];
    MinimumResponseTime: 204 [sec*10**-3];
    MaximumResponseTime: 204 [sec*10**-3];
    # errors: 0;
    Loaded since (raw): ea58a24a69;
    Loaded since: Fri Nov 23 04:56:18 EST 2001;

  Triplet #: 4.4; Type: ServletIntervalSection;
    Name: JSP 1.1 Processor;
    # requests: 1;
    AverageResponseTime: 482 [sec*10**-3];
    MinimumResponseTime: 482 [sec*10**-3];
    MaximumResponseTime: 482 [sec*10**-3];
    # errors: 0;
    Loaded since (raw): ea54948b66;
    Loaded since: Thu Nov 22 10:02:48 EST 2001;
```

Disabling SMF recording

SMF recording can be enabled for WebSphere Application Server and z/OS and OS/390. Therefore, there are two options when disabling SMF recording:

- Disabling SMF recording for WebSphere.
- Disabling SMF recording for z/OS and OS/390.

Step for disabling SMF recording for WebSphere for z/OS Application Server

Before you begin: Ensure that you have proper access to the administrative console.

Follow the step below to disable SMF recording for WebSphere Application Server through the administrative console.

- Disabling SMF recording can be achieved by the steps outlined in “Steps for using the WebSphere Application Server administrative console” on page 114, and by deselecting the records that have been enabled.
- When you have finished making the administrative console entries, recycle the server.

The step for disabling SMF recording for WebSphere Application Server has been successfully completed when SMF records of records type 120 are no longer being recorded.

Disabling SMF recording for the entire MVS system

Before you begin: Make sure that you have your own working copy of SMFPRMxx and SMF is running.

If you want to disable SMF recording for the whole MVS system follow the step outlined below.

- Edit the SMFPRMxx parmlib member and set SMFPRMxx to `'NOACTIVE'` which will disable the writing of SMF records to DASD.

SMF recording has successfully been disabled for the whole MVS system when SMF records for for both z/OS and OS/390 and WebSphere Application Server are no longer being written to DASD.

SMF record type 120 (78) — WebSphere for z/OS

See “SMF record type 120 (WebSphere for z/OS)” on page 121 for details on WebSphere for z/OS SMF record type 120. Also see *z/OS MVS System Management Facilities (SMF)*, SA22-7630.

SMF record type 80 — WebSphere for z/OS

See “SMF record type 80” on page 144 for details on WebSphere for z/OS SMF record type 80. Also see *z/OS MVS System Management Facilities (SMF)*, SA22-7630.

Appendix A. Auditing in WebSphere

There are two SMF record types you can use to audit in WebSphere: SMF record type 120 and SMF record type 80. This appendix describes both.

SMF record type 120 (WebSphere for z/OS)

Overview of SMF record type 120

This section describes the layout of SMF records as created by WebSphere for z/OS.

Information resulting from the SMF data gathering process is typically presented with the help of an SMF data viewing tool. This record format description is intended to enable your tool providers to design an SMF data viewing tool. Your system administrators will use an SMF data viewing tool with a description presented by your tool provider, since it requires them to make proper selections that limit the amount of presentation data. For example, they might want to view a specific time frame and only specific containers, classes, and methods. They may also occasionally need to refer to the record descriptions.

For additional information about using SMF records, see *z/OS MVS System Management Facilities (SMF)*, SA22-7630.

Record Type 120 (78) - WebSphere for z/OS performance statistics

The following section defines the SMF Record Type 120 (78) - WebSphere for z/OS performance statistics. WebSphere for z/OS writes record type 120 to collect WebSphere for z/OS performance statistics. For more information about SMF record types, see *z/OS MVS System Management Facilities (SMF)*, SA22-7630.

Record Type 120 (78) - WebSphere for z/OS performance statistics

All subtypes of the record type 120 have the following format:
- Standard header section
- Individual header extension for subtype x
- Product section
- Subtype-specific sections listed below.

Record type 120 has the following subtypes:
- **Subtype 1: Server activity record**
  - **Server activity section** (one section per record):
    Contains information about each activity that occurred within one server.
  - **Communication session section** (zero, one, or multiple sections per record):
    Contains information about each communication session.
  - **JVM heap section** (zero, one, or multiple sections per record):
    Contains information about the heap in a server region.
- **Subtype 3: Server interval record**
  - **Server interval section** (one section per record):

  Contains aggregated information about all activities that occurred within the specified server interval.
  - **Server region section** (zero, one, or multiple sections per record):

    Contains information about server regions in the specified interval.
- **Subtype 5: J2EE container activity record**
  - **J2EE container activity section** (one section per record):

    Contains information about each activity that occurred within one J2EE container.
  - **Bean section** (multiple (0..n) sections per record):

    Contains information about all beans involved in this activity.
  - **Bean method section** (multiple (0..n) sections per bean section):

    Contains information about all methods of this bean involved in this activity.
- **Subtype 6: J2EE container interval record**
  - **J2EE container interval section** (one section per record):

    Contains aggregated information about all activities that occurred within one J2EE container in the specified interval.
  - **Bean section** (multiple (0..n) sections per record, see subtype 5):

    Contains information about all beans involved in this activity in the specified interval.
  - **Bean method section** (multiple (0..n) sections per bean section, see subtype 5):

    Contains information about all methods of this class involved in this activity in the specified interval.
- **Subtype 7: WebContainer activity record (Version 2)**
  - **WebContainer activity section** (one section per record):

    Contains information about each activity that occurred within one WebContainer.
  - **HttpSessionManager activity section** (one section per record):

    Contains information about all sessions involved in this activity.
  - **WebApplication section** (multiple (0..n) sections per record):

    Contains information about all WebApplications involved in this activity.
  - **Servlet section** (multiple (0..n) sections per WebApplication section):

    Contains information about all Servlets involved in this activity.
- **Subtype 8: WebContainer interval record (Version 2)**
  - **WebContainer interval section** (one section per record):

    Contains information about each activity that occurred within one WebContainer.
  - **HttpSessionManager section** (one section per record):

    Contains information about all sessions involved in this activity.
  - **WebApplication section** (multiple (0..n) sections per record):

    Contains information about all WebApplications involved in this activity.
  - **Servlet section** (multiple (0..n) sections per WebApplication section):

    Contains information about all Servlets involved in this activity.

Record environment

The following conditions exist for the generation of this record:

- **Record environment**

  Macro SMFWTM (record exit: IEFU83)

Mode Task

Storage Residency
31–bit

Record mapping

This section includes the header/self-defining and product sections.

Header/Self-defining section: This section contains the common SMF record header fields and the triplet fields (offset/length/number), if applicable, that locate the other sections on the record. For a description of triplets, see "Triplets and splitting SMF records" on page 140 and *z/OS MVS System Management Facilities (SMF)*, SA22-7630.

Offset	Offset	Name	Length	Format	Description
0	0	SM120LEN	2	binary	Record length. This field and the next field (total of four bytes) form the RDW (record descriptor word). See "Standard SMF record header" in *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838, for a detailed description.
2	2	SM120SEG	2	binary	Segment descriptor (see record length field)
4	4	SM120FLG	1	binary	**Bit meaning when set** 0: New SMF record format 1: Subtypes used 2: Reserved 3-6: Version indicators* 7: Reserved *See "Standard SMF record header" in *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838, for a detailed description.
5	5	SM120RTY	1	binary	Record type 120(X'78')
6	6	SM120TME	4	binary	Time since midnight, in hundredths of a second, that the record was moved into the SMF buffer.

10	A	SM120DTE	4	packed	Date when the record was moved into the SMF buffer, in the form 0*cyydddF*. See ″Standard SMF record header″ in *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838, for a detailed description.
14	E	SM120SID	4	EBCDIC	System identification (from the SMFPRMxx SID parameter)
18	12	SM120SSI	4	EBCDIC	Subsystem identification from SUBSYS parameter
22	16	SM120RST	2	binary	Record subtype: 1: Server activity 2: Container activity 3: Server interval 4: Container interval. 5: J2EE container activity 6: J2EE container interval 7: WebContainer activity 8: WebContainer interval
24	18	SM120TRN	4	binary	Number of triplets in this record. A triplet is a set of three SMF fields (offset/length/number values) that defines a section of the record. The offset is the offset from the RDW. Subtypes: 1: Value is equal to the number of sessions +2 2 and 4: Value is equal to the number of classes +2.
28	1C	SM120PRS	4	binary	Offset to product section from RDW.
32	20	SM120PRL	4	binary	Length of product section.
36	24	SM120PRN	4	binary	Number of product sections.

Individual header extension for subtype 1					
40	28	SM120SAS	4	binary	Offset to server activity section from RDW

44	2C	SM120SAL	4	binary	Length of server activity section
48	30	SM120SAN	4	binary	Number of server activity sections
52	34	SM120CSS	4	binary	Offset to communication session section from RDW
56	38	SM120CSL	4	binary	Length of communication session section
60	3C	SM120CSN	4	binary	Number of communication session sections
64	40	SM120JHS	4	binary	Offset to JVM heap section from RDW
68	44	SM120JHL	4	binary	Length of JVM heap section
72	48	SM120JHN	4	binary	Number of jvm heap sections

Individual header extension for subtype 3					
40	28	SM120SIS	4	binary	Offset to server interval section from RDW
44	2C	SM120SIL	4	binary	Length of server interval section
48	30	SM120SIN	4	binary	Number of server interval sections
52	34	SM120SRS	4	binary	Offset to server region section from RDW
56	38	SM120SRL	4	binary	Length of server region section
60	3C	SM120SRN	4	binary	Number of server region sections

Individual header extension for subtype 5					
40	28	SM120JA1	4	binary	Offset to J2EE container activity section from RDW
44	2C	SM120JA2	4	binary	Length of J2EE container activity section
48	30	SM120JA3	4	binary	Number of J2EE container activity sections

The following triplet appears 0–n times; once for each bean section.					
52	34	SM120JAS	4	binary	Offset to bean section from RDW
56	38	SM120JAL	4	binary	Length of bean section
60	3C	SM120JAN	4	binary	Number of bean sections

Individual header extension for subtype 6					
40	28	SM120JI1	4	binary	Offset to J2EE container interval section from RDW
44	2C	SM120JI2	4	binary	Length of J2EE container interval section

48	30	SM120JI3	4	binary	Number of J2EE container interval sections
The following triplet appears 0–n times; once for each bean section.					
52	34	SM120JIS	4	binary	Offset to bean section from RDW
56	38	SM120JIL	4	binary	Length of bean section
60	3C	SM120JIN	4	binary	Number of bean sections
Individual header extension for subtype 7					
40	28	SM120WA1	4	binary	Offset to WebContainer activity section from RDW.
44	2C	SM120WA2	4	binary	Length of WebContainer activity section.
48	30	SM120WA3	4	binary	Number of WebContainer activity sections.
52	34	SM120WA4	4	binary	Offset to HttpSessionManager activity section from RDW.
56	38	SM120WA5	4	binary	Length of HttpSessionManager activity section.
60	3C	SM120WA6	4	binary	Number of HttpSessionManager activity sections.
The following triplet appears 0–n times, once for each WebApplication section.					
64	40	SM120WA7	4	binary	Offset to WebApplication section from RDW.
68	44	SM120WA8	4	binary	Length of WebApplication section.
72	48	SM120WA9	4	binary	Number of WebApplication sections.
Individual header extension for subtype 8					
40	28	SM120WI1	4	binary	Offset to WebContainer interval section from RDW.
44	2C	SM120WI2	4	binary	Length of WebContainer interval section.
48	30	SM120WI3	4	binary	Number of WebContainer interval sections.
52	34	SM120WI4	4	binary	Offset to HttpSessionManager interval section from RDW.
56	38	SM120WI5	4	binary	Length of HttpSessionManager interval section.
60	3C	SM120WI6	4	binary	Number of HttpSessionManager interval sections.
The following triplet appears 0–n times, once for each WebApplication section.					

64	40	SM120WI7	4	binary	Offset to WebApplication section from RDW.
68	44	SM120WI8	4	binary	Length of WebApplication section.
72	48	SM120WI9	4	binary	Number of WebApplication sections.

Product section:

Offset	Offset	Name	Length	Format	Description
0	0	SM120MFV	4	binary	CB SMF version
4	4	SM120COD	8	EBCDIC	Character codeset in which strings in the SMF record are encoded
12	C	SM120END	4	binary	Encode of numbers in the SMF record
16	10	SM120TSF	4	binary	Encoding of timestamps: 1: S390STCK64: The time values are encoded in 64-bit S/390 Store Clock format.

Reassembly information.					
20	14	SM120IXR	4	binary	Index of this record
24	18	SM120NRC	4	binary	Total number of records
28	1C	SM120NTR	4	binary	Total number of triplets

Subtype 1: Server activity record:

1. **Server activity section** (one section per record):
   Contains information about each activity that occurred within one server.
2. **Communications session section** (zero, one, or multiple sections per record):
   Contains information about each communication session.
3. **JVM heap section** (zero, one, or multiple sections per record):
   Contains information about the heap in each server region.

Subtype 1: Server activity section

Offset	Offset	Name	Length	Format	Description
0	0	SM120HNM	64	EBCDIC	WebSphere for z/OS transaction server host name
64	40	SM120SNM	8	EBCDIC	WebSphere for z/OS transaction server name
72	48	SM120SIN	8	EBCDIC	WebSphere for z/OS transaction server instance name

80	50	SM120SNM	4	binary	Total number of server regions that were involved to process this activity. If applicable, up to the first five server region address space IDs are listed within the next five fields.
84	54	SM120SR1	4	binary	The specific WebSphere for z/OS transaction server instance server region where the request ran
88	58	SM120SR2	4	binary	The specific WebSphere for z/OS transaction server instance server region where the request ran
92	5C	SM120SR3	4	binary	The specific WebSphere for z/OS transaction server instance server region where the request ran
96	60	SM120SR4	4	binary	The specific WebSphere for z/OS transaction server instance server region where the request ran
100	64	SM120SR5	4	binary	The specific WebSphere for z/OS transaction server instance server region where the request ran
104	68	SM120CRE	8	EBCDIC	The user credentials under which the activity began.
112	70	SM120ATY	4	binary	Type of activity that this record references: 1: Method request: This record refers to a method request that is not part of a global transaction. 2: Transaction: This record refers to a transaction.
116	74	SM120AID	20	HEX	Identity of the activity
136	88	SM120WLM	8	HEX	WLM enclave token
144	90	SM120AST	16	S390STCK	Activity start time
160	A0	SM120AET	16	S390STCK	Activity stop time
176	B0	SM120NIM	4	binary	Number of input methods
180	B4	SM120NGT	4	binary	Number of global transactions that were started in the server region
184	B8	SM120NLT	4	binary	Number of local transactions that were started in the server region
188	BC	SM120J2E	4	binary	J2EE server
192	C0	SM120CEL	8	EBCDIC	WebSphere for z/OS cell name

200	C8	SM120NOD	8	EBCDIC	WebSphere for z/OS node name
208	D0	SM120WCP	8	binary	Total CPU time accumulated by the WLM enclave

Subtype 1: Communications session section

Offset	Offset	Name	Length	Format	Description
0	0	SM120CSH	8	HEX	Communications session handle
8	8	SM120CSA	64	EBCDIC	Communications session address
72	48	SM120CSO	4	binary	Communications session optimization 1: Local communications session: The session is a local OS/390 optimized communications session. 2: Remote communications session: The session is a remote communications session. 3: Remote encrypted (SSL) 4: Remote within sysplex. 5: HTTP session. 6: HTTP encrypted session.
76	4C	SM120SDR	4	binary	Data received; the number of bytes received by the server
80	50	SM120SDT	4	binary	Data transferred; the number of bytes transferred from the server back to the client.

Subtype 3: Server interval record:

1. **Server interval section** (one section per record):

   Contains information about each activity that occurred within the specified server interval.
2. **Server region section** (zero, one, or multiple sections per record):

   Contains information about each server region in the specified server interval.

Subtype 3: Server interval section

Offset	Offset	Name	Length	Format	Description
0	0	SM120HNM	64	EBCDIC	WebSphere for z/OS transaction server host name

64	40	SM120SNM	8	EBCDIC	WebSphere for z/OS transaction server name
72	48	SM120SIN	8	EBCDIC	WebSphere for z/OS transaction server instance name
80	50	SM120SST	16	S390STCK	Time that the sample began in the server
96	60	SM120SET	16	S390STCK	Time that the sample ended
112	70	SM120NGT	4	binary	Number of global transactions that have run through the server instance during the interval that have been initiated by the server instance during the interval
116	74	SM120NLT	4	binary	Number of local transactions that have been initiated by the server instance during the interval
120	78	SM120NCS	4	binary	Number of communications sessions that exist at the end of the interval
124	7C	SM120NCA	4	binary	The number of communications sessions that have been active during the interval
128	80	SM120NLS	4	binary	Number of local communication sessions that exist at the end of the interval
132	84	SM120NLA	4	binary	Number of active local communication sessions that have been attached and active within the server instance during the interval
136	88	SM120NRS	4	binary	Number of remote communication sessions that exist at the end of the interval
140	8C	SM120NRA	4	binary	Number of active remote communication sessions that have been attached and active within the server instance during the interval
144	90	SM120BTS	4	binary	Number of bytes that have been transferred to the server from all attached clients
148	94	SM120BFS	4	binary	Number of bytes that have been sent from the server to all attached clients

152	98	SM120BTL	4	binary	Number of bytes that have been transferred to the server from all locally attached clients
156	9C	SM120BFL	4	binary	Number of bytes that have been transferred from the server to all locally attached clients
160	A0	SM120BTR	4	binary	Number of bytes that have been transferred to the server from all remotely attached clients
164	A4	SM120BFR	4	binary	Number of bytes that have been transferred from the server to all remotely attached clients
168	A8	SM120J2E	4	binary	J2EE server.
172	AC	SM120CEL	8	EBCDIC	WebSphere for z/OS transaction server cell name
180	B4	SM120NOD	8	EBCDIC	WebSphere for z/OS transaction server node name
188	BC	SM120NHS	4	binary	Number of HTTP communication sessions that exist at the end of the interval
192	C0	SM120NHA	4	binary	Number of HTTP communication sessions that have been attached and active within the server instance during the interval
196	C4	SM120BTH	4	binary	Number of bytes that have been transferred to the server from all HTTP attached clients
200	C8	SM120BFH	4	binary	Number of bytes that have been transferred from the server to all HTTP attached clients
204	CC	SM120TEC	8	binary	Total CPU time accumulated by the WLM enclaves

Subtype 5: J2EE container activity record (Version 2):

1. **J2EE container activity section** (one section per record):

   Contains information about each activity that occurred within one J2EE container.
2. **Bean section** (multiple sections per record):

   Contains information about all beans involved in this activity.
3. **Bean method section** (multiple sections per bean section):

   Contains information about all methods of beans involved in this activity.

Subtype 5: J2EE container activity section

Offset	Offset	Name	Length	Format	Description
0	0	SM120JA4	64	EBCDIC	WebSphere for z/OS transaction server host name
64	40	SM120JA5	8	EBCDIC	WebSphere for z/OS transaction server name
72	48	SM120JA6	8	EBCDIC	WebSphere for z/OS transaction server instance name
80	50	SM120JA7	4	binary	The specific WebSphere for z/OS transaction server instance server region where the request ran
84	54	SM120JA8	512	Unicode	WebSphere for z/OS container name. **Note:** This is hardcoded to ″Default″ for the 4.0.1 timeframe.
596	254	SM120JA9	8	HEX	The WLM enclave token
604	25C	SM120JAA	4	binary	RESERVED
608	260	SM120JAB	20	HEX	The identity of the activity
628	274	SM120CEL	8	EBCDIC	Cell
636	27C	SM120NOD	8	EBCDIC	Node

Subtype 5: Bean section

Offset	Offset	Name	Length	Format	Description
0	0	SM120JB1	512	Unicode	AMCName of the bean activated by the container. **Note:** If the length of the AMCName exceeds 256 DBCS characters (512 bytes), the rightmost 256 characters are recorded.
512	200	SM120JB2	60	binary	UUID based AMC name
572	23C	SM120JB3	4	binary	The bean’s type. 0: CMP entity bean. 1: BMP entity bean. 2: Stateless session bean. 3: Stateful session bean.
576	240	SM120JB4	4	binary	RESERVED
580	244	SM120JB5	4	binary	RESERVED
584	248	SM120JB6	4	binary	RESERVED
588	24C	SM120JB7	4	binary	The bean’s reentrance policy. 0: Not reentrant within transaction. 1: Reentrant within transaction.
592	250	SM120JB8	4	binary	RESERVED

596	254	SM120JMC	4	binary	RESERVED
600	258	SM120JM6	4	binary	RESERVED
604	25C	SM120JB9	4	binary	Number of method triplets in this bean section
The following triplet appears 0–n times; once for each bean method section.					
608	260	SM120JBS	4	binary	Offset to bean method section from the beginning of this bean section
612	264	SM120JBL	4	binary	Length of bean method section
616	268	SM120JBN	4	binary	Number of bean method sections

Subtype 5: Bean method section

Offset	Offset	Name	Length	Format	Description
0	0	SM120JM1	1,024	Unicode	The name of the method including its signature in its externalized, human-readable form. If the length of the method exceeds 512 DBCS characters (1024 bytes), the leftmost 512 characters are recorded.
1024	400	SM120JM2	4	binary	The number of times the method was invoked during the activity.
1028	404	SM120JM3	4	binary	Average response time. The response time is measured in milliseconds (the granularity provided by the JVM - hopefully, it will be equal to 0 in most cases).
1032	408	SM120JM4	4	binary	Maximum response time. The response time is measured in milliseconds.

1036	40C	SM120JM5	4	binary	The bean method's transaction policy. Values from com.ibm.WebSphere for z/OS.csi. TransactionAttribute.java:
					`0: "TX_NOT_SUPPORTED"`
					`1: "TX_BEAN_MANAGED"`
					`2: "TX_REQUIRED"`
					`3: "TX_SUPPORTS"`
					`4: "TX_REQUIRES_NEW"`
					`5: "TX_MANDATORY"`
					`6: "TX_NEVER"`

1040	410	SM120JM8	4	binary	RESERVED.
1044	414	SM120JM9	4	binary	RESERVED.

1048	418	SM120JMA	512	Unicode	List of ejbRoles associated with the method. Separator character: ″;″ (semicolon). If the length of the concatenated string exceeds 256 characters (512 bytes), only its leftmost 256 characters are recorded.
1560	618	SM120JMB	4	binary	RESERVED.
1564	61C	SM120JMD	4	binary	RESERVED.
1568	620	SM120JME	4	binary	ejbLoad: # of invocations
1572	624	SM120JMF	4	binary	ejbLoad: avg execution time
1576	628	SM120JMG	4	binary	ejbLoad: max execution time
1580	62C	SM120JMH	4	binary	ejbStore: # of invocations
1584	630	SM120JMI	4	binary	ejbStore: avg execution time
1588	634	SM120JMJ	4	binary	ejbStore: max execution time
1592	638	SM120JMK	4	binary	ejbActivate: # of invocations
1596	63C	SM120JML	4	binary	ejbActivate: avg execution time
1600	640	SM120JMM	4	binary	ejbActivate: max execution time
1604	644	SM120JMN	4	binary	ejbPassivate: # of invocations
1608	648	SM120JMO	4	binary	ejbPassivate: avg execution time
1612	64C	SM120JMP	4	binary	ejbPassivate: max execution time
1616	650	SM120JMQ	8	binary	RESERVED
1624	658	SM120JMR	8	binary	RESERVED
1632	660	SM120JMS	8	binary	RESERVED

Subtype 6: Container interval record (Version 2):

1. **J2EE container interval section** (one section per record):

   Contains information about each activity that occurred within one J2EE container in the specified interval.
2. **Bean section** (multiple sections per record):

   Contains information about all beans involved in this activity in the specified interval.
3. **Bean method section** (multiple sections per bean section):

   Contains information about all methods of all beans involved in this activity in the specified interval.

Subtype 6: J2EE container interval section

Offset	Offset	Name	Length	Format	Description

0	0	SM120JI4	64	EBCDIC	The WebSphere for z/OS transaction server host name.
64	40	SM120JI5	8	EBCDIC	The WebSphere for z/OS transaction server name.
72	48	SM120JI6	8	EBCDIC	The WebSphere for z/OS transaction server instance name.
80	50	SM120JI7	512	Unicode	The WebSphere for z/OS container name. **Note:** This is hardcoded to ″Default″ for the 4.0.1 timeframe.
592	250	SM120JI8	16	S390STCK	The time that the sample began in the server.
608	260	SM120JI9	16	S390STCK	The time that the sample ended.
624	270	SM120CEL	8	EBCDIC	Cell
632	278	SM120NOD	8	EBCDIC	Node

Subtype 6: Bean section: (See Subtype 5: Bean section)

Subtype 6: Bean method section: (See Subtype 5: Bean method section)

Subtype 7: WebContainer activity record (Version 2):

1. **WebContainer activity section** (one section per record):

   Contains information about each activity that occurred within one web container.
2. **HttpSessionManager section** (one section per record):

   Contains information about all (there may be zero or one) http sessions associated to one single activity.
3. **WebApplication section** (multiple (0–n) sections per record):

   Contains information about all WebApplications involved in this activity.
4. **Servlet activity section** (multiple (0–n) sections per WebApplication section):

   Contains information about each servlet associated with WebApplications involved in this activity.

Subtype 7: WebContainer activity section

Offset	Offset	Name	Length	Format	Description
0	0	SM120WAA	64	EBCDIC	The WebSphere transaction server host name.
64	40	SM120WAB	8	EBCDIC	The WebSphere transaction server name.
72	48	SM120WAC	8	EBCDIC	The WebSphere transaction server instance name.
80	50	SM120WAD	8	HEX	The WLM enclave token.
88	58	SM120WAE	20	HEX	The identity of the activity.
108	6C	SM120WAF	16	S390STCK	The time the activity began in the server.

124	7C	SM120WAG	16	S390STCK	The time the activity ended.
140	8C	SM120CEL	8	EBCDIC	Cell
148	94	SM120NOD	8	EBCDIC	Node

Subtype 7: HttpSessionManager activity section

Offset	Offset	Name	Length	Format	Description
0	0	SM120WAH	4	binary	″created Sessions″: Number of http sessions that were created.
4	4	SM120WAI	4	binary	″invalidatedSessions″: Number of http session that were invalidated.
8	8	SM120WAJ	4	binary	″activeSessions″: Number of http sessions that were referenced during this activity.
12	C	SM120WAK	4	binary	″sessionLifeTime″: lifetime of the session in milliseconds. If ″invalidatedSessions″ > 0, this is the average lifetime (in milliseconds) of the invalidated http session.

Subtype 7: WebApplication activity section

Offset	Offset	Name	Length	Format	Description
0	0	SM120WAL	256	Unicode	The name of the WebApplication.
256	100	SM120WAM	4	binary	Number of servlet triplets in this web application section.
The following triplet appears 0–n times, once for each servlet section.					
260	104	SM120WAN	4	binary	Offset to servlet section from the beginning of this WebApplication section.
264	108	SM120WAO	4	binary	Length of servlet section.
268	10C	SM120WAP	4	binary	Number of servlet sections.

Subtype 7: Servlet activity section

Offset	Offset	Name	Length	Format	Description
0	0	SM120WAQ	256	Unicode	The name of the servlet.
256	100	SM120WAR	4	binary	″responseTime″: Response time in milliseconds.
260	104	SM120WAS	4	binary	″numErrors″: The number of errors that were encountered during the servlet execution.

264	108	SM120WAT	4	binary	″loaded″: 0: The servlet did not have to be loaded as a result of this request. 1: The servlet had to be loaded as the result of this request.
268	10C	SM120WAU	16	EBCDIC	″loadedSince″: Timestamp from System.currentTimeMillis() when the servlet was loaded, in HEX format. **Sample:**The data as it appears in the record has the format `e7ef7c577c`, which needs to be converted to a Java long: `996155348860`. The Java long digits can be converted to java.util.Date: `Thu Jul 26 15:49:08 GMT+02:00 2001`
284	11C	SM120CPU	8	binary	RESERVED

Subtype 8: WebContainer interval record (Version 2):

1. **WebContainer interval section** (one section per record):

   Contains information about each activity that occurred within one WebContainer record.
2. **HttpSessionManager section** (one section per record):

   Contains information about each http session involved in this activity.
3. **WebApplication section** (multiple(0..n) sections per record):

   Contains information about all WebApplications involved in this activity.
4. **Servlet section** (multiple(0..n) sections per WebApplication section):

   Contains information about all servlets involved per WebApplication in this activity.

Subtype 8: WebContainer interval section

Offset	Offset	Name	Length	Format	Description
0	0	SM120WIA	64	EBCDIC	The WebSphere transaction server host name.
64	40	SM120WIB	8	EBCDIC	The WebSphere transaction server name.
72	48	SM120WIC	8	EBCDIC	The WebSphere transaction server instance name.
80	50	SM120WID	16	S390STCK	The time the sample began.
96	60	SM120WIE	16	S390STCK	The time the sample ended.
112	70	SM120CEL	8	EBCDIC	Cell
120	78	SM120NOD	8	EBCDIC	Node

Subtype 8: HttpSessionManager interval section

Offset	Offset	Name	Length	Format	Description
0	0	SM120WIF	4	binary	″createdSessions″: Number of http sessions that were created.
4	4	SM120WIG	4	binary	″invalidatedSessions″: Number of http sessions that were invalidated.
8	8	SM120WIH	4	binary	″activeSessions″: Current number of http sessions that are actively referenced in the server at the end of the interval.
12	C	SM120WII	4	binary	″minActiveSessions″: Minimum number of active http sessions during the interval..
16	10	SM120WIJ	4	binary	″maxActiveSessions″: Maximum number of active http sessions during the interval.
20	14	SM120WIK	4	binary	″sessionLifeTime″: Average lifetime (in milliseconds) of invalidated http sessions.
24	18	SM120WIL	4	binary	″sessionInvalidateTime″: Average time (in milliseconds) that was required to process the invalidation of http sessions.
28	1C	SM120WIM	4	binary	″finalizedSessions″: Number of sessions that were finalized.
32	20	SM120WIN	4	binary	″liveSessions″: Total number of http sessions being tracked by the server at the end of the interval. This includes both active and inactive sessions.
36	24	SM120WIO	4	binary	″minLiveSessions″: Minimum number of live http sessions during the interval.
40	28	SM120WIP	4	binary	″maxLiveSessions″: Maximum number of live http sessions during the interval.

Subtype 8: WebApplication interval section

Offset	Offset	Name	Length	Format	Description
0	0	SM120WIQ	256	Unicode	The WebApplication name.

256	100	SM120WIR	4	binary	″numLoadedServlets″: Number of servlets that were loaded. **Note:** This value might differ from the number of servlet sections in this record since servlets might exist that have been inactive during the interval.
260	104	SM120WIS	4	binary	Number of servlet triplets in this web application section.
The following triplet appears 0–n times, once for each servlet section.					
264	108	SM120WIT	4	binary	Offset to servlet section from the beginning of this WebApplication section.
268	10C	SM120WIU	4	binary	Length of the servlet section.
272	110	SM120WIV	4	binary	Number of servlet section.

Subtype 8: Servlet interval section

Offset	Offset	Name	Length	Format	Description
0	0	SM120WIW	256	Unicode	The servlet name.
256	100	SM120WIX	4	binary	″totalRequests″: Number of times the servlet service was requested during the interval.
260	104	SM120WIY	4	binary	″responseTime″: Average response time in milliseconds.
264	108	SM120WIZ	4	binary	″minResponseTime″: Minimum response time in milliseconds.
268	10C	SM120WJ1	4	binary	″maxResponseTime″: Maximum response time in milliseconds.
272	110	SM120WJ2	4	binary	″numErrors″: The number of errors that were encountered during servlet execution.
276	114	SM120WJ3	16	EBCDIC	″loadedSince″: Timestamp when the servlet was loaded. Sample: `Fri May 25 08:42:25 EDT 2001`
292	124	SM120WJ4	8	binary	RESERVED
300	12C	SM120WJ5	8	binary	RESERVED
308	134	SM120WJ6	8	binary	RESERVED

Triplets and splitting SMF records

Triplets

You can use triplets to build self-describing SMF records that contain various types of data sections and a varying number of each of these sections. All data sections are described by triplets that consist of:

1. An offset that specifies the start position of the data
2. A length that describes the length of the section
3. A count that describes how many instances of the section are included in this record.

The two triplets that describe the product section and the general record information section (for example, the section describing the container itself in a container activity record) are located at fixed positions within the record. This allows one to start evaluating the record right after having evaluated the record header.

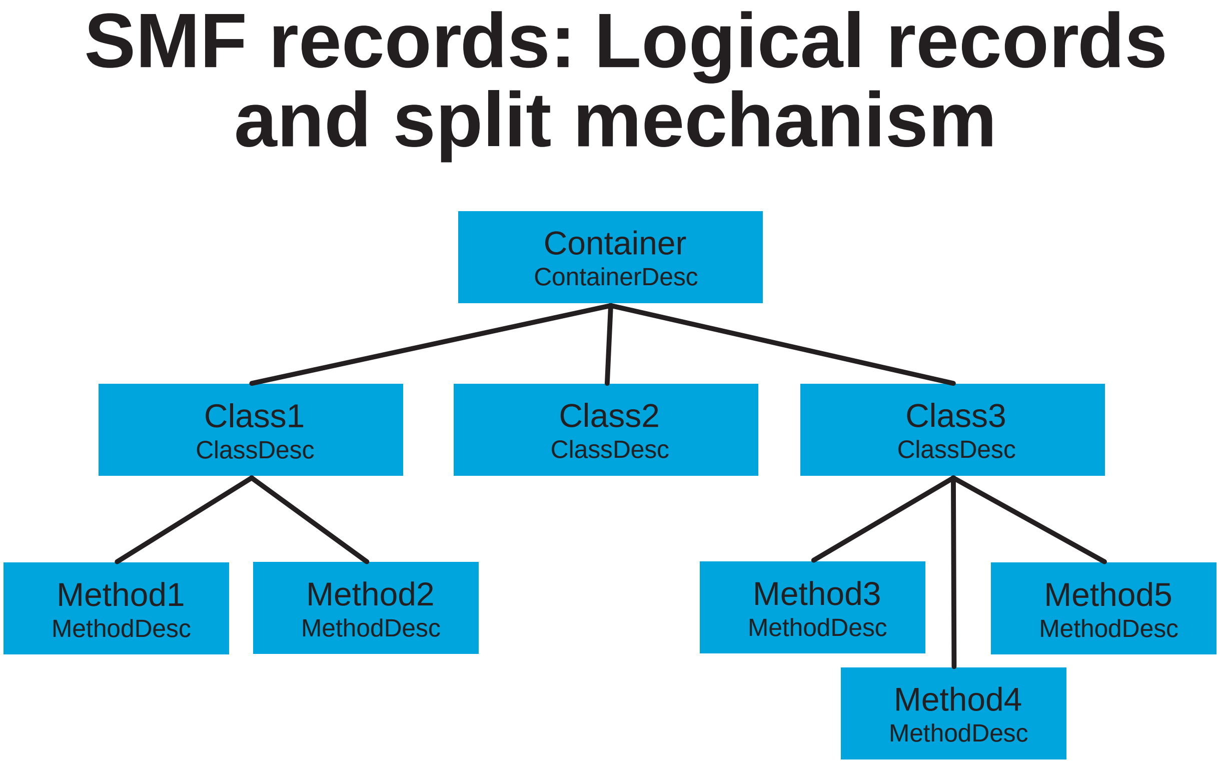
Splitting SMF records

Since most of the WebSphere Application Server SMF records are used to describe variable-length data structures (for example, there might be hundreds of classes by container and hundreds of methods by class), the SMF records may be larger than the maximum record size supported by SMF (32KB). In this case, the logical records need to be split into several physical records.

Each of those physical records needs to be self-describing and self-contained. *Self-describing* indicates what we described in the paragraph on triplets before; it is a purely mechanical structure to help read a record. *Self-contained* indicates that, even if we have only a subset of the physical records at hand that together describe the original logical record, we need to be able to evaluate these records, combine the information stored in them, and set an ’incomplete’ flag. This is required since, as we break up a logical record into physical records and write them to SMF one after the other, SMF might decide that only the first few physical records fit into the primary SMF dump dataset whereas the remaining physical records are written into an alternate SMF dump dataset. At the time when a formatted SMF dump dataset is evaluated, we may not assume that all physical records that make up one logical record are present. For example, self-containedness of a physical container activity record means that it contains the description of the container, but not necessarily all of its classes.

We use a similar splitting mechanism like the one that is currently used in the RMF product. Note that,in the case of container records (subtype 2 and 4), we cannot assume that records will be split at a class boundary, but we must consider the case when the methods that belong to one class also need to be split over multiple physical records, as shown in the diagrams below.

Note: The section length numbers used throughout the following diagrams are only for demonstrative purposes. In particular, the arrows indicating 32K boundaries or the total length of the records are placed at random. You can fit many more classes and methods into a physical record than suggested by the diagrams.



SMF records: Logical records and split mechanism

Container
ContainerDesc

Class1
ClassDesc

Class2
ClassDesc

Class3
ClassDesc

Method1
MethodDesc

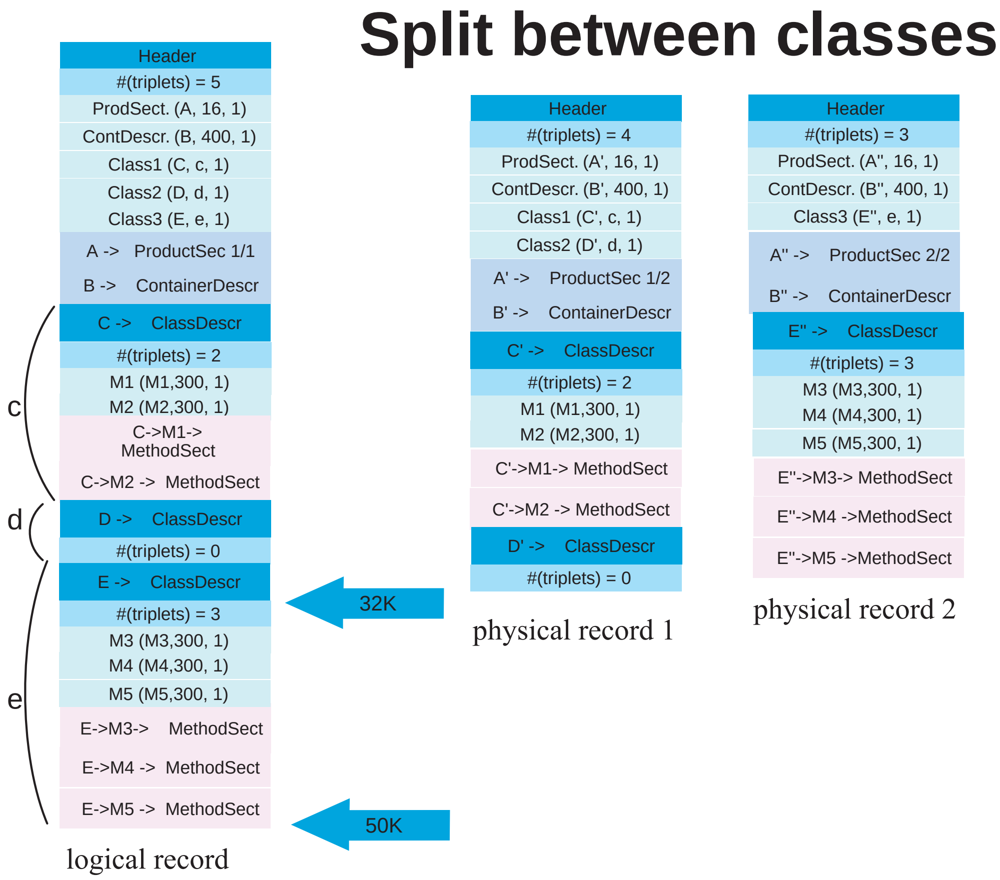
Method2
MethodDesc

Method3
MethodDesc

Method4
MethodDesc

Method5
MethodDesc

Figure 3. SMF records: logical records and split mechanism



Split between classes

Header
#(triplets) = 5
ProdSect. (A, 16, 1)
ContDescr. (B, 400, 1)
Class1 (C, c, 1)
Class2 (D, d, 1)
Class3 (E, e, 1)
A -> ProductSec 1/1
B -> ContainerDescr
C -> ClassDescr
#(triplets) = 2
M1 (M1,300, 1)
M2 (M2,300, 1)
C->M1-> MethodSect
C->M2 -> MethodSect
D -> ClassDescr
#(triplets) = 0
E -> ClassDescr
#(triplets) = 3
M3 (M3,300, 1)
M4 (M4,300, 1)
M5 (M5,300, 1)
E->M3-> MethodSect
E->M4 -> MethodSect
E->M5 -> MethodSect

logical record

Header
#(triplets) = 4
ProdSect. (A', 16, 1)
ContDescr. (B', 400, 1)
Class1 (C', c, 1)
Class2 (D', d, 1)
A' -> ProductSec 1/2
B' -> ContainerDescr
C' -> ClassDescr
#(triplets) = 2
M1 (M1,300, 1)
M2 (M2,300, 1)
C'->M1-> MethodSect
C'->M2 -> MethodSect
D' -> ClassDescr
#(triplets) = 0

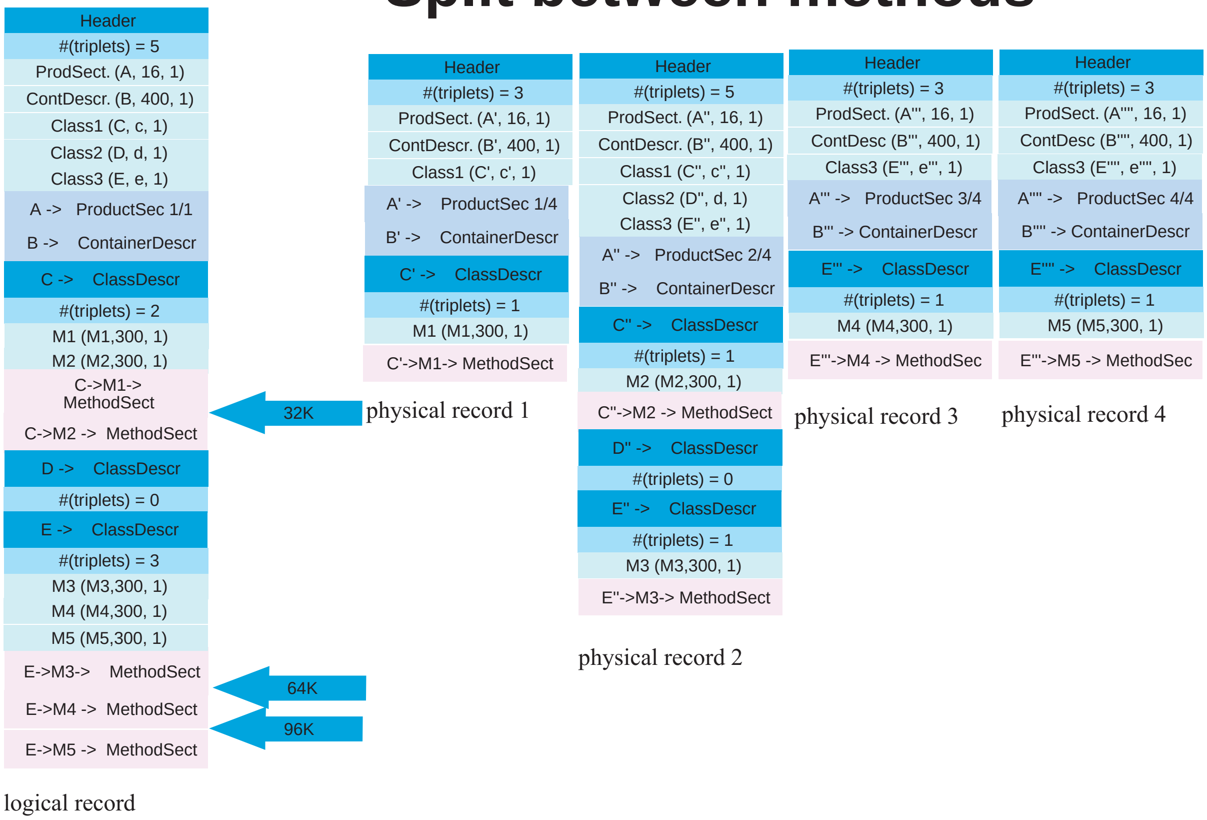
physical record 1

Header
#(triplets) = 3
ProdSect. (A'', 16, 1)
ContDescr. (B'', 400, 1)
Class3 (E'', e, 1)
A'' -> ProductSec 2/2
B'' -> ContainerDescr
E'' -> ClassDescr
#(triplets) = 3
M3 (M3,300, 1)
M4 (M4,300, 1)
M5 (M5,300, 1)
E''->M3-> MethodSect
E''->M4 ->MethodSect
E''->M5 ->MethodSect

physical record 2

Figure 4. SMF record: split between classes

Split between methods



Header
#(triplets) = 5
ProdSect. (A, 16, 1)
ContDescr. (B, 400, 1)
Class1 (C, c, 1)
Class2 (D, d, 1)
Class3 (E, e, 1)
A -> ProductSec 1/1
B -> ContainerDescr
C -> ClassDescr
#(triplets) = 2
M1 (M1,300, 1)
M2 (M2,300, 1)
C->M1-> MethodSect
C->M2 -> MethodSect
D -> ClassDescr
#(triplets) = 0
E -> ClassDescr
#(triplets) = 3
M3 (M3,300, 1)
M4 (M4,300, 1)
M5 (M5,300, 1)
E->M3-> MethodSect
E->M4 -> MethodSect
E->M5 -> MethodSect

logical record

Header
#(triplets) = 3
ProdSect. (A', 16, 1)
ContDescr. (B', 400, 1)
Class1 (C', c', 1)
A' -> ProductSec 1/4
B' -> ContainerDescr
C' -> ClassDescr
#(triplets) = 1
M1 (M1,300, 1)
C'->M1-> MethodSect

physical record 1

Header
#(triplets) = 5
ProdSect. (A'', 16, 1)
ContDescr. (B'', 400, 1)
Class1 (C'', c'', 1)
Class2 (D'', d, 1)
Class3 (E'', e'', 1)
A'' -> ProductSec 2/4
B'' -> ContainerDescr
C'' -> ClassDescr
#(triplets) = 1
M2 (M2,300, 1)
C''->M2 -> MethodSect
D'' -> ClassDescr
#(triplets) = 0
E'' -> ClassDescr
#(triplets) = 1
M3 (M3,300, 1)
E''->M3-> MethodSect

physical record 2

Header
#(triplets) = 3
ProdSect. (A''', 16, 1)
ContDesc (B''', 400, 1)
Class3 (E''', e''', 1)
A''' -> ProductSec 3/4
B''' -> ContainerDescr
E''' -> ClassDescr
#(triplets) = 1
M4 (M4,300, 1)
E'''->M4 -> MethodSec

physical record 3

Header
#(triplets) = 3
ProdSect. (A'''', 16, 1)
ContDesc (B'''', 400, 1)
Class3 (E'''', e'''', 1)
A'''' -> ProductSec 4/4
B'''' -> ContainerDescr
E'''' -> ClassDescr
#(triplets) = 1
M5 (M5,300, 1)
E'''->M5 -> MethodSec

physical record 4

Figure 5. SMF record: split between methods

SMF record type 80

Overview of SMF record type 80

As WebSphere becomes more capable of authentication and setting or changing the identity on a thread, so arises the need for the ability to audit these changes. Along with this also comes the need to audit the accompanying authorization requests made through EJBRoles checking, intending to produce audit records that include the original authenticated identity. This auditing in WebSphere is managed not through WebSphere itself, but through its External Security Manager (RACF or equivalent), where the SMF records are cut.

Preparing audit support

In order to take advantage of auditing in WebSphere, you need to set up SMF and RACF and have both running.

Setting up SMF for audit support

For information on setting up and starting SMF, see *z/OS MVS System Management Facilities (SMF)*, SA22-7630.

Setting up RACF for audit support

To set up RACF for auditing in WebSphere, enable auditing for the EJB Roles by including the `AUDIT` attribute.

Example: You can turn on auditing for the ADMIN and PAYROLL classes with the following command:

```
RALTER EJBROLE (ADMIN,PAYROLL) AUDIT(ALL)
```

Notes:

1. Alternately, you could modify the RACFROLE job to put the AUDIT information there.
2. For more information and additional parameters for the AUDIT attribute, see *z/OS Security Server RACF Auditor's Guide*, SA22-7684.

See *z/OS Security Server RACF Auditor's Guide*, SA22-7684 for more information.

Using audit support

Auditing is performed using SMF records issued by RACF or an equivalent External Security Manager. This means that SMF audit records are cut as part of the WebSphere use of SAF interfaces such as IRRSIA00 (to manage ACEEs) and the RACROUTE macro.

The table below lists the various security authentication mechanisms and the corresponding data that is written to each part of the ACEE X500NAME field (this data is also in the RACO and SMF records). The information under ″Service Name″ is the constant string that is included in the ″Issuer's Distinguished Name″ field of X500NAME. The information under ″Authenticated Identity″ is the principal that is recorded in the ″Subject's Distinguished Name″ field.

Table 19. Security authentication mechanisms and the corresponding data that is written to each part of the ACEE X500NAME field

Authentication mechanism	Service name	Authenticated identity
Custom Registry	WebSphere Custom Registry	Custom registry principal name

Table 19. Security authentication mechanisms and the corresponding data that is written to each part of the ACEE X500NAME field (continued)

Kerberos	WebSphere Kerberos	Kerberos principal, in the ″DCE″ format used for extracting the corresponding MVS userid using IRRSIM00 (/.../realm/principal)
RunAs Rolename	WebSphere Role Name	Role name
RunAs Server	WebSphere Server Credential	MVS userid
Trust Interceptor	WebSphere Authorized Login	MVS userid
RunAs Userid/Password	WebSphere Userid/Password	MVS Userid

In addition to tracking by MVS userid, events need to be traced to an originating userid. This is especially true for originating userids that are not MVS-based, such as EJB Roles, Kerberos principals, and Custom Registry principals.

Note: See *z/OS MVS System Management Facilities (SMF)*, SA22-7630 and *z/OS Security Server RACF Auditor's Guide*, SA22-7684 for more information on auditing in WebSphere with the SMF record type 80.

Appendix B. Naming conventions for application servers

Reviewers Note:
No changes have been made to this chapter. I will need to review it with someone who understands this for V5.

This chapter contains guidelines for establishing a naming convention for your application servers.

Understanding the need for application server naming conventions

There are a number of reasons why you need to establish a naming convention for application servers:

1. **Because WebSphere Application Server servers are like IMS or CICS regions.**
   - They contain tailored procedures for the control and server regions.
   - They contain tailored environmental variables for each instance of a server.
   - They contain environmental variables for each instance of a server.
   - Servers may be self-contained or dependent on other servers.
2. **For security.**
   - Regions have user IDs associated with them.
   - Users are allowed access to servers and objects within.
3. **For Workload Manager (WLM).**
   - Classification of regions and work within the regions
   - Application environments.

A WebSphere for z/OS application server consists of a number of address spaces which require the installation to manage configuration files, security profiles, workload classification constructs, and so forth. To create, manage, and recognize application servers, a template is needed for stamping out servers and server instances. The template needs to apply to the following:

- **Server names:**
  - Control region PROC names
  - Server region PROC names
  - Application Environment names
  - Instance names
- **Security:**
  - User/group/uid/gid
  - Control regions
  - Server regions
  - Instance names
- **Procedures:**
  - Environmental files
  - Library names
- **Other:**
  - DB2 collection and package names

– Log stream names

Here is a system for creating servers based on a 4–character application naming scheme, which we refer to as XXXX. Since multiple instances of a server may exist on one or more systems in the WebSphere for z/OS environment, there is also a requirement to distinguish between servers. You can use a system that looks like the following:

Note: Everything is determined by 4 characters: XXXX (and Y).

CBserver name	= CBXXXX
- APPLENV name	= CBXXXX
CBserver instance name	= CBXXXXAY
- ORBsrvname default value	= CBXXXXAY
Userid for control region	= CBXXXXC
- PROC for control region	= CBXXXXC
Group id for control region	= CBXXXXG
Userid for server region	= CBXXXXS
- PROC for server region	= CBXXXXS
Group id for server region	= CBXXXX
Default remote userid	= CBXXXXI
Default local userid	= CBXXXXD
Group id for default ids	= CBXXXXP

Here are the user IDs. Change as desired.

CBXXXXC 0 - do not change.
CBXXXXS 1100
CBXXXXD 1101
CBXXXXI 1102

Here are the groups/GIDS. Change as desired.

CBXXXXG 1000
CBXXXXR 1001
CBXXXXP 1002

The naming convention is also applied to:

Server-specific log streams	= CBXXXX.ERROR.LOG
LRMs	= CBXXXX_LRM_DB2
LRMIs	= CBXXXXAY_LRMI_DB2
DB2 collections	= CBXXXX_PK
HFS File system names	= /WSCapps/CBXXXX/bin and /WSapps/CBXXXX/lib
OS File names	= hlq.CBXXXX.LOADLIB, hlq.CBXXXX.HFS, and hlq.CBXXXXAY.PARMS

and so forth.

The part of the naming scheme which breaks down is the management of the UID/GID associated with RACF identities. There appears to be no easy mechanism to automate the assignment or association of these entities with userids.

For example, below you will see one way of defining the procs for the control and server regions associated with application server APP1. Notice that each server instance has its own unique data set containing environmental settings. You could easily change this scheme so that there is one PDS for the entire sysplex specifying different members. The important limitation to remember is that there is minimal capability to pass symbolic parameter overrides to the server regions.

Also notice that data set names indicate whether the data set is unique to the server or common across the sysplex. In our naming scheme, the second level qualifier indicates whether the data set is to be used:

- sysplex -wide
- only for servers running on a specific system
- server-wide
- only for a given server instance.

Control Region Proc:

```
//BBOASR1 PROC SRVNAME='BBOASR1A',
//       PARMS='',
//       CBCONFIG='/WebSphere390/CB390'
//* See instructions at the bottom of this file
//  SET BBOLIB='BBO'
//  SET LELIB='CEE'
//  SET DB2='DB2'
//  SET RELPATH='controlinfo/envfile'
//BBOASR1 EXEC PGM=BBOCTL,REGION=0M,
// PARM='/ -ORBsrvname &SRVNAME &PARMS'
//*STEPLIB  DD DSN=&BBOLIB..SBBOLD2,DISP=SHR
//*         DD DSN=&BBOLIB..SBBOLOAD,DISP=SHR
//*         DD DSN=&LELIB..SCEERUN,DISP=SHR
//*         DD DSN=&DB2..SDSNLOAD,DISP=SHR
//BBOENV    DD PATH='&CBCONFIG/&RELPATH/&SYSPLEX/&SRVNAME/current.env'
//CEEDUMP   DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//SYSOUT    DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//SYSPRINT  DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
```

Server Region Proc:

```
//BBOASR1S PROC IWMSSNM='BBOASR1A',PARMS='-ORBsrvname ',
//      CBCONFIG='/WebSphere390/CB390'
//* See instructions at the bottom of this file
//  SET BBOLIB='BBO'
//  SET LELIB='CEE'
//  SET DB2='DB2'
//  SET RELPATH='controlinfo/envfile'
//BBOASR1S EXEC PGM=BBOSR,REGION=0M,TIME=NOLIMIT,
// PARM='/ &PARMS &IWMSSNM'
//STEPLIB  DD DSN=&BBOLIB..SBBOULIB,DISP=SHR
//*         DD DSN=&BBOLIB..SBBOLD2,DISP=SHR
//*         DD DSN=&BBOLIB..SBBOLOAD,DISP=SHR
//*         DD DSN=&LELIB..SCEERUN,DISP=SHR
//*         DD DSN=&DB2..SDSNLOAD,DISP=SHR
//BBOENV   DD PATH='&CBCONFIG/&RELPATH/&SYSPLEX/&IWMSSNM/current.env'
//CEEDUMP  DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
```

Naming conventions for application servers

Appendix C. 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:

CICS
DB2
IBM
IMS
IMS/ESA
MVS
OS/390
RACF
RMF
S/390
VTAM
WebSphere
z/OS

Lotus, Notes, Domino, and Lotus Go Webserver, are trademarks of the Lotus Development Corporation in the United States, other countries, or both.

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

Microsoft, ActiveX, Visual Basic, Visual C++, Windows, Windows NT, and the Windows logo are trademarks or registered trademarks of Microsoft Corporation in the United States, other countries, or both.

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

Program Number: 5655–I35

Printed in the United States of America

SA22-7912-00