

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



Operations and Administration

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



Operations and Administration

Note

Before using this information and the product it supports, be sure to read the general information under Appendix D, "Notices" on page 189.

Sixth edition (September 2002)

This is a major revision of SA22-7839-04

This edition applies to WebSphere Application Server V4.0.1 for z/OS and OS/390 (5655-F31), and to all subsequent releases and modifications until otherwise indicated in new editions.

The most current versions of the WebSphere Application Server V4.0.1 for z/OS and OS/390 publications are at this Web site:

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

© Copyright International Business Machines Corporation 2000, 2002. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	vii
Tables	ix
About this book	xi
Who should read this book	xi
How this book is organized	xi
Where to find related information, tools, and supplements	xii
How to send your comments	xiv
Summary of changes	xv
Chapter 1. Introduction	1
Overview of WebSphere for z/OS operations	2
Overview of WebSphere for z/OS administration	2
Overview of required WebSphere for z/OS elements and subsystems	2
WebSphere for z/OS elements	2
Required z/OS or OS/390 subsystems	3
Chapter 2. Identifying where to perform WebSphere for z/OS operations	5
Operating WebSphere for z/OS from the Systems Management User Interface	9
Operating WebSphere for z/OS from the MVS console	9
Chapter 3. Operating WebSphere for z/OS	11
Start operations	12
Starting the WebSphere for z/OS environment	12
Starting servers and server instances	15
Cold starting WebSphere for z/OS	16
Hot starting WebSphere for z/OS	16
Warm starting WebSphere for z/OS	16
Quick starting WebSphere for z/OS	16
Restart operations	17
Peer restart and recovery	17
Stop operations	29
Stop vs. cancel	29
Stopping the WebSphere for z/OS environment	29
Stopping servers and server instances	30
Cancel operations	30
Canceling application servers and server instances	30
Canceling the Daemon	30
Display operations	31
Displaying the status of ARM-registered address spaces including WebSphere for z/OS servers and server instances	31
Displaying active address spaces	32
Displaying active replies	32
Displaying units of work (threads) for DB2	32
Displaying indoubt units of work (threads) for DB2	33
Displaying units of work for CICS	33
Displaying units of work (transactions) for IMS	33
Displaying units of work for RRS	34
Modify operations	34
Getting help for the modify command	34
Canceling application servers and server instances with the modify command	34
Modifying trace variables	34
Displaying control region and trace settings status	36
Other WebSphere for z/OS operations	37
ARM and restart	37
Setting up error log streams for different servers and server instances	38
Using Workload Management for WebSphere for z/OS operations	38
Chapter 4. Operational considerations for z/OS or OS/390 subsystems	43
Operational considerations for z/OS or OS/390 subsystems	43
DB2 Universal Database for z/OS and OS/390 operations	43
IMS operations	44
RRS operations	44
Workload Management (WLM) operations	45
Chapter 5. WebSphere for z/OS backup guidelines and procedures	47

Guidelines for backup of the WebSphere for z/OS system 47

Chapter 6. Monitoring and recovering WebSphere for z/OS and dependent subsystems. 49

Startup order for WebSphere for z/OS and dependent subsystems 49
Automation and recovery scenarios and guidelines. 50 |
 APPC automation and recovery scenarios 50 |
 WLM automation and recovery scenarios 51 |
 RACF automation and recovery scenarios 52 |
 RRS automation and recovery scenarios. 53 |
 Unix System Services automation and recovery scenarios 54 |
 TCP/IP automation and recovery scenarios 55 |
 DB2 automation and recovery scenarios. 55 |
 CICS automation and recovery scenarios 56 |
 IMS automation and recovery scenarios. 57 |
 LDAP automation and recovery scenarios 59 |
 NFS automation and recovery scenarios. 60 |
 WebSphere for z/OS (Daemon) automation and recovery scenarios 61 |
 Naming automation and recovery scenarios 62 |
 Interface Repository automation and recovery scenarios 62 |
 Systems Management (SM) automation and recovery scenarios 63 |
 WebServer automation and recovery scenarios 64

Chapter 7. WebSphere for z/OS administration procedures 65

Logging messages and trace data for Java server applications. 65 |
 Determining where to issue the messages 65 |
 System performance when logging messages and trace data 66 |
 Issuing application messages to the MVS master console 66

Chapter 8. Security administration. 69

Adding a new administrator for the Administration application 69
 Steps for updating the access control list for LDAP. 69
 Step for granting the new administrator database authorities 71

Chapter 9. WebSphere for z/OS performance tuning guidelines 73

Introduction 73
Recommended hardware configuration . . . 74
Organization of tuning 74
Tuning the operating system 75
 Storage. 75
 Tuning tips for z/OS or OS/390 78
 Tuning tips for the system logger 80
 Tuning tips for LE 82
 z/OS UNIX System Services 84
 Tuning tips for WLM 85
Tuning for subsystems 87
 Tuning tips for DB2 87
 Tuning tips for RACF. 90
 Tuning tips for TCP/IP 91
 Tuning tips for WebSphere MQ and JMS 92
 Tuning tips for GRS 99
 Tuning tips for JVM 99
 Tuning tips for LDAP 101
 Tuning tips for CICS. 101
Tuning the WebSphere for z/OS runtime . . 102
 Server configuration example 103
 Locations of executable programs . . . 106
 Security 107
 Servlet / EJB integrated runtime. . . . 108
Tuning for J2EE applications 108
 Topology. 108
 J2EE container and applications 111
 J2EE application programming tips . . . 117
 SOAP tuning tips 118

Chapter 10. WebSphere for z/OS performance monitoring procedures . . 121

Performance information and accounting 121
Steps for diagnosing a performance problem 121
Collecting performance diagnosis information 125
 Steps for capturing RMF data. 126
 Steps for capturing a workload activity report. 129
Collecting additional performance diagnosis information 130

Chapter 11. Systems Management Facility (SMF) recording and monitoring 133

SMF record types 134
 Server activity record 134
 Container activity record 134
 Server interval record 135

Container interval record	135		Overview of SMF record type 80.	181
J2EE container activity record	135		Preparing audit support	181
J2EE container interval record.	136		Using audit support	181
WebContainer activity record	136			
WebContainer interval record	136			
Functions of SMF recording	137			
Enabling SMF recording	137			
Formatting the output data set	141			
Viewing the output data set	141			
Disabling SMF recording	147			
SMF record type 120 (78) — WebSphere for z/OS	147			
SMF record type 80 — WebSphere for z/OS	148			
Appendix A. Auditing in WebSphere.	149			
SMF record type 120 (WebSphere for z/OS)	149			
Overview of SMF record type 120	149			
Record Type 120 (78) - WebSphere for z/OS performance statistics	149			
Triplets and splitting SMF records	176			
SMF record type 80	181			
			Appendix B. MOFW tuning considerations	183
			Appendix C. Naming conventions for application servers.	185
			Understanding the need for application server naming conventions.	185
			Appendix D. Notices	189
			Examples in this book	191
			Programming Interface information	191
			Trademarks	191
			Glossary	193
			Index	195

Figures

1.	Organization of servers in a recoverable communication manager setup	19	5.	Setting the isolation policy	110
2.	Disabling the Server Region Recycle Function	78	6.	SMF records: logical records and split mechanism	178
3.	Setting heapsizes in the Administration application.	101	7.	SMF record: split between classes	179
4.	Having applications on a single system avoids calls from one system to another	109	8.	SMF record: split between methods	180

Tables

1. WebSphere for z/OS operations tasks	5	13. NFS automation and recovery scenarios	60
2. Behavior of automatic restart management regarding WebSphere for z/OS server instances	32	14. WebSphere for z/OS automation and recovery scenarios	61
3. APPC automation and recovery scenarios	50	15. Naming automation and recovery scenarios	62
4. Workload Manager (WLM) automation and recovery scenarios	51	16. Interface Repository (IR) automation and recovery scenarios	62
5. RACF automation and recovery scenarios	52	17. Systems Management (SM) automation and recovery scenarios	63
6. RRS automation and recovery scenarios	53	18. WebServer (Servlet) automation and recovery scenarios	64
7. UNIX System Services automation and recovery scenarios	54	19. Recommended default setting for LOGR	80
8. TCP/IP automation and recovery scenarios	55	20. Connection management policy setting effect on bean connection reclamation	114
9. DB2 automation and recovery scenarios	55	21. Security authentication mechanisms and the corresponding data that is written to each part of the ACEE X500NAME field	182
10. CICS automation and recovery scenarios	56		
11. IMS automation and recovery scenarios	57		
12. LDAP automation and recovery scenarios	59		

About this book

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

Note: The full product name is "WebSphere Application Server V4.0.1 for z/OS and OS/390," 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 *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838, which describes the Administration and Operations applications that manage WebSphere for z/OS. 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 Systems Management GUI 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 43 describes helpful hints for when customizing z/OS or OS/390 subsystems.
- Chapter 5, "WebSphere for z/OS backup guidelines and procedures" on page 47 describes WebSphere for z/OS backup guidelines and procedures.
- Chapter 6, "Monitoring and recovering WebSphere for z/OS and dependent subsystems" on page 49 describes guidelines for monitoring and recovering WebSphere for z/OS and its dependent subsystems.
- Chapter 7, "WebSphere for z/OS administration procedures" on page 65 describes WebSphere for z/OS administration tasks.

- Chapter 8, “Security administration” on page 69 describes security issues within WebSphere for z/OS.
- Chapter 9, “WebSphere for z/OS performance tuning guidelines” on page 73 describes WebSphere for z/OS performance tuning guidelines.
- Chapter 10, “WebSphere for z/OS performance monitoring procedures” on page 121 describes WebSphere for z/OS performance monitoring guidelines.
- Chapter 11, “Systems Management Facility (SMF) recording and monitoring” on page 133 describes Systems Management Facility (SMF) recording and monitoring for WebSphere for z/OS.
- “SMF record type 120 (WebSphere for z/OS)” on page 149 describes Systems Management Facility (SMF) record type 120 for WebSphere for z/OS.
- Appendix B, “MOFW tuning considerations” on page 183 describes tuning considerations specific to MOFW objects.
- Appendix C, “Naming conventions for application servers” on page 185 describes how to establish a naming convention for your application servers.
- Appendix D, “Notices” on page 189 provides notices about programming interfaces, examples used in this book, and trademarks.

Where to find related information, tools, and supplements

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

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

- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Program Directory*, GI10-0680, describes the elements of and the installation instructions for WebSphere for z/OS.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: License Information*, LA22-7855, describes the license information for WebSphere for z/OS.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834, describes the planning, installation, and customization tasks and guidelines for WebSphere for z/OS.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Messages and Diagnosis*, GA22-7837, provides diagnosis information and describes messages and codes associated with WebSphere for z/OS.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Operations and Administration*, SA22-7835, describes system operations and administration tasks.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836, describes how to develop, assemble, and install J2EE applications in a WebSphere for z/OS J2EE server.

- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling CORBA Applications*, SA22-7848, describes how to develop, assemble, and deploy CORBA applications in a WebSphere for z/OS (MOFW) server.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838, describes the system administration and operations tasks as provided in the Systems Management User Interface.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management Scripting API*, SA22-7839, describes the functionality of the WebSphere for z/OS Systems Management Scripting API product.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Migration*, GA22-7860, describes migration procedures for WebSphere for z/OS.

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

- *WebSphere Application Server for OS/390 V3.5 Standard Edition Planning, Installing, and Using*, GC34-4835, provides information about running the Version 3.5 runtime shipped with the V4.0.1 product within the HTTP Server address space. You can use this configuration if you want to continue running non-J2EE-compliant Web applications in the V3.5 runtime within the HTTP Server address space while migrating to the full WebSphere for z/OS run time.
- *Building Business Solutions with WebSphere*, SC09-4432

The integrated WebSphere Application Server Advanced Edition and WebSphere Application Server Enterprise Edition InfoCenter includes CORBA (MOFW) information you need to code CORBA (MOFW) components. Go to: <http://www.ibm.com/software/webservers/appserv/infocenter.html>

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

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

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

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

How to send your comments

Your feedback is important in helping to provide the most accurate and highest quality information. You can e-mail your comments to:

`wasdoc@us.ibm.com`

or fax them to 919-254-0206.

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

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

Summary of changes

Summary of changes
for SA22-7835-05
WebSphere for z/OS V4.0.1
as updated, September 2002
PTFs UQ90051, UQ90052, and UQ70037
APARs PQ65206, PO65207, and PQ66463
service level W401400

This book contains information previously presented in SA22-7835-04, which supports WebSphere for z/OS. The following is a summary of changes to this information:

The following sections were added or changed:

- Modify Operations command added to Table 1 on page 5.
- Modify Operations command added to “Modify operations” on page 34.
- “Prerequisites” on page 17 was added to the Peer Restart and Recovery information.
- “New work” on page 21 was added to the Peer Restart and Recovery information.
- Chapter 9, “WebSphere for z/OS performance tuning guidelines” on page 73 was completely restructured and updated.
- Chapter 10, “WebSphere for z/OS performance monitoring procedures” on page 121 was completely restructured and updated.
- “Steps for using a DB2 SPUFI script to turn SMF recording on and off” on page 139 describes a DB2 SPUFI script for turning SMF recording on and off.
- Appendix A, “Auditing in WebSphere” on page 149 was renamed from “SMF record type 120 (WebSphere for z/OS)” to “Auditing in WebSphere” and the previous contents were made a section within. Added to the newly-reorganized appendix was another section “SMF record type 80” on page 181 that describes the new function with an overview and sections on preparing and using the audit support.

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

**Summary of changes
for SA22-7835-04**

**WebSphere for z/OS V4.0.1
as updated, July 2002
service level W401082**

This book contains information previously presented in SA22–7835–03, which supports WebSphere for z/OS. The following is a summary of changes to this information:

The following sections were added or changed:

- “Peer restart and recovery” on page 17 (due to APAR PQ57396)
- “Steps for checking and starting the workload management application environment” on page 40
- “TCP/IP automation and recovery scenarios” on page 55
- “Tuning tips for z/OS or OS/390” on page 78
- “Tuning tips for DB2” on page 87
- “Real storage” on page 75
- “Garbage collection and JVM_HEAPSIZE” on page 76
- “Server region recycling” on page 77(due to APAR PQ57888)
- “Performance information and accounting” on page 121
- “One tran or multiple trans?” on page 109
- “SMF record types” on page 134(due to APAR PQ59911)
- “WebContainer activity record” on page 136(due to APAR PQ59911)
- “WebContainer interval record” on page 136(due to APAR PQ59911)
- “Functions of SMF recording” on page 137(due to APAR PQ59911)
- “Subtype 7: WebContainer activity record (Version1)” on page 151(due to APAR PQ59911)
- “Subtype 7: WebContainer interval record (Version1)” on page 151(due to APAR PQ59911)
- “Header/Self-defining section” on page 152(due to APAR PQ59911)
- “Subtype 7: WebContainer activity record (Version 1)” on page 171(due to APAR PQ59911)
- “Subtype 8: WebContainer interval record (Version 1)” on page 173(due to APAR PQ59911)

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

**Summary of changes
for SA22–7835–03**

**WebSphere for z/OS V4.0.1
as updated, March 2002
service level W401038**

This book contains information previously presented in SA22-7835-02, which supports WebSphere for z/OS. The following is a summary of changes to this information:

The following sections were added or changed:

- “IMS operations” on page 44 (due to APARs PQ55873, PQ53749, and PQ52868)
- “Startup order for WebSphere for z/OS and dependent subsystems” on page 49 (due to APARs PQ55873, PQ53749, and PQ52868)
- “RRS automation and recovery scenarios” on page 53 (due to APARs PQ55873, PQ53749, and PQ52868)
- “CICS automation and recovery scenarios” on page 56 (due to APARs PQ55873, PQ53749, and PQ52868)
- “IMS automation and recovery scenarios” on page 57 (due to APARs PQ55873, PQ53749, and PQ52868)
- “Introduction” on page 73
- “Tuning tips for z/OS or OS/390” on page 78
- “Tuning tips for the system logger” on page 80
- “Tuning tips for LE” on page 82
- “Tuning tips for WLM” on page 85
- “Tuning tips for DB2” on page 87
- “Tuning tips for RACF” on page 90
- “Tuning tips for TCP/IP” on page 91
- “Tuning tips for WebSphere MQ and JMS” on page 92
- “Tuning tips for GRS” on page 99
- “Tuning tips for JVM” on page 99
- “Server configuration example” on page 103
- “Virtual storage” on page 75
- “Real storage” on page 75
- “Fine tuning the LE Heap” on page 82
- “Garbage collection and JVM_HEAPSIZE” on page 76
- “Server region recycling” on page 77 (due to APAR PQ57888)
- “EJBs” on page 111
- “HTTP transport” on page 108
- “One tran or multiple trans?” on page 109

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

**Summary of changes
for SA22-7835-02
WebSphere for z/OS V4.0.1
as updated, October 2001**

This book contains information previously presented in SA22-7835-01, which supports WebSphere for z/OS. The following is a summary of changes to this information:

The following sections were added or changed:

- Table 1 on page 5
- “Operating WebSphere for z/OS from the Systems Management User Interface” on page 9
- “Operating WebSphere for z/OS from the MVS console” on page 9
- Chapter 3, “Operating WebSphere for z/OS” on page 11. The whole chapter was reorganized and many individual sections were modified—revision bars appear throughout.
- “DB2 Universal Database for z/OS and OS/390 operations” on page 43
- “APPC automation and recovery scenarios” on page 50
- “WLM automation and recovery scenarios” on page 51
- “RACF automation and recovery scenarios” on page 52
- “RRS automation and recovery scenarios” on page 53
- “Unix System Services automation and recovery scenarios” on page 54
- “DB2 automation and recovery scenarios” on page 55
- “LDAP automation and recovery scenarios” on page 59
- “NFS automation and recovery scenarios” on page 60
- “WebSphere for z/OS (Daemon) automation and recovery scenarios” on page 61
- “Naming automation and recovery scenarios” on page 62
- “Interface Repository automation and recovery scenarios” on page 62
- “Systems Management (SM) automation and recovery scenarios” on page 63
- “WebServer automation and recovery scenarios” on page 64
- Chapter 8, “Security administration” on page 69. This is a new chapter of security material, some of which appeared previously in other parts of the book.
- Chapter 10, “WebSphere for z/OS performance monitoring procedures” on page 121. The whole chapter was reorganized and many individual sections were modified—revision bars appear throughout.

- “Tuning tips for WebSphere MQ and JMS” on page 92
- “Tuning tips for JVM” on page 99
- “Security” on page 107
- Chapter 11, “Systems Management Facility (SMF) recording and monitoring” on page 133
- “SMF record types” on page 134
- “J2EE container activity record” on page 135
- “J2EE container interval record” on page 136
- “Record Type 120 (78) - WebSphere for z/OS performance statistics” on page 149
- “Header/Self-defining section” on page 152
- “Subtype 5: J2EE container activity record (Version 1)” on page 166
- “Subtype 6: Container interval record (Version 1)” on page 170
- Appendix B, “MOFW tuning considerations” on page 183. This is a new appendix containing the MOFW tuning information that was previously in Chapter 10, “WebSphere for z/OS performance monitoring procedures” on page 121.

**Summary of changes
for SA22-7835-01
WebSphere for z/OS
as updated, June 2001
service level W400018**

This book contains information previously presented in SA22-7835-00, which supports WebSphere for z/OS. The following is a summary of changes to this information:

- Chapter 2, “Identifying where to perform WebSphere for z/OS operations” on page 5
- Chapter 10, “WebSphere for z/OS performance monitoring procedures” on page 121

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 servers and server instances using the Systems Management User Interface that runs on Windows. You can display the status of all server instances, stop application servers and server instances, cancel application servers and server instances, cancel and restart servers and server instances, and filter the operations window. For information on how to use the WebSphere for z/OS operations application, see the *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface, SA22-7838*.

This book, *WebSphere Application Server V4.0.1 for z/OS and OS/390: Operations and Administration, SA22-7835*, 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

The WebSphere for z/OS administration application allows you to use the Systems Management User Interface on Windows 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 administration application, see the *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface, SA22-7838*. Additional administration tasks and guidelines are in Chapter 7, “WebSphere for z/OS administration procedures” on page 65 and Chapter 10, “WebSphere for z/OS performance monitoring procedures” on page 121.

Overview of required WebSphere for z/OS elements and subsystems

WebSphere for z/OS elements

The following are the required elements for a WebSphere for z/OS host system:

- WebSphere for z/OS System Server Instances:
 - Daemon
 - System Management Server (SMS)
 - Naming
 - Interface Repository (IR)

- WebSphere for z/OS Applications Server Instances:
 - Control Region (CR)
 - Server Region (SR)

The following are optional elements that can run on other WebSphere Application Server hosts (either S/390 or distributed):

- WebSphere Application Server Clients

See *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834 for more information.

Required z/OS or OS/390 subsystems

See Chapter 6, “Monitoring and recovering WebSphere for z/OS and dependent subsystems” on page 49 for information regarding the order in which subsystems must be started and stopped, and how to recover your system when a subsystem fails.

For information about the required z/OS or OS/390 subsystems, see *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834.

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 Systems Management User Interface (SMEUI) on Windows (some operations)
- TSO or RRS panels (some operations)

For more information on the Systems Management User Interface, see *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838.

Table 1. WebSphere for z/OS operations tasks

Task	MVS console	SMEUI	TSO panel	Reference to associated procedure
Start operations				
Starting the WebSphere for z/OS environment / Daemon	Yes	No	No	See "Starting the WebSphere for z/OS environment" on page 12.
Determining if the Daemon is up	Yes	Yes	No	See "Starting servers and server instances" on page 15.
Starting a server or application server instance	Yes	Application server only	No	See "Starting servers and server instances" on page 15 and <i>WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface</i> , SA22-7838.
Cold starting WebSphere for z/OS	Yes	No	No	See "Cold starting WebSphere for z/OS" on page 16.
Hot starting WebSphere for z/OS	Yes	Application server only	No	See "Hot starting WebSphere for z/OS" on page 16.
Warm starting WebSphere for z/OS	Yes	Application server only	No	See "Warm starting WebSphere for z/OS" on page 16.

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

Task	MVS console	SMEUI	TSO panel	Reference to associated procedure
Quick starting WebSphere for z/OS	Yes	Application server only	No	See “Quick starting WebSphere for z/OS” on page 16.
Stop operations				
Stopping the WebSphere for z/OS environment / Daemon	Yes	No	No	See “Stopping the WebSphere for z/OS environment” on page 29.
Stopping a server or application server instance	Yes	Application server only	No	See “Stopping servers and server instances” on page 30 and <i>WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface</i> , SA22-7838.
Cancel operations				
Canceling the Daemon	Yes	No	No	See “Canceling application servers and server instances” on page 30.
Canceling a server or application server instance	Yes	Yes	No	See “Canceling application servers and server instances” on page 30.
Display operations				
Displaying the status of ARM-registered address spaces including WebSphere for z/OS servers and server instances	Yes	No	No	See “Displaying the status of ARM-registered address spaces including WebSphere for z/OS servers and server instances” on page 31.
Displaying units of work (threads) for DB2	Yes	No	No	See “Displaying units of work (threads) for DB2” on page 32.
Displaying indoubt units of work (threads) for DB2	Yes	No	No	See “Displaying indoubt units of work (threads) for DB2” on page 33.

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

Task	MVS console	SMEUI	TSO panel	Reference to associated procedure
Displaying units of work for RRS	No	No	Yes	See “Displaying units of work for RRS” on page 34. See <i>z/OS MVS Programming: Resource Recovery</i> , SA22-7616, 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 33. See <i>CICS Operations and Utilities Guide</i> , 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 33. Also see <i>IMS/ESA Summary of Operator Commands</i> , SC26-8766.
Displaying the status of a server or server instance	Yes	Yes	No	See Chapter 3, “Operating WebSphere for z/OS” on page 11 and <i>WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface</i> , SA22-7838.
Displaying active address spaces	Yes	No	No	See “Displaying active address spaces” on page 32.
Displaying active replies	Yes	No	No	See “Displaying active replies” on page 32.
Modify operations				
Get help for the modify command	Yes	No	No	See “Getting help for the modify command” on page 34.
Canceling application servers and server instances	Yes	No	No	See “Canceling application servers and server instances with the modify command” on page 34.
Modifying trace variables dynamically	Yes	No	No	See “Modifying trace variables” on page 34.
Display control region and trace setting status	Yes	No	No	See “Displaying control region and trace settings status” on page 36.
Other WebSphere for z/OS operations				
ARM and restart	Yes	No	No	See “ARM and restart” on page 37.

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

Task	MVS console	SMEUI	TSO panel	Reference to associated procedure
Checking the contents of the name space	No	No	No	See <i>WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization</i> , GA22-7834, for details on checking the contents of the namespace.
Setting up error log streams for different servers and server instances	No	You can associate a log stream with a server from the SMEUI.	No	See “Setting up error log streams for different servers and server instances” on page 38. See <i>WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization</i> , GA22-7834, for details on setting up error log streams.
Setting up SMF recording	Yes	Enable it from here, but initiate it from the console.	No	See “Functions of SMF recording” on page 137. Also see <i>WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface</i> , SA22-7838.
Shutting down the WebSphere for z/OS environment	Yes	No	No	See “Stopping the WebSphere for z/OS environment” on page 29.
Taking a WebSphere for z/OS system server out of service	Yes	Application server only; You cannot take a WebSphere for z/OS system server out of service from the SMEUI (Daemon, IR, Naming, SM)	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 38.

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

Task	MVS console	SMEUI	TSO panel	Reference to associated procedure
Handling workload management and server failures	Yes	No	No	See “Using Workload Management for WebSphere for z/OS operations” on page 38.
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 38.
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 38.

Operating WebSphere for z/OS from the Systems Management User Interface

The operations application of the Systems Management User Interface allows you to perform the following tasks to operate your WebSphere for z/OS environment:

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

See *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838 for more information.

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:

Daemon

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

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.

server

A logical grouping of server instances. All server instances within a server are identical in structure and run the same set of applications. Administration is usually done at the server level. In addition, a server, from a management perspective, is a single entity in the sysplex. The server presents a single interface to the network and operator for control.

server instance

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 server instances within a server are identical in structure.

You can manage a server instance through the Systems Management User Interface Operations application or the MVS console via its unique name.

A server instance has two kinds of address spaces: a control region and one or more server regions. Application server code runs in a server region. A server region can be replicated based on the workload demands of the system. The control regions queue messages to the server regions.

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 V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834, for details about the required subsystems that must be in place before you can start up the environment.

Steps for starting the WebSphere for z/OS environment

Before you begin: When you start the Daemon, you will get SMS, Naming, and IR by default. However, you will need to go to the console or through automation to start the WebSphere for z/OS environment.

Note: This procedure also includes steps to start DB2. WebSphere for z/OS uses DB2 and requires that, in a sysplex configuration, each system that runs WebSphere for z/OS has access to a data-sharing DB2 instance.

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

1. Start all prerequisite subsystems (see *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834).

2. Start RRS with the MVS start command:

```
start atrrrs,sub=master
```

Note: You must start RRS before starting DB2.

3. Start DB2 on each system.

WebSphere for z/OS requires that a shared DB2 configuration be running on all the systems in the sysplex on which WebSphere for z/OS runs. This is because WebSphere for z/OS places its operational and management data in this shared DB2.

The following example demonstrates how DB2 is started on each system in the sysplex:

Example: One DB2 is shared between all systems. However, you need to use a unique DB2 name for each subsystem.

-DB1G start DB2

where DB1G is the name of the DB2 subsystem on system 1 in a 3-way sysplex.

-DB2G start DB2

where DB2G is the name of the DB2 subsystem on system 2 in a 3-way sysplex.

-DB3G start DB2

where DB3G is the name of the DB2 subsystem on system 3 in a 3-way sysplex.

4. Start the Daemon.

Because the Daemon may be a single point of failure, IBM recommends that, in a two-way sysplex, you have both Daemons running. The advantage of having more than one Daemon in the sysplex is availability. If one system goes down, your jobs will continue to run.

The full **WebSphere for z/OS runtime configuration** consists of:

- Daemon
- System Management (SM) — (started by the Daemon)
- Naming — (started by the Daemon)
- Interface Repository (IR) — (started by the Daemon)

Notes:

- a. WebSphere for z/OS servers can only run on systems where a Daemon has been started.
- b. Because WebSphere for z/OS is symmetric, IBM recommends that customer application servers be started on each system in the sysplex for availability.

See *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834, for more information.

To start the Daemon on the first system in your sysplex, issue the following command:

```
s bbodmn.daemon01,srvname='DAEMON01'
```

where `daemon01` is the step name of the control region you want to start and `DAEMON01` is the name of the server instance (server instance names are different on each system).

To start the Daemon on the second system in your sysplex, issue the following command:

```
s bbodmn.daemon02,srvname='DAEMON02'
```

where daemon02 is the step name of the control region you want to start and DAEMON02 is the name of the server instance.

Start the Daemon on all of the images or systems in your sysplex.

Example: This is an example from the syslog of the command and responses to the start of the Daemon.

```
s bbodmn.daemon01,svrname='DAEMON01'  
BBOU0007I CB SERIES DAEMON DAEMON01 IS STARTING.  
START BBOSMS.SYSMGT01,SRVNAME='SYSMGT01',PARMS=''  
BBOU0001I CB SERIES CONTROL REGION SYSMGT01 IS STARTING.  
START BBONM.NAMING01,SRVNAME='NAMING01',PARMS=''  
BBOU0001I CB SERIES CONTROL REGION NAMING01 IS STARTING.  
START BBOIR.INTFRP01,SRVNAME='INTFRP01',PARMS=''  
BBOU0001I CB SERIES CONTROL REGION INTFRP01 IS STARTING.  
BBOU0016I INITIALIZATION COMPLETE FOR DAEMON DAEMON01.  
BBOU0020I INITIALIZATION COMPLETE FOR CB SERIES CONTROL REGION SYSMGT01.  
BBOU0020I INITIALIZATION COMPLETE FOR CB SERIES CONTROL REGION INTFRP01.  
BBOU0020I INITIALIZATION COMPLETE FOR CB SERIES CONTROL REGION NAMING01.
```

Note: The only command entered is the start of the Daemon. The remaining start commands are generated internally.

-
5. Start the application servers (assuming you have already configured them).

The identifier associated with the started task, when specified on the start command, is used by other MVS commands.

- a. Start the first application server:

```
s bboasr1.bboasr1a,svrname='BBOASR1A'
```

where BBOASR1A is the application server name.

- b. Start the second application server:

```
s bboasr1.bboasr1b,svrname='BBOASR1B'
```

where BBOASR1B is the application server name.

- c. Start the third application server:

```
s bboasr1.bboasr1c,svrname='BBOASR1C'
```

where BBOASR1C is the application server name.

You know you are done when you see an initialization complete message such as:

```
BBOU0020I INITIALIZATION COMPLETE FOR CB SERIES CONTROL REGION BBOASR1C
```

Starting servers and server instances

This section describes how to start servers and server instances.

Notes:

1. Before you start any server or server instance, make sure that the Daemon is up by issuing one of the display commands (described in “Determining if the Daemon is up” on page 15). If the Daemon is not up, you must start it. Please see “Steps for starting the WebSphere for z/OS environment” on page 12 for this procedure.
2. From the MVS console, you need to start each individual server instance that you wish to run. The SMEUI, however, gives you the option to either do that or start a server which will start all defined server instances automatically.

Determining if the Daemon is up

When starting both servers and server instances, you need to know if the 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.

Steps for starting a server

Perform the following steps to start a server:

1. Determine if the Daemon is up. If it is up, go to the next step to start the server.

2. To start a server from the MVS console, you need to start each individual server instance that you wish to have started. See “Steps for starting an application server instance” on page 16.

Steps for starting an application server instance

Perform the following steps to start an application server instance:

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

Please see "Appendix B. Commands for operations" in *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838, for information on how to check that your resource managers are available.

2. Determine if the Daemon is up. If it is up, go to the next step to start the server instance.
-

3. For the actual procedure of starting an application server instance, please see "Appendix B. Commands for operations" in *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838.
-

Cold starting WebSphere for z/OS

See:

- *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838, for information on preparing for a cold start.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834, for the cold start procedure.

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

See *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834, for the hot start procedure.

Warm starting WebSphere for z/OS

See *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834, for the warm start procedure.

Quick starting WebSphere for z/OS

See *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834, for the quick start procedure.

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 V4.0.1 for z/OS and OS/390
- 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 37.

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:

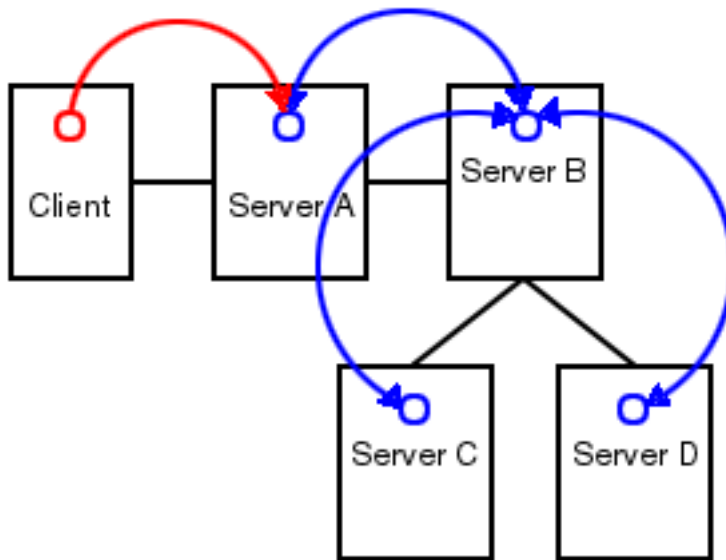


Figure 1. Organization of servers in a recoverable communication manager setup

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 18.

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 IIOF 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 V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834 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,opens 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.BB0ASR4A.IBM
RESOURCE MANAGER MAY RESTART ON ANY SYSTEM
```

You do **not** want to see:

```
RESOURCE MANAGER=BSS00.SY2.BB0ASR4A.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 V4.0.1 for z/OS and OS/390: Migration*, GA22-7860, *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834, 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 and SM server activities when attempting restart and recovery:

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

For the full message explanations, see *WebSphere Application Server V4.0.1 for z/OS and OS/390: Messages and Diagnosis*, GA22-7837.

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

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 an 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 22 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. cancel

Stop means that all currently-running transactions are carried out before the application control region is taken down, while **cancel** means that the application control region is immediately taken down without waiting for the active transactions to complete.

Example: When you **stop** a server instance, the current work is finished before the server is stopped, and when you **cancel** a server instance, it ends it 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 one system, it doesn't bring down the servers on the other system(s). Also, you don't have to individually stop the Systems Management, Naming, and IR servers. The Daemon brings down Systems Management, which in turn brings down Naming, IR, and all other control regions. You may optionally stop these servers directly. However, if you bring down Systems Management, it will bring down everything except the Daemon because control regions cannot exist without Systems Management.

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

1. Stop all of your application control regions

A **stop** can be issued for the server from the SMEUI.

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

All server instances in the sysplex associated with this server will stop. In addition, a **stop** can be issued against each individual server instance 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.

-
2. Stop the Daemon on each system in the sysplex (or cancel if it takes too long).

This must be done from the MVS console. The Daemon brings down Systems Management which, in turn, brings down Naming, IR, and all other control regions.

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 server instances

Please see the "Appendix B. Commands for operations" in *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838, for information that describes how to stop the application server instances that make up a server.

Cancel operations

Note: See "Stop vs. cancel" on page 29 for a description of the difference between stopping and cancelling an application control region.

Canceling application servers and server instances

Please see the "Appendix B. Commands for operations" in *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838, for information that describes how to cancel the application server instances that make up a server.

Canceling the Daemon

Note: Use caution with the cancel command.

This section describes how to cancel the Daemon. Canceling the Daemon brings down System Management which, in turn, brings down Naming, IR, and all other application server instances on this system.

Steps for canceling the Daemon

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

Issue **one** of the following commands to cancel the Daemon:

- `cancel bbodmn.daemon01`
- `cancel bbodmn.daemon01,armrestart`

Note: The command containing "armrestart" must be used if ARM is active and you want it to restart the 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 V4.0.1 for z/OS and OS/390: Installation and Customization, GA22-7834*, 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 server or server instance

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 server or server instance:

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 <i>address_space</i>	not restart the address space
CANCEL <i>address_space</i>	not restart the address space
CANCEL <i>address_space</i> , ARMRESTART	restart the address space
MODIFY <i>address_space</i> ,CANCEL	not restart the address space
MODIFY <i>address_space</i> ,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

This section describes how to get help for the modify command and how to use the modify command to cancel a server instance, alter trace variables for a server instance dynamically, and display server instance status.

Getting help for the modify command

To show a list of all the keywords you may use with the modify command, use the following command:

```
modify server-instance,help
```

or

```
f server-instance,help
```

Example: Here is a sample display:

```
BBOU0773I THE COMMAND MODIFY MAY BE FOLLOWED BY ONE OF THE FOLLOWING KEYWORDS:  
BBOU0774I CANCEL - CANCEL THIS CONTROL REGION  
BBOU0774I TRACEALL - SET OVERALL TRACE LEVEL  
BBOU0774I TRACEBASIC - SET BASIC TRACE COMPONENTS  
BBOU0774I TRACEDetail - SET DETAILED TRACE COMPONENTS  
BBOU0774I TRACESPECIFIC - SET SPECIFIC TRACE POINTS  
BBOU0774I TRACEINIT - RESET TO INITIAL TRACE SETTINGS  
BBOU0774I TRACENONE - TURN OFF ALL TRACING  
BBOU0774I TRACETOSYSPRINT - SEND TRACE OUTPUT TO SYSPRINT (YES/NO)  
BBOU0774I DISPLAY - DISPLAY STATUS
```

Canceling application servers and server instances with the modify command

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

To cancel a server instance using the modify cancel command:

```
f server-instance,cancel
```

Modifying trace variables

You can use the modify command to alter trace variables while the trace function is running.

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

```
f server-instance,traceall=x
```


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.

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-instance,tracebasic=(n,...)
```

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

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

Valid values for *n* when TRACEBASIC=(*n*,...) and TRACEDETAIL=(*n*,...) are:

- 0: RAS
- 1: Common Utilities
- 2: COS/Naming
- 3: COMM
- 4: ORB
- 5: IM
- 6: OTS
- 7: Shasta
- 8: Systems Management
- 9: OS/390 Wrappers
- A: Daemon
- B: IR
- C: Test
- D: COS/Query
- E: Security
- F: Externalization
- G: Adapter
- H: Lifecycle
- I: Identity
- J: JRAS (internal tracing—via direction from IBM support)
- K: Reference collections
- L: J2EE

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-instance, tracedetail=(n,...)
```

This command activates the most detailed tracing for the specified WebSphere for z/OS components and overrides lower 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.

To set specific trace points, use the following command:

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

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

Examples:

```
TRACESPECIFIC=03004020
```

```
TRACESPECIFIC=('xyz','abc',03004021)  
[where xyz and abc are environment variable names]
```

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

```
f server-instance, traceinit
```

To turn off tracing, use the following command:

```
f server-instance, tracenone
```

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

```
f server-instance, tracetosysprint=yes
```

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

```
f server-instance, tracetosysprint=no
```

Displaying control region and trace settings status

You can use the modify command to display the status of servers or trace settings.

To display a list of the other keywords that can be used with the display keyword on the modify command, use the following command:

```
f server-instance,display,help
```

Example: Here is a sample display:

```
BBOU0773I THE COMMAND DISPLAY, MAY BE FOLLOWED BY ONE OF THE FOLLOWING KEYWORDS:
BBOU0774I SERVERS - DISPLAY ACTIVE CONTROL REGIONS
BBOU0774I TRACE - DISPLAY INFORMATION ABOUT TRACE SETTINGS
BBOU0783I END OF OUTPUT FOR COMMAND DISPLAY,HELP
```

To display the status of servers in the active control region, use the following command:

```
f server-instance,display,servers
```

Example: Here is a sample display:

```
BBOU0777I SERVER          ASID  SYSTEM  LEVEL
BBOU0778I CBSYSGMT/SYSGMT01  40x  SY1     W401069
BBOU0778I CBNAMING/NAMING01  41x  SY1     W401069
BBOU0778I CBINTFRP/INTFRP01  42x  SY1     W401069
BBOU0778I BBOASR4 /BBOASR4A  45x  SY1     W401069
BBOU0783I END OF OUTPUT FOR COMMAND DISPLAY,SERVERS
```

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

```
f server-instance,display,trace
```

Example: Here is a sample display:

```
BBOU0791I TRACE INFORMATION FOR SERVER BBOASR4/BBOASR4A
BBOU0792I LOCATION = SYSPRINT BUFFER
BBOU0792I AGGREGATE TRACE LEVEL = 3
BBOU0792I EXCEPTION TRACING =
BBOU0792I BASIC TRACING = RAS(0), Common Utilities(1), COS/Naming(2),
ORB(4), IM(5), OTS(6), Shasta(7), Systems Management(8), OS/390
Wrappers(9), Daemon(A), IR(B), Test(C), COS/Query(D), Security(E),
Externalization(F), Adapter(G), Lifecycle(H), Identity(I), JRAS(J),
Reference Collections(K), J2EE(L)
BBOU0792I DETAILED TRACING = COMM(3)
BBOU0792I TRACE SPECIFIC = NONE SPECIFIED
BBOU0783I END OF OUTPUT FOR COMMAND DISPLAY,TRACE
```

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 daemon, 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 V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834, 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. See the *WebSphere Application Server V4.0.1 for z/OS and OS/390: Messages and Diagnosis*, GA22-7837, for instructions on how to access the error log.

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.

Note: WLM commands are sysplex in scope. Therefore, if you quiesce an application environment, it is, in effect, on all systems in the sysplex. Keep this in mind when using WLM commands.

Displaying the status of a WLM application environment

This section shows you how to display the status of an application environment.

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

Steps for displaying the status of application environments: Perform the following steps to display the status of your application environments:

1. To display the status of **all** your application environments, issue the following command:

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

Where "*" displays all application environments and states.

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

Example: Here is a sample display:

```
- SY1      d wlm,applenv=*
  SY1      IWM029I 11.21.11 WLM DISPLAY 469
           APPLICATION ENVIRONMENT NAME      STATE      STATE DATA
           BBOABBOA                          AVAILABLE
           BBOASR1                            AVAILABLE
           BBOASR2                            AVAILABLE
           BBOASR3                            AVAILABLE
```

BBOASR4	AVAILABLE
BBOASR5	AVAILABLE
BBOASR6	AVAILABLE
BBOASR7	AVAILABLE
BBOASR8	AVAILABLE
BBOASR9	AVAILABLE
CBINTFRP	AVAILABLE
CBNAMING	AVAILABLE
CBSYSMGT	AVAILABLE
PAAWYSV	AVAILABLE
PAAXFSV	AVAILABLE
PAAX1SV	AVAILABLE
PAAYYSV	AVAILABLE

-
- To display the status of a **specific** application environment, issue the following command:

```
d wlm,applenv=bboasr1
```

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

Example: Here is a sample display:

```
0- SY1      d wlm,applenv=bboasr1
   SY1      IWM029I 11.21.30 WLM DISPLAY 474
           APPLICATION ENVIRONMENT NAME      STATE      STATE DATA
           BBOASR1                          AVAILABLE
           ATTRIBUTES: PROC=BBOASR1S SUBSYSTEM TYPE: CB
```

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

When you issue the display command, there are three things of interest for WebSphere for z/OS:

- Whether or not the WLM application environment names match the server names.
- The state of the WLM application environment.
- Whether the proc associated with the application environment is the intended proc for the corresponding server region.

The most important information is the state of the application servers.

The states that may be displayed indicate the following:

available

Indicates that the everything is normal. The application servers are available.

quiesced (q)

Indicates that no server regions will start. A server will only be in the quiesced state when there is a problem with the application (e.g., it is abending). The quiesced state takes the control region out of service. New requests will continue to come in, but WLM will not start the quiesced server region.

stopped

Acts like the quiesced state but it won't start any application server regions. Requests will come into the control region and stay there without being processed. The server gets into the stopped state if there are terminating errors in the server region. If three address spaces terminate in ten minutes, the server gets into the stopped state. There are other reasons a server may stop, such as when an operator cancels a server region. It is not a good idea to cancel server regions. To take a server out of service, you should take down the control region rather than canceling the server region

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:

```
BB0U199E 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, it is stopped across the entire sysplex. When you resume, it is resumed on every system in the sysplex.

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 the command:

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

2. To start the application environment, issue the command:

```
v wlm,applenv=environment_name,resume
```

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 WLM command `V WLM,APPLENV=BBOASR4,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.

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 SMEUI 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 V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834 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 V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834, 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.
- If you need to stop DB2 to do maintenance, you must also stop WebSphere for z/OS. WebSphere for z/OS uses DB2 for its control information. Therefore, DB2 must be running to allow the WebSphere for z/OS runtime servers to run.
- 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 V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834 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 V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834, for more information.

- **IMS OTMA support:** When defining the logical resource manager for a MOFW server, you must select IMS_OTMA_PAA. 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 V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834, 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 53 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* 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. Follow your own backup procedures to back up the LDAP database that contains naming and interface repository data and the JNDI name space. If you restore LDAP data, be sure to coordinate the restoration with other WebSphere systems in the federated naming space. Otherwise, your naming space will not be consistent.
4. Incorporate the following in your normal backup procedures:
 - WebSphere for z/OS proclibs
 - WebSphere for z/OS loadlibs
 - The directory where WebSphere for z/OS run-time information is written (the value of the CBCONFIG environment variable; the default is /WebSphere390/CB390).
5. Back up your own application executables, databases, and bindings.
6. When you activate a conversation, System Management automatically backs up the current environment files for each server instance in */path/envfile/sysplex/server_instance/backup/*, where

path

Is the value of the CBCONFIG environment variable (default is /WebSphere390/CB390).

sysplex

Is the name of your sysplex.

server_instance

Is the name of the server instance.

The backup files have a time stamp in their names. You may wish to erase the older backup files as the backup directory fills up.

7. If you wish to back up a single server instance, you can use the export/import function in the Administration application. For details on how to do this, see *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836.
8. Follow your own backup procedures to back up the WebSphere for z/OS database that contains system management, reference collection, and J2EE data. Back up all table spaces created by jobs BBOMCRDB and BBOICD. Coordinate your backup of the WebSphere for z/OS database with the backup of the LDAP database, because it is very important that both be synchronized. If you restore the database, be sure to coordinate the restoration with that of the LDAP database.

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

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 V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834, 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) <i>automatically started</i>		
2	RACF		
3	System Logger <i>automatically started</i>		
4	RRS	Unix System Services (doesn't complete until TCP/IP is done, but can be started before TCP/IP)	
5	VTAM	TCP/IP	
6	APPC	TSO	
7	DB2	CICS and CICS Transaction Gateway	IMS and IMS Connect
8	LDAP	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 <ul style="list-style-type: none"> • JVM: Started inside WebSphere for z/OS and WebServer • LE: Started inside WebSphere for z/OS and WebServer 	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 <i>could</i> 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.
Handling in-flight or indoubt transactions if there is a failure	If you are using APPC for communications and it fails, do the following: <ol style="list-style-type: none"> 1. Shutdown all servers with APPC connectivity. 2. Restart APPC (if it totally failed). 3. Restart the WebSphere for z/OS server. <p>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.</p>

Table 3. APPC automation and recovery scenarios (continued)

Task	APPC automation and recovery scenarios
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	<ul style="list-style-type: none"> • 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 catastrophic failure of the WebSphere for z/OS server Regions	Following a catastrophic failure of the WebSphere for z/OS server regions, you can use the following WLM resume command: <code>v wlm,applenv=applenvname,resume</code>
Where to find more information	<ul style="list-style-type: none"> • <i>z/OS MVS Planning: Workload Management</i>, SA22-7602 • <i>z/OS MVS Programming: Workload Management Services</i>, 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	<ul style="list-style-type: none"> • <i>z/OS Security Server RACF System Programmer's Guide</i>, SA22-7681 • <i>z/OS Security Server RACF Security Administrator's Guide</i>, SA22-7683

RRS automation and recovery scenarios

Table 6. RRS automation and recovery scenarios

Task	RRS automation and recovery scenarios
Startup	<p>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:</p> <pre>start atrrrs,sub=master</pre> <p>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.</p>
Shutdown	<p>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:</p> <ol style="list-style-type: none"> 1. Quiesce WebSphere for z/OS. 2. Shutdown WebSphere for z/OS. 3. Cancel RRS. <p>Note: You may want to bring down the DB2 you are using for WebSphere for z/OS before canceling RRS.</p> <p>To cancel RRS, issue the command:</p> <pre>setrrs cancel</pre>
Handling in-flight and indoubt transactions if there is a failure	<p>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.</p> <p>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.</p>

Table 6. RRS automation and recovery scenarios (continued)

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

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	<i>z/OS UNIX System Services Planning, GA22-7800</i>

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.

Table 9. DB2 automation and recovery scenarios (continued)

Task	DB2 automation and recovery scenarios
Handling in-flight or indoubt transactions if there is a failure	<p>Use the RRS panels to resolve. See <i>z/OS MVS Programming: Resource Recovery</i>, 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:</p> <pre>DISPLAY THREAD(*) TYPE(INDOUBT)</pre> <p>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:</p> <pre>-RECOVER INDOUBT (RRSAF) ACTION(COMMIT) NID(B1D379D17ED6CF900000009401010000)</pre> <p>where the nid is the one that you cut from the display command. You can issue this command to roll back the transaction:</p> <pre>-RECOVER INDOUBT (RRSAF) ACTION(ABORT) NID(B1D379D17ED6CF900000009401010000)</pre>
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 restart 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 xii.

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.

Table 10. CICS automation and recovery scenarios (continued)

Task	CICS automation and recovery scenarios
Shutdown	Shutdown the WebSphere for z/OS application control region that uses CICS as a backing store, then shutdown the CICS service.
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: <ol style="list-style-type: none"> 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	<i>CICS Operations and Utilities Guide, SC34-5991</i>

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

Table 11. IMS automation and recovery scenarios (continued)

Task	IMS automation and recovery scenarios
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.
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: <ol style="list-style-type: none"> 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	<i>IMS/ESA Operator's Reference, SC26-8742</i>

LDAP automation and recovery scenarios

Table 12. LDAP automation and recovery scenarios

Task	LDAP automation and recovery scenarios
Startup	<p>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 <i>before</i> Naming and IR. To run the LDAP server, TCPIP, OMVS, and DB2 must all be up before the LDAP server.</p> <p>Note: There are two LDAP modes supported:</p> <ol style="list-style-type: none"> 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	<p>Shutdown Naming and IR, then OMVS and DB2. For the LDAP server, shutdown the LDAP server, then TCPIP and DB2, and then OMVS.</p>
Handling in-flight or indoubt transactions if there is a failure	<p>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.</p>
How to determine if LDAP is running	<p>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.</p>
What happens to WebSphere for z/OS if LDAP goes down?	<ul style="list-style-type: none"> • In MOFW WebSphere for z/OS regions, LDAP runs within the WebSphere for z/OS address space, so this is not an issue. If the server goes down, then LDAP also 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.

Table 12. LDAP automation and recovery scenarios (continued)

Task	LDAP automation and recovery scenarios
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	<i>z/OS Security Server LDAP Server Administration and Use</i> , 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.

Table 13. NFS automation and recovery scenarios (continued)

Task	NFS automation and recovery scenarios
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.
Where to find more information	<ul style="list-style-type: none"> • <i>z/OS Network File System User's Guide</i>, SC26-7419 • <i>z/OS Network File System Customization and Operation</i>, 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 29.
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 "Steps for starting the WebSphere for z/OS environment" on page 12.
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.

Naming automation and recovery scenarios

Table 15. Naming automation and recovery scenarios

Task	Naming automation and recovery scenarios
Startup	The Naming control region is started automatically (if you set it up as recommended in <i>WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization</i> , GA22-7834) and implicitly when WebSphere for z/OS is started. Naming server regions (that actually do the work) are started on demand by the server when work has to be done.
Shutdown	When WebSphere for z/OS is stopped, the Naming control region and all active server regions are automatically stopped.
Handling in-flight or indoubt transactions if there is a failure	If there is a failure during processing, Naming relies on RRS to issue a rollback directly to LDAP and DB2. As a result, any work is rolled back along with it.
How to determine if Naming is running	If WebSphere for z/OS is running, Naming is running. If Naming is down, WebSphere for z/OS is unusable. In addition, the Naming task can be monitored using SDSF.
What happens to WebSphere for z/OS if Naming goes down?	Naming server region failures are recovered by WLM. WLM just starts a new server region. If the Naming control region drops and isn't restarted by ARM, WebSphere for z/OS is unusable.
What happens to other subsystems if Naming goes down?	This doesn't have any impact on other subsystems.
Note: In a sysplex, you need only one Naming server. Therefore, if Naming comes down on one system, you can keep running as long as one Naming server is running somewhere in the sysplex.	

Interface Repository automation and recovery scenarios

Table 16. Interface Repository (IR) automation and recovery scenarios

Task	Interface Repository automation and recovery scenarios
Startup	The IR control region is started automatically (if you set it up as recommended in <i>WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization</i> , GA22-7834) and implicitly when WebSphere for z/OS is started. IR server regions (that actually do the work) are started on demand by the server when work has to be done.
Shutdown	When WebSphere for z/OS is stopped, the IR control region and all active server regions are automatically stopped.
Handling in-flight or indoubt transactions if there is a failure	If there is a failure during processing, IR relies on RRS to issue a rollback directly to LDAP and DB2, and, as a result, any work is rolled back along with it.

Table 16. Interface Repository (IR) automation and recovery scenarios (continued)

Task	Interface Repository automation and recovery scenarios
How to determine if IR is running	The IR server can be monitored using SDSF.
What happens to WebSphere for z/OS if IR goes down?	IR server region failures are recovered by WLM. WLM just starts a new server region.
What happens to other subsystems if IR goes down?	This doesn't have any impact on other subsystems.
Note: In a sysplex, you only need one IR server. Therefore, if IR comes down on one system, you can keep running as long as one IR is running somewhere in the sysplex.	

Systems Management (SM) automation and recovery scenarios

Table 17. Systems Management (SM) automation and recovery scenarios

Task	Systems Management (SM) automation and recovery scenarios
Startup	Systems Management is a WebSphere for z/OS server that is started automatically during WebSphere for z/OS Daemon startup. As a WebSphere for z/OS server, it prerequisites by the WebSphere for z/OS infrastructure (DB2, RRS, OMVS, LDAP, WLM, etc.).
Shutdown	The Systems Management server is shutdown automatically when the Daemon is shutdown.
Handling in-flight or indoubt transactions if there is a failure	Systems Management lets ORB+OTS handle its transactions. Each request that is routed to Systems Management implicitly starts a transaction (this is handled by ORB/OTS). If something goes wrong, either Systems Management explicitly requires a rollback from OTS or the rollback is done automatically by the ORB. ORB and OTS rely on RRS to manage commits and rollbacks.
How to determine if Systems Management is running	Check whether the Systems Management control region is operating. Use SDSF.
What happens to WebSphere for z/OS if Systems Management goes down?	WebSphere for z/OS cannot operate if Systems Management fails.
What happens to other subsystems if Systems Management goes down?	Because WebSphere for z/OS goes down if Systems Management goes down, other subsystems can be severely affected. If WebSphere for z/OS can't resolve its indoubt transactions when it fails, it holds up CICS, IMS, etc.

Table 17. Systems Management (SM) automation and recovery scenarios (continued)

Task	Systems Management (SM) automation and recovery scenarios
Where to find more information	<ul style="list-style-type: none"> • <i>WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization, GA22-7834</i> • <i>WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface, SA22-7838</i>

WebServer automation and recovery scenarios

Table 18. 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.
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 unaffected if WebServer goes down.
Where to find more information	<i>z/OS HTTP Server Planning, Installing, and Using, SC34-4826</i>

Chapter 7. WebSphere for z/OS administration procedures

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

For further information, please see:

- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834 to:
 - Set up RACF and DCE system security and userids
 - Update DNS definitions
 - Set up your TCP/IP network
 - Update your hosts file as you expand your sysplex.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836 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 *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836, 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:

- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836, for more general information.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834, 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.	<i>z/OS TSO/E Administration, SA22-7780, or z/OS Security Server RACF Security Administrator's Guide, SA22-7683</i>
Updating the access control list for LDAP	"Steps for updating the access control list for LDAP"
Defining the new administrator to the Administration application	<i>WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface, SA22-7838</i>
Granting the administrator user ID System Management database authority	"Step for granting the new administrator database authorities" on page 71

Steps for updating the access control list for LDAP

If you add an administrator for the Administration application, you must add that administrator to the access control list in LDAP.

Before you begin: You need to set up the LDAP server. We assume you have already set up an exclusive LDAP server for WebSphere for z/OS administrative purposes. For more information about setting up the LDAP server, see *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization, GA22-7834*.

You also need the `bboslapd.conf` file currently in use by the LDAP server.

Perform the following steps to change the access control list for LDAP:

1. View the `bboslapd.conf` file and note the following:
 - a. Administrator distinguished name.

Example:

```
adminDN          "cn=CBAAdmin"
```

- b. Administrator password.

Example:

```
adminPW      mypass
```

- c. Root naming context (RDN) for the WebSphere for z/OS name space structure.

Example:

```
suffix      "o=BOSS,c=US"
```

2. Start the LDAP server:

```
S BBOLDAP
```

3. Extract the current access control list with the `ldapcp` command.

Example:

```
/u/myself-> ldapcp -p 1389 -h 127.0.0.1 -d "cn=CBAdmin" -w *****
GLD6019I Communicating with server on port 1389.
ldapcp> acl q ob "o=boss,c=us"
object = o=boss,c=us
aclSource = O=BOSS,C=US
aclPropagate = TRUE

acl = access-id:CBADMIN:object:ad:normal:rws
acl = access-id:CBSYMCRI:object:ad:normal:rws

acl = group:CN=ANYBODY:normal:rsc

acl = access-id:CN=BOSSAdmin,O=BOSS,C=US:object:ad:normal:rws

ldapcp>quit
```

4. Create a new file in your home directory (for example, `acl_update.txt`). Add these lines to the file:

```
dn: o=boss, c=us
changetype:modify
replace:x
```

5. Following the first three lines you added to the file, add `aclentry` statements for each of the `acl` lines you extracted in step 3. Add a new `aclentry` statement for `USER1`.

Notes:

- a. It is important to add the dash ('-') at the end.
- b. The output format of the `ldapcp` command is not the same as the input `aclentry` lines ("`acl=`" must change to "`aclentry:`", for example).
- c. The `aclentry` for `USER1` in the example gives `USER1` the same authority as `CBADMIN`.

Example:

```
aclentry: access-id:cn=BOSSAdmin, o=boss, c=us:normal:rWSC:object:ad
aclentry: access-id:USER1:normal:rWSC:object:ad
aclentry: access-id:CBADMIN:normal:rWSC:object:ad
aclentry: access-id:CBSYMCRI:normal:rWSC:object:ad
aclentry: group:CN=ANYBODY:normal:rsc
-
```

-
6. Save the update file and issue the following `ldapmodify` command:

```
u/myself-> ldapmodify -v -p 1389 -D "cn=CBAdmin,o=BOSS,c=US" -w mypass -f acl_update.txt
```

Result: `ldapmodify` responds with:

```
modifying entry o=BOSS, c=US
```

-
7. Repeat step 3 on page 70 to verify that you have added a new user to the access control list.
-

You know you are done when you see the new user in the access control list.

Step for granting the new administrator database authorities

Your new administrator requires execute authority for `CBSYSMGT_PKG` and select, update, insert, and delete authority for the tables required for an administrator to deploy a J2EE application in the system management database.

Before you begin: You need to have a user ID with DB2 for z/OS or 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. Tuning considerations specific to MOFW objects are included in Appendix B, “MOFW tuning considerations” on page 183. For steps in diagnosing performance problems, refer to Chapter 10, “WebSphere for z/OS performance monitoring procedures” on page 121.

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

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, or auxiliary, storage default should be about 350 MB per server region, which includes a 256 MB default heap size and a default initial LE heap size of 80 MB. You can reduce this to about 225 MB by reducing the heap size to 128 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 either a fast DASD subsystem (for example, IBM Shark), running with caching turned on, or DASD FastWrite.

- Networking

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

- 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

The first three will be dealt with in this chapter separately. The fourth will be briefly touched on, but more information on tuning the application can be found in *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836 and *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling CORBA Applications*, SA22-7848.

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 384MB 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 server region 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 V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834.

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 the additional virtual storage used.

Real storage

From reading "Virtual storage" on page 75, 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. The control regions utilize around 18MB of that real storage, and the amount of real storage utilized in the server regions is dependent 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. As a general rule when you configure for 64 bit real you would dedicate most of the storage to central storage.

Garbage collection and JVM_HEAPSIZE

Specifying a sufficient JVM_HEAPSIZE 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.

Tips: Use the following guidelines to specify JVM_Heapsize:

- 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 JVM_HEAPSIZE. The default of 256M is a good starting point but may need to be raised for larger applications. By default, the servers run with JVM_HEAPSIZE=256M and JVM_MINHEAPSIZE=256M. When specifying either a larger or smaller JVM heapsize value, IBM recommends that you code **both** the maximum and minimum values that you desire.
- It is good for the JVM_MINHEAPSIZE to equal the JVM_HEAPSIZE because it allows the allocated storage to be completely filled before GC kicks in. Otherwise, GC would be running constantly, trying to maintain small chunks of storage, and performance would be compromised.
- Make sure the region is large enough to hold the specified JVM heap.
- Beware of making the JVM_HEAPSIZE *too* large. While it initially improves performance by delaying garbage collection, it ultimately effects response time when garbage collection eventually kicks in (because it runs for a longer time).
- To determine if you are being affected by garbage collection, you can specify JVM_ENABLE_VERBOSE_GC. 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 2% 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 76,

$$118/32225 * 100 = 0.366\%$$

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.

Server region recycling

WebSphere for z/OS has a feature called server region recycling. This gives the installation a threshold for the number of transactions to execute in a specific server region before the server region is thrown away. This can be very helpful in improving the performance of applications which have storage leaks.

The default specification for server region recycling is 50 thousand transactions. What that means is that, after 50 thousand transactions, the server region will no longer pick up any new work. A new server region is started to pick up the new work while the old server region finishes up the work it already has. When it finishes, it terminates, and the process continues with the new server region.

When your application code is leaking storage, it can cause each storage obtain to get slower and slower. After the specified number of transactions, you get a new SR and storage obtains are fast again. This produces a somewhat saw-toothed performance curve where performance starts good and degrades until the SR is recycled and performance once again improves. Clearly the long term solution for application leaks is to fix the application, but server region recycling can help when you can't fix the application immediately.

Note: If your application exploits one or more of the following, then you must disable the Server Region Recycle Function by setting the Server Region Recycle Limit to 0 (zero):

- HTTP Session Affinity (i.e., Session-in-memory using cookies over the HTTP Transport Handler).
- 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."

Example:

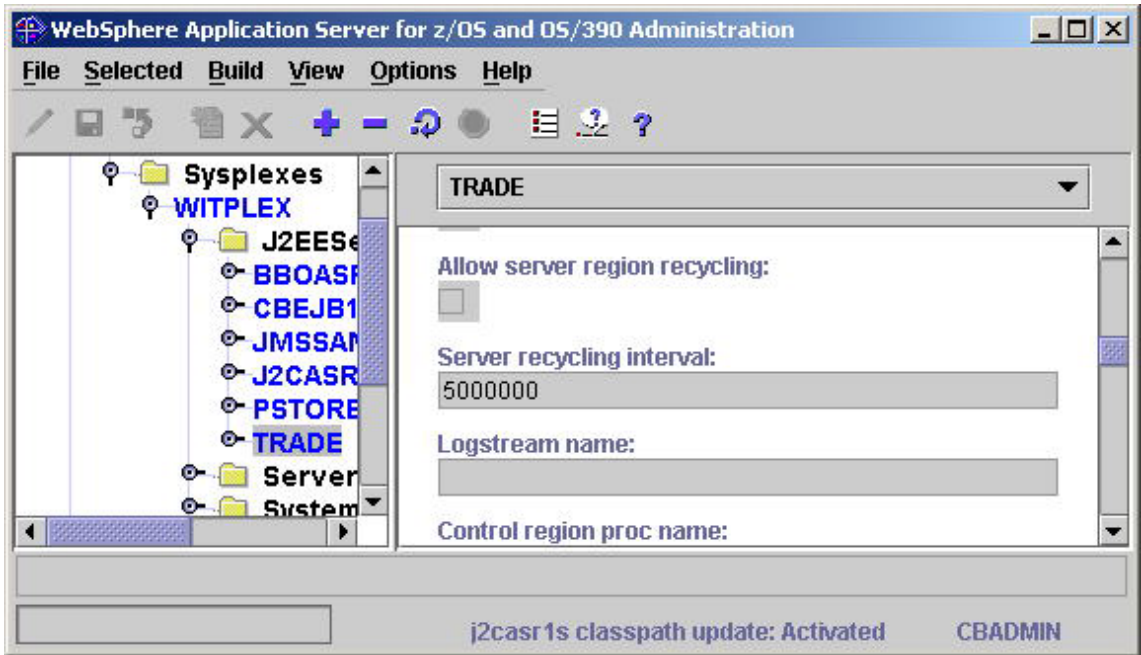


Figure 2. Disabling the Server Region Recycle Function

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(EXIT(S,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 artifacts, 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 149.

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.

- Ensure that the SMF dataset CI sizes are large to support the most efficient writing of SMF data to the dataset.

Tuning tips for the system logger

- 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.
- If it's not possible to use CF logs, use fastwrite DASD and make sure the logs are allocated with large CI sizes.
- Before OS/390 R10 (which had DB2 6.1), you could write as many as four log records to a logger. 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 for 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).
- The default values of LOGR policy may have an impact on performance. We recommend the default settings in Table 19.

Table 19. 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 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)

```


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

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

The following example shows the server region joblog 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 on page 82 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 server region 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,)
. . .
```

- From the second storage utilization report, shown in 2 on page 83, 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 LE Parm= 'RPTOPTS(ON),RPTSTG(ON,32,8,80,43,192,48,520,20,1232,5,2048,20) '
```

The following example shows the server region 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,)
. . .

```

- 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.
- Make sure that you remove RPTSTG since it does incur a small performance penalty to collect the storage use information.
- 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 82.

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

Performance of the USS file system can have a large impact on your application. This is especially true if your application accesses files in the HFS frequently. In addition, the runtime itself stores information such as your java classes, property files, and files that make up your web application in the file

system. Frequently used files can be cached beyond what is done by default in the file system using the USS filecache command.

In order to determine what files would be good candidates for file caching you can use SMF 92 records.

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. WebSphere for z/OS requires you to have a shared HFS 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 dataspace. 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.

Tuning tips for WLM

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:

- Daemon, Naming, SMS, IR, Application Control Regions

- Classify as SYSSTC or high Velocity

- Application Server regions

- Classify with velocity goal high enough to get started
Application work is actually classified under the application environment
- Application Environment for work running under server regions
 - 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 server region per system.
 - Under WLM, you can control how many server regions can be started for each server instance. If you need more than one server region in a server instance, make sure that "No Limit" is selected for the application environment associated with your server instance. For information about setting up WLM performance goals, see *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization, GA22-7834* or *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. **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)
 - Response times
 - How much of the machine you are using

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 information on application programming for JDBC, refer to "JDBC topic" on page 117.

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. Make sure DASD fastwrite is enabled if you have it. In our runs, the difference was about 2x (based on an I/O difference of 35msec). We also found that the RMF reports do not seem to show the real I/O difference. They showed .1msec before and after the change, but a GTF I/O trace definitely showed the problem was an SSCH and two I/O interrupts—one quick and one later. The RRS CTRACE showed that the time for exits was rather high during commit. It will help to make sure that the database tables are defined in multiples of cylinders.
- Next ensure that you have tuned your bufferpools so that the most often-read data is in memory as much as possible. The setting-up of buffer pool size is a balancing act between defining enough memory to hold everything and not defining more than 2G.
- You may want to consider pre-formatting tables that are going to be heavily used. This avoids formatting at runtime.
- Allocate tables in cylinders (multiple of 720).
- Manage EDM pool sizes to minimize DASD access without too much storage.
- 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.

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

```
//DSNTIJUZ JOB (DB2R71),
//          'SYSPROG',NOTIFY=&SYSUID,
//          CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),
//          REGION=0M
//*
//*****/00010000
/** JOB NAME = DSNTIJUZ                               */00020000
/**                                                    */00030000
/** DESCRIPTIVE NAME = INSTALLATION JOB STREAM        */00040000
/**                                                    */00050000
/** LICENSED MATERIALS - PROPERTY OF IBM              */00060000
/** 5675-DB2                                           */00070000
/** (C) COPYRIGHT 1982, 2000 IBM CORP. ALL RIGHTS RESERVED. */00080000
/**                                                    */00090000
/** STATUS = VERSION 7                                */00100000
```

```

|      /*                                                    */00110000
|      /* FUNCTION = DSNZPARM AND DSNHDECP UPDATES          */00120000
|      /*                                                    */00130000
|      /* PSEUDOCODE =                                     */00140000
|      /* DSNTIZA STEP ASSEMBLE DSN6.... MACROS, CREATE DSNZPARM */00150000
|      /* DSNTIZL STEP LINK EDIT DSNZPARM                  */00160000
|      /* DSNTLOG STEP UPDATE PASSWORDS                    */00170000
|      /* DSNTIZP STEP ASSEMBLE DSNHDECP DATA-ONLY LOAD MODULE */00180000
|      /* DSNTIZQ STEP LINK EDIT DSNHDECP LOAD MODULE      */00190000
|      /* DSNTIMQ STEP SMP/E PROCESSING FOR DSNHDECP       */00200000
|      /*                                                    */00210000
|      /* NOTES = STEP DSNTIMQ MUST BE CUSTOMIZED FOR SMP. SEE THE NOTES */00220000
|      /*          NOTES PRECEDING STEP DSNTIMQ BEFORE RUNNING THIS JOB. */00230000
|      /*                                                    */00240000
|      /* CHANGES:                                        */
|      /* 08/30/01-AJC CHANGED IDFORE 2000 TO 200         */
|      /*                                                    */
|      /*          IDBACK 500 TO 1800                       */
|      /*                                                    */
|      /*          CTHREAD 70 TO 2000                       */
|      /*                                                    */
|      /******/00250000
|      /*                                                    */00260000
|      /*DSNTIZA EXEC PGM=ASMA90,PARM='OBJECT,NODECK'      */00270000
|      /*SYSLIB DD DISP=SHR,                                */00280000
|      /*          DSN=DSN710.SDSNMACS                      */00290000
|      /*          DD DISP=SHR,                              */00300000
|      /*          DSN=SYS1.MACLIB                          */00310000
|      /*SYSLIN DD DSN=&&LOADSET(DSNTILMA),DISP=(NEW,PASS), */00320000
|      /*          UNIT=SYSDA,                               */00330000
|      /*          SPACE=(800,(50,50,2)),DCB=(BLKSIZE=800) */00340000
|      /*SYSPRINT DD SYSOUT=*                               */00350000
|      /*SYSUDUMP DD SYSOUT=*                               */00360000
|      /*SYSUT1 DD UNIT=SYSDA,SPACE=(800,(50,50),,,ROUND) */00370000
|      /*SYSUT2 DD UNIT=SYSDA,SPACE=(800,(50,50),,,ROUND) */00380000
|      /*SYSUT3 DD UNIT=SYSDA,SPACE=(800,(50,50),,,ROUND) */00390000
|      /*SYSIN DD *                                        */00400000
|      DSN6ENV MVS=XA                                     */00410000
|      DSN6SPRM RESTART,                                  X00410001
|
|      .
|      .
|      .
|      AUTH=YES,                                         X00410005
|      .
|      .
|      .
|      CACHEDYN=YES,                                     X00410009
|      .
|      .
|      .
|      MAXKEEPD=16000,                                   X00410036
|      .
|      .

```

.
.
.

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 allocated.

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).

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
          TCPRCVBFRSIZE 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 WebSphere MQ and JMS

There are a number of WebSphere MQ (formerly MQSeries) 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 WebSphere MQ or JMS:

- Turn off MQ tracing in the MQ ZPARMS (IBM recommends).

The default value for this is

```
TRACSTR=YES,          TRACING AUTO START          X
```

as seen in the example below. You need to change this value to NO and recompile the the MQ ZPARMS.

Example:

```
//WITAZPRM JOB (MVS,SUPPORT),'EDARNOLD',
//          CLASS=A,MSGCLASS=X,
//          MSGLEVEL=(1,1),NOTIFY=&SYSUID;//*****
//*
//* @START_COPYRIGHT@
//* Statement:      Licensed Materials - Property of IBM
//*
//*                5655-F10
//*                (C) Copyright IBM Corporation. 1993, 2000
//*
//* Status:         Version 5 Release 2
//* @END_COPYRIGHT@
//*
//* 05/22/01 ELA Turned off logging
//*                changed OFFLOAD=YES, to OFFLOAD=NO,
//*
//*
//*****
//* CUSTOMIZE THIS JOB HERE FOR YOUR INSTALLATION
//* YOU MUST DO GLOBAL CHANGES ON THESE PARAMETERS USING YOUR EDITOR
//*****
//*
//*                IBM MQSeries for OS/390
//* This job assembles and links a new system parameter module.
//* Edit the parameters for the CSQ6LOGP,
//* CSQ6ARVP and CSQ6SYSP macros to determine your
//* system parameters.
//* See "MQSeries for OS/390 System Setup Guide"
//* for a full description of the parameters.
//*
//* Replace   DOU.MQ
//*                with the high level qualifier of the
```

```

/**          SCSQMACS and SCSQAUTH target libraries.
/**
/**  Replace  ++HLQ.USERAUTH++
/**          with the data set name of the authorized
/**          load library in which to store your
/**          system parameter module.
/**
/**  Replace  ++NAME++
/**          with the name of your system parameter
/**          module.
/**          Note - do NOT use the default version
/**          name of CSQZPARM if you are using the
/**          IBM library SCSQAUTH to store your
/**          system parameter module.
/**
/**
/*******
/**
/**          Assemble step for CSQ6LOGP
/**
/**LOGP  EXEC  PGM=ASMA90,PARM='DECK,NOOBJECT,LIST,XREF(SHORT)',
/**          REGION=4M
/**SYSLIB DD DSN=DOU.MQ.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=DOU.MQ.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 RETENTION 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=DOU.MQ.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=300,      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=20,                   NUMBER OF NON-TSO CONNECTIONS     X
          IDFORE=100,                  NUMBER OF TSO CONNECTIONS         X
          LOGLOAD=500000,               LOG RECORD CHECKPOINT NUMBER      X
          OTMACON=(,DFSYPDRO0,2147483647,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=YES,                   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
/*

```

Example:

RESAUDIT=YES,	RESLEVEL AUDITING	X<--maybe should be no
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=YES,	TRACING AUTO START	X<--probably should be no
TRACTBL=99,	GLOBAL TRACE TABLE SIZE X4K	X
WLMTIME=30,	WLM QUEUE SCAN INTERVAL (SEC)	X

- 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 be updated at any time and 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 you 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 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 z/OS 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 93.
- 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 than 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 about in-flight transactions. When there are multiple server instances defined in a system or a sysplex, a request may end up on the wrong server instance. 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. **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

Tips: When using Java, you should 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 V4.0.1 for z/OS and OS/390 is 1.3.1 and PTF 13.
- Have the most recent PTFs, since almost every PTF level has improved performance of the JVM.
- Have sufficient JVM_HEAPSIZE. Figure 3 on page 101 shows an example of setting HEAP sizes in the Administration application.
- Run with the JIT (Just In Time) compiler active. This is done by omitting the "JVM_COMPILER=" option from the environment file. To ensure that you

are running with JIT enabled, look in the joblog (see “Server configuration example” on page 103 for a joblog example). It should say “JIT enabled” somewhere.

- 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.
- Have `CLASSPATH` point to only the classes you need (the classes that are referenced most frequently should be located near the front of the path, if possible).
- Verify the `CLASSPATH` as part of the Java configuration.
- JNDI naming has been improved in 4.0.1 to add caching of LDAP-backed names. There are configuration parameters to control this caching—please find them in *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836.
- 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 either collect `CTRACE` records from the file system or you can use `JVM_DEBUG=1` to turn on the JVM verbose mode. This will generate a lot of information but one of the things will be messages about class not found. You should only do this to determine if this is the problem and then remove `JVM_DEBUG` from the current `.env`. 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.

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_HEAPSIZE` and `JVM_ENABLE_VERBOSE_GC`, see “Garbage collection and `JVM_HEAPSIZE`” on page 76.

Example: This example shows the screen shot of the Administration application on how to set the heap sizes.

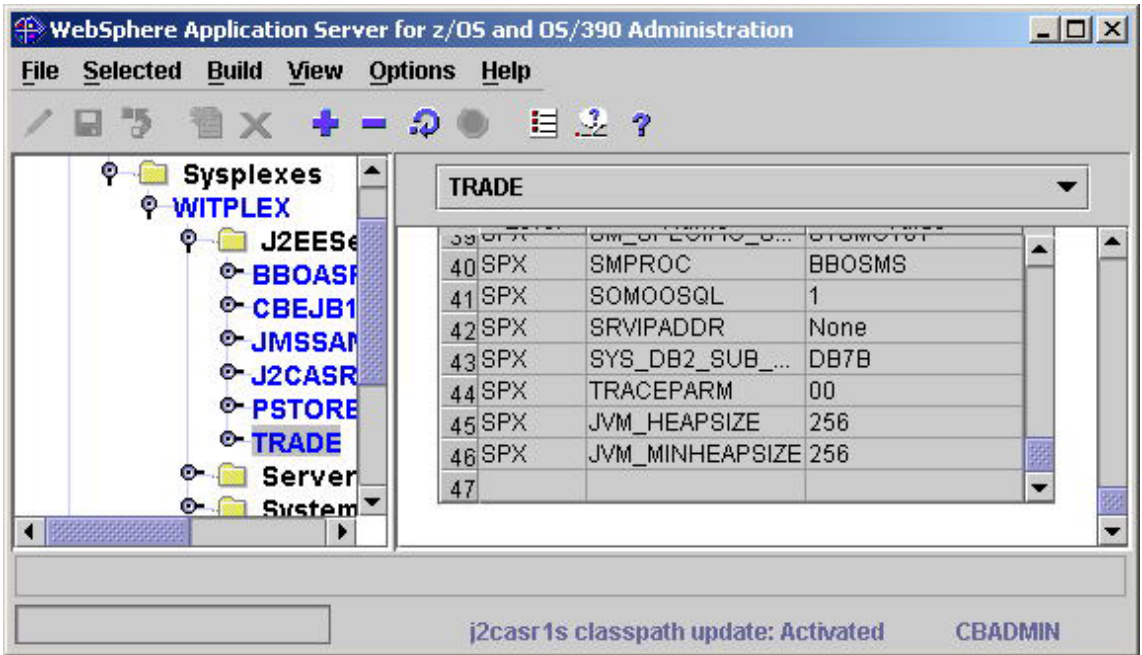


Figure 3. Setting heap sizes in the Administration application.

Tuning tips for LDAP

Tips: When using LDAP, you should ensure that:

- LDAP has enough threads (maxthreads) in the slapd.conf file. We recommend 1 thread per server region.
- Select the TDBM data module to store data.

When installing LDAP, you can select either of two different data models for storing data: TDBM or RDBM. TDBM performance is better than RDBM.

Tuning tips for CICS

These recommendations only apply to WebSphere applications 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 a value 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.

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 “Server configuration example” on page 103 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 TRACEALL=0 or 1, and that TRACEBASIC and TRACEDetail are not set.

Note: Please see *WebSphere Application Server V4.0.1 for z/OS and OS/390: Messages and Diagnosis, GA22-7837* for more information on trace options and settings.

2. If you use any level of tracing, including TRACEALL=1, ensure that you set TRACEBUFFLOC to BUFFER. TRACEALL=1 will write exceptions to the trace log as well as to the ERROR log.
3. To reduce memory requirements, you can set the TRACEBUFFERNUMBER=4 and TRACEBUFFERSIZE=128, which will get 512KB of storage for the trace buffers (the minimum allowed).
4. If you are tracing to sysprint with TRACEALL=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.

Tip for JRAS tracing: Make sure you disable JRAS tracing. To do this, look for the following lines in the trace.dat file pointed to by the JVM properties file:**Example:**

```
com.ibm.ejs.*=all=(disable)
com.ibm.ws390.orb=all=(disable)
```

Ensure that both lines are set to "=disable" or delete the two lines altogether. For more information, see "Chapter 4. Tracing Java server applications" in *WebSphere Application Server V4.0.1 for z/OS and OS/390: Messages and Diagnosis*, GA22-7837.

Note: If TRACEBUFFLOC=BUFFER, you may be tracing and not know it.

Tip for application trace and debug: WebSphere for z/OS provides the ability to trace and debug an application in each server definition. IBM recommends not using Object Level Trace (OLT) or distributed debugger on a production server. To disable OLT and debugger, uncheck the "Debugger allowed" checkbox. This will prevent the WebSphere for z/OS runtime from calling the OLT/debug interfaces for each method, which will improve performance.

Server configuration example

Example: This is an example of a joblog output for a BBOASR2S address space. Pay particular attention to the bold lines.

```
BBOU0025I ERRORS WILL BE WRITTEN TO CERR FOR JOB BBOASR2S.
BBOJ0011I JVM Build is "J2RE 1.3.1 IBM OS/390 Persistent Reusable
VM build hm131s-20020105 (JIT enabled: jitc)".
1          1 //BBOASR2S JOB MSGLEVEL=1                                STC05899
          2 //STARTING EXEC BBOASR2S,IWMSSNM=BBOASR2A
          3 XXBBOASR2S PROC IWMSSNM='BBOASR2A',PARMS='-ORBsrvname '
          4 XX SET CBCONFIG='/WebSphere390/CB390'
          5 XX SET RELPATH='controlinfo/envfile'
          XX* SET LEPARM='XPLINK(ON)'
          XX* SET LEPARM='HEAP(20M,1M,ANY,KEEP,,)'
          XX* SET LEPARM='HEAP(20M,1M,ANY,KEEP,,),RPTSTG(ON),RPTOPTS(ON)'
          XX* SET LEPARM='RPTSTG(ON),RPTOPTS(ON)'
          XX* SET LEPARM='trace(on,,NODUMP,le=4)'
          6 XXBBOASR2S EXEC PGM=BBOASR,REGION=0M,TIME=NOLIMIT,
          XX PARM='/ &PARMS &IWMSSNM'
          XX*PARM='&LEPARM / &PARMS &IWMSSNM'
          IEFC653I SUBSTITUTION JCL - PGM=BBOASR,REGION=0M,TIME=NOLIMIT,
          PARM='/ -ORBsrvname BBOASR2A'
          7 XXSTEPLIB DD DISP=SHR,DSN=WEB.CB4XP&WASPTF..SBBOULIB
          IEFC653I SUBSTITUTION JCL - DISP=SHR,DSN=WEB.CB4XP011.SBBOULIB
          8 XX DD DISP=SHR,DSN=EPFDB2.DB2710.SDSNEXIT
          9 XXBBOENV DD PATH='&CBCONFIG/&RELPATH/&SYSPLEX/&IWMSSNM/current.env'
          IEFC653I SUBSTITUTION JCL -
          PATH='/WebSphere390/CB390/controlinfo/envfile/WEBDB2/
          BBOASR2A/current.env'
          10 XXCEEDUMP DD SYSOUT=*
          11 XXSYSOUT DD SYSOUT=*
          12 XXSYSPRINT DD SYSOUT=*
```

```

13 XX
STMT NO. MESSAGE
  2 IEF001I PROCEDURE BBOASR2S WAS EXPANDED USING SYSTEM LIBRARY
    SYS1.PROCLIB
IEF236I ALLOC. FOR BBOASR2S BBOASR2S
IEF237I 9A20 ALLOCATED TO STEPLIB
IEF237I 9921 ALLOCATED TO
IGD103I SMS ALLOCATED TO DDNAME BBOENV
IEF237I JES2 ALLOCATED TO CEEDUMP
IEF237I JES2 ALLOCATED TO SYSOUT
IEF237I JES2 ALLOCATED TO SYSPRINT
BBOU0025I ERRORS WILL BE WRITTEN TO CERR FOR JOB BBOASR2S.
IGD103I SMS ALLOCATED TO DDNAME SYS00001
IGD104I HFS FILE WAS RETAINED, DDNAME IS (SYS00001)
FILENAME IS (/etc/resolv.conf)
IEF237I 9914 ALLOCATED TO SYS00003
IEF285I   TCPIP.STANDARD.TCPXLBIN                               KEPT
IEF285I   VOL SER NOS= USRWB1.
BBOU0245I CURRENT CB SERVICE LEVEL IS build level W401400 release
cb401_serv date 09/03/02 17:39:29.
BBOU0119I CURRENT LOCAL CB SERIES CONFIGURATION DATA:
BBOU0129I           BBOLANG: ENUS.
BBOU0129I           CBCONFIG: /WebSphere390/CB390.
BBOU0129I           CLASSPATH:
/usr/lpp/WebSphere40/lib/ws390srt.jar:/usr/lpp/WebSphere40/lib/xerces.
jar:/usr/lpp/WebSphere40/lib/waswebc.jar:/usr/lpp/WebSphere40/lib/bboa
xrt.jar:/usr/lpp/db2/db2/db2710/classes/db2j2classes.zip:/usr/lpp/db2/
db2/db2710/classes/converters.jar:/u/arti/ibmejbtools.jar:.
BBOU0129I           CLIENT_DCE_QOP: NOT SET, DEFAULT=NO_PROTECTION.
BBOU0129I           CLIENT_HOSTNAME: NOT SET.
BBOU0129I           CLIENTLOGSTREAMNAME: CERR.
BBOU0129I           CLIENT_TIMEOUT: NOT SET, DEFAULT=0.
BBOU0129I           CLIENT_RESOLVE_IPNAME: NOT SET.
BBOU0129I           CONFIGURED_SYSTEM: WEBDB2.
BBOU0129I           DAEMON_IPNAME: EFRWY1.PDL.POK.IBM.COM.
BBOU0129I           DAEMON_PORT: 5555.
BBOU0129I           DAEMON_SSL_PORT: 5556.
BBOU0129I           DATA.CTRLHOST: NOT SET.
BBOU0129I           DATA.CTRLPORT: NOT SET, DEFAULT=5000.
BBOU0129I           DATASHARING: 1.
BBOU0129I           DEFAULT_UNAUTH_CLIENT_ID: NOT SET, DEFAULT=CBGUEST.
BBOU0129I           DM_GENERIC_SERVER_NAME: CBDAEMON.
BBOU0129I           DM_SPECIFIC_SERVER_NAME: DAEMON01.
BBOU0129I           HOME: NOT SET.
BBOU0129I           ICU_DATA: /usr/lpp/WebSphere40/bin.
BBOU0129I           IIOP_SERVER_SESSION_KEEPALIVE: NOT SET.
BBOU0129I           IR_GENERIC_SERVER_NAME: CBINTFRP.
BBOU0129I           IR_SPECIFIC_SERVER_NAME: INTFRP01.
BBOU0129I           IRPROC: BBOIR.
BBOU0129I           IVB_DEBUG_ENABLED: NOT SET.
BBOU0129I           IVB_DRIVER_PATH: /usr/lpp/WebSphere40.
BBOU0129I           IVB_HOME: NOT SET.
BBOU0129I           JAVA_COMPILER: NOT SET.
BBOU0129I           JVM_DEBUG: NOT SET.
BBOU0129I           JVM_DEBUG_PORT: NOT SET.

```

```

BBOU0129I          JVM_HEAPSIZE: NOT SET, DEFAULT=256.
BBOU0129I          LDAPBINDPW: *****.
BBOU0129I          LDAPCONF:
/WebSphere390/CB390/WEBDB2/etc/ldap/WEBDB2.bboslapd.conf.
BBOU0129I          LDAPHOSTNAME: NOT SET.
BBOU0129I          LDAPIRBINDPW: *****.
BBOU0129I          LDAPIRCONF:
/WebSphere390/CB390/WEBDB2/etc/ldap/WEBDB2.bboslapd.conf.
BBOU0129I          LDAPIRHOSTNAME: NOT SET.
BBOU0129I          LDAPIRNAME: NOT SET.
BBOU0129I          LDAPIRROOT: o=BOSS,c=US.
BBOU0129I          LDAPNAME: NOT SET.
BBOU0129I          LDAPROOT: o=BOSS,c=US.
BBOU0129I          LIBPATH:
/usr/lpp/db2/db2/db2710/lib:/usr/lpp/java13_40/IBM/J1.3/bin:/usr/lpp/j
ava13_40/IBM/J1.3/bin/classic:/usr/lpp/WebSphere40/lib:/usr/lib:/usr/t
ools/jprof.
BBOU0129I          LOGSTREAMNAME: BBO.BOSSFR13
BBOU0129I          MIN_SRS: 2.
BBOU0129I          NM_GENERIC_SERVER_NAME: CBNAMING.
BBOU0129I          NM_SPECIFIC_SERVER_NAME: NAMING01.
BBOU0129I          NMPROC: BBONM.
BBOU0129I          OTS_DEFAULT_TIMEOUT: 3200.
BBOU0129I          OTS_MAXIMUM_TIMEOUT: 3200.
BBOU0129I          PATH: NOT SET.
BBOU0129I          RAS_MINORCODEDEFAULT: NODIAGNOSTICDATA.
BBOU0129I          RECOVERY_TIMEOUT: NOT SET, DEFAULT=15.
BBOU0129I          REM_DCEPASSWORD: NOT SET.
BBOU0129I          REM_DCEPRINCIPAL: NOT SET.
BBOU0129I          REM_PASSWORD: *****.
BBOU0129I          REM_USERID: CBGUEST.
BBOU0129I          RESOLVE_IPNAME: EFRWY1.PDL.POK.IBM.COM.
BBOU0129I          RESOLVE_PORT: 900.
BBOU0129I          SM_DEFAULT_ADMIN: CBADMIN.
BBOU0129I          SM_GENERIC_SERVER_NAME: CBSYSMGT.
BBOU0129I          SM_SPECIFIC_SERVER_NAME: SYSMGT01.
BBOU0129I          SMPROC: BBOSMS.
BBOU0129I          SOMOOSQL: 1.
BBOU0129I          SRVIPADDR: None.
BBOU0129I          SSLIOP_SERVER_SESSION_KEEPALIVE: NOT SET.
BBOU0129I          SYS_DB2_SUB_SYSTEM_NAME: DBEP.
BBOU0129I          TRACEALL: 1.
BBOU0129I          TRACEBASIC: NOT SET.
BBOU0129I          TRACEBUFFCOUNT: NOT SET.
BBOU0129I          TRACEBUFFLOC: BUFFER.
BBOU0129I          TRACEBUFFSIZE: NOT SET.
BBOU0129I          TRACEDetail: NOT SET.
BBOU0129I          TRACEMINORCODE: NOT SET.
BBOU0129I          TRACEPARAM: 00.
BBOU0129I          TRACESPECIFIC: NOT SET.
BBOU0589I          CURRENT SERVER SETTINGS:
BBOU0590I          Server name: BBOASR2.
BBOU0590I          Control region identity: CBACRU2.
BBOU0590I          Control region stack size (in bytes): 0.
BBOU0590I          Control region thread pool size: 25.

```

```

BBOU0590I          Server region identity: CBASRU2.
BBOU0590I          Server region stack size (in bytes): 0.
BBOU0590I          Server region thread pool size: 3.
BBOU0590I          Production server: YES.
BBOU0590I          Enable Setting OS Thread Identity to RunAs Identity: NOT
SET.
BBOU0590I          Debugger allowed: NO.
BBOU0590I          Isolation policy: Multiple
transactions per server region.
BBOU0590I          Replication policy: Replicate as needed.
BBOU0590I          Server region requires JVM: NO.
BBOU0590I          Local identity: CBGUEST.
BBOU0590I          Remote identity: CBGUEST.
BBOU0590I          Register transaction factory: NO.
BBOU0590I Allow server region garbage collection: YES.
BBOU0590I          Garbage collection interval: 15000000.
BBOU0590I          Logstream name: BBO.BOSSF13
.
BBOU0590I          ConfiguredPort: 0.
BBOU0590I          SSLFirewallPort: 0.
BBOU0590I          SSLKeyring: CBKeyring.
BBOU0590I          HTTPListenerPort: 0.
BBOU0590I          HTTPSListenerPort: 0.
BBOU0590I          Control region proc name: BBOASR2.
BBOU0590I          Allow non-authenticated clients: YES.
BBOU0590I          Function level: 2.
BBOU0025I ERRORS WILL BE WRITTEN TO BBO.BOSSF13 LOG STREAM FOR JOB
BBOASR2S.

```

Locations of executable programs

The next thing to review in the configuration is where your program code is located. IBM recommends that you install as much of the WebSphere for z/OS code itself in LPA as is reasonable, and the remainder in the linklist. This ensures that you have eliminated any unnecessary steplibs which can adversely affect performance. Refer to “z/OS UNIX System Services” on page 84 for tuning tips for USS HFS tuning considerations. There is a performance cost to doing this, however. You should ensure that only server regions have visibility to your application code. The control region usually runs with no steplibs since all the code required is located in system locations. Verify that the STEPLIB DD in the control region and sever region procs do not point to anything unnecessary.

If you choose to not put most of the runtime in LPA, you may find that your paging subsystem gets a bigger workout as the load increases. At a minimum, WebSphere for z/OS will start seven address spaces so that any code that is not shared will load seven copies rather than one. As the load increases, many more server regions may start and will contribute additional load on the paging subsystem.

Review the PATH statement to ensure that only required programs are in the PATH and that the order of the PATH places frequently-referenced programs in the front. If you are using Java, refer to “Tuning tips for JVM” on page 99.

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.

Authentication

You have several options when dealing with authentication:

- **Local authentication:** Local authentication 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. We can't comment on specifics yet, but a starting point is to follow the configuration options for the Webserver's SSL.

Note: For more information on security, see *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834.

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 transport

The HTTP transport is an option on 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 transport. For more information, see *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836.

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 4 on page 109. 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 4. Having applications on a single system avoids calls from one system to another

One tran or multiple trans?

You also have a choice of running server regions with an isolation policy of one tran per server region or multiple trans per server region. From a performance perspective, running more threads in a server region will consume less memory but at the cost of thread contention. This contention is application-dependent. We generally recommend the use of multiple trans unless you run into contention problems.

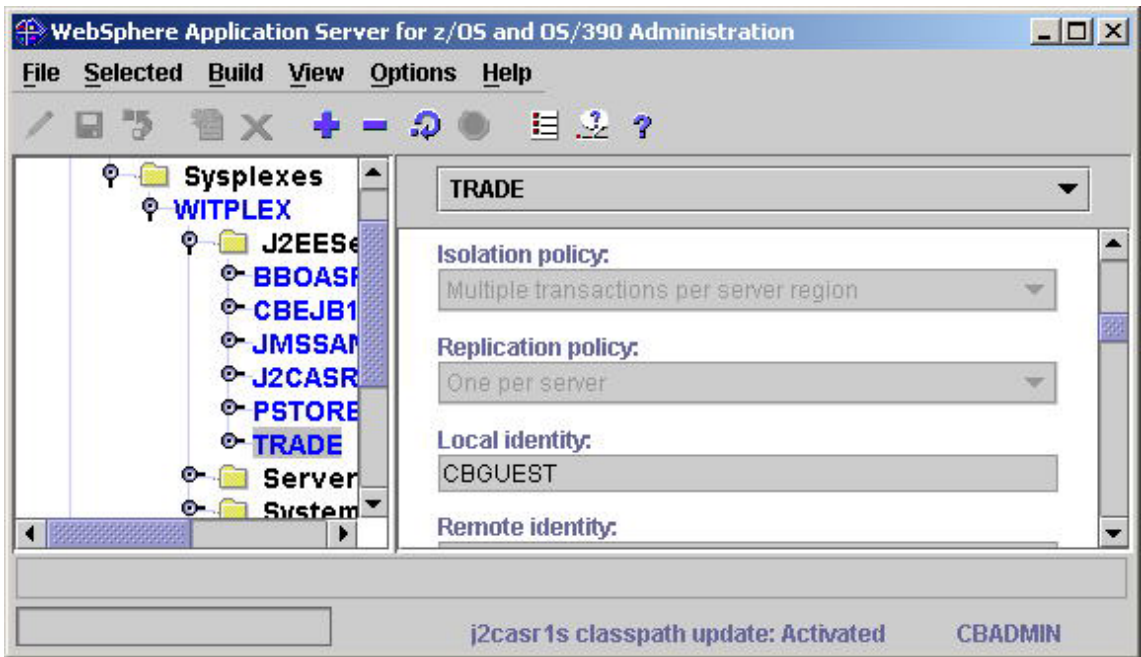


Figure 5. Setting the isolation policy

Setting the number of threads: Specify the threads setting using the `BB00_WORKLOAD_PROFILE` option in the `current.env` file. The variables include:

- NORMAL - the default which results in a value of 3.
- CPUBOUND
- IOBOUND
- LONGWAIT

The thread value increases with each variable to the maximum number available with the LONGWAIT setting.

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. 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 Transport

It is more efficient to have the HTTP Transport installed on the WAS 4.x runtime application server instance. With the WAS 4.1 September refresh server side SSL is now supported, but other restrictions might apply.

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 V4.0.1 for z/OS and OS/390, 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.

Component management

The parameters that specify how containers manage components are control by three mechanisms:

- Application Assembly tool
- Java virtual machine properties file (`jvm.properties`)
- Java Database Connectivity (JDBC) parameters (`db2sqljdbc.properties`)

This section defines some recommended settings for these parameters.

Application Assembly tool: The IBM deployment descriptor extensions provide additional, proprietary, policy meta-data for J2EE application components to be deployed on WebSphere servers. Use the Application Assembly tool (AAT) to change the parameters of the IBM deployment descriptor extensions. For more information about these extensions, see *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836

Tips: :

- Transaction attributes — selects transaction attributes defined in EJB 1.1 specification to each method of the EJBs.
 - TX_NOT_SUPPORTED recommended

| **Note:** By default the AAT defines the attribute to TX_REQUIRED. If a
| method doesn't need transactional behavior like DB updates,
| TX_NOT_SUPPORTED or TX_SUPPORTS is recommended to
| reduce overhead of transaction.

- **ReadOnly** — reduces ejbStore operation at the end of transaction in a method. If you mark a method ReadOnly, data will not be written to DB2 at the end of the transaction.

| **Note:** This parameter is used for methods in a CMP.

- **Transaction isolation** — defines how a method in an EJB locks DB2.
 - TRANSACTION_NONE is recommended for methods that perform queries.
 - TRANSACTION_REPEATABLE_READ is used for methods that perform updates.

| **Note:** We recommend TRANSACTION_NONE for methods that don't have
| update operations.

- **Activation and load policy** — the combination of these values specify commit time options that are defined in the EJB 1.1 specification. They define the disposition of the instance state at transaction commit time.
 - Activation policy: AtTran
 - Load policy: AtActivation

| With these settings, the replicate Entity Beans are created for each
| transaction. Data integrity is managed by the database. The beans are
| passivated after the transactions end.

| **Note:** These parameters should be considered when entity beans are used.

- **Bean pool size** — the maximum and minimum number of bean pool that the container can use for each bean type.
 - Min: 10
 - Max: 250
- **Connection management policy** — specifies how and when the connection that is used by a J2EE component is reclaimed.
 - Aggressive mode recommended

| The parameter is defined for each resources. There are two values:
| normal and aggressive. In aggressive mode, the connections that are
| used by J2EE components are reclaimed even through the component
| instances still exist in the container as pool. Another reason to use
| aggressive mode is to prevent connection shortages that result from
| premature applications that don't close connections correctly at the end
| of logic. In aggressive mode the application is able to reclaim the

suspended connections in the pooled instances by containers. So, the parameter is especially useful in the implementation phase of applications.

Table 20. Connection management policy setting effect on bean connection reclamation

Beans	Normal mode	Aggressive Mode
Stateless	Reclaim connections on EJB Remove (The instance doesn't exist in the container).	Reclaim connections between business methods (The instance is pooled in the container).
Stateful	Reclaim connections on EJB Remove (The instance doesn't exist in the container).	Reclaim connections on EJB Remove and Passivate (The instance is temporary serialized.)
Entity	Reclaim connections on UnSetEntityContext (The instance doesn't exist in the container).	Reclaim connections on EJB Remove and Passivate (The instance is pooled in the container).

Java virtual machine properties file (jvm.properties): The jvm.properties file changes the default settings for the JVM that runs in a WebSphere for z/OS J2EE or MOFW server.

Tips:

- com.ibm.CORBA.iiop.noLocalCopies — reduces the overhead of Java serialization by enabling object by reference for interaction between components rather than the WebSphere Application Server for OS/390 default of object by value.
 - com.ibm.CORBA.iiop.noLocalCopies=true

Note: This parameter is effective when all components in an application reside in one J2EE server. This function is not J2EE compliant.

- com.ibm.ws390.ConnectionUsageScopeDefault — enables the connection pools of backend system
 - com.ibm.ws390.ConnectionUsageScopeDefault=SeriallyReusable

Note: You should also define bborbrt.jar in the classpath of the current.env for the J2EE server.

Java Database Connectivity (JDBC) parameters: These parameters specify the characteristics of JDBC connections.**Tips:**

- db2.connpool.max.size — specifies the maximum number of concurrent physical connections (DB2 threads) that the driver maintains.
 - db2.connpool.max.size=500

Note: Need to adjust with the DB2 threads parameters.

- `db2.connpool.idle.timeout` — specifies the minimum number of seconds that an unused physical connection remains in the connection pool before the thread is closed.
 - `db2.connpool.idle.timeout=600` (default)
- `db2.connpool.connect.create.timeout` — specifies the timeout value to create and get a new connection.
 - `db2.connpool.connect.create.timeout=0` (default)
- `DB2SQLJMULTICONTEXT` — specifies whether each connection in an application is independent of other connections in the application, and each connection is a separate unit of work, with its own commit scope.
 - `DB2SQLJMULTICONTEXT=yes`

Note: This parameter is valid when RRSAF is used for connection.

- `DB2CURSORHOLD` — specifies whether the cursor in a ResultSet is destroyed when the transaction is committed.
 - `DB2CURSORHOLD=no`

Note: It is likely that `DB2CURSORHOLD=NO` reduces the number of interactions between DB2 and the client. It would be effective when J2EE applications access remote DB2.

Container Managed Persistence (CMP) Connection and Prepared Statement Pooling

Overview of Container Managed Persistence (CMP) Connection and Prepared Statement Pooling: WebSphere Container Managed Persistence (CMP) Connection and Prepared Statement Pooling, aids in the management of CMP Bean performance. It does so by pooling and reusing both JDBC connections and prepared statements, and allowing reliable across-transaction pooling. The two-tier structure pools prepared statements with their corresponding connections. This in turn is associated with a thread (across transactions) or transaction (within a transaction).

In order for a CMP Bean to persist its data, the EJB container must utilize a database connection and a set of prepared statements. The database to which it connects may be accessed several times, so it is redundant to create a new connection and a new set of prepared statements with each access.

This section will give you a general introduction to managing performance with CMP Connection Pooling and discuss the two pooling methods.

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 *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836 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 88.
2. Use Prepare statement to allow dynamic statement cache of DB2 on z/OS.
3. Don't include literals in the prepare statement, 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.

SOAP tuning tips

Tips: The Simple Object Access Protocol (SOAP) is a lightweight protocol which provides a mechanism to use XML for exchanging structured and typed information between peers in a decentralized, distributed environment. The following tips may help you tune the SOAP environment:

1. Replace the default xml parser shipped with WAS 4.0.x if are using a version of WebSphere prior to 4.0.1 September refresh
You can use the xerces 1.4.4 level of the parser from the xml.apache.org web site. You must then set the property `com.ibm.ws390.wc.SERVICE.classpath=<fully qualified path to the xerces.jar>`(for example, `com.ibm.ws390.wc.SERVICE.classpath=/shared/local/lib/xerces_1_4_4.jar`
Do not remove the `soap.jar` unless you are replacing it and do not include `.jar` files containing parsers in the `.war` file containing a web app or the `.ear` file containing the J2EE application if you are replacing the parser as in this step. You will avoid a number of class/method mismatch problems.
2. Specify `noLocalCopies`. This will allow passing of parameters by reference instead of by value. This only applies if you are creating a webservice over an EJB. Specify `com.ibm.CORBA.iiop.noLocalCopies=true` in the `jvm.properties` file for your server.
3. Define your WebSphere server as a production Server, otherwise the Server Region address space will be configured with a single thread (TCB).
Check the box labeled `ProductionServer` when creating your server with the Administration application.
4. Specify your isolation policy as multiple transactions per server.
You can set this property for servers when defining them with the Administration application.
5. Make certain that Object Level Trace is disabled.
To do this, ensure that "Debugger allowed" is not checked in the Administration application server parameters.
6. If using EJBs, make sure EJB activation policies are NOT specified as 'once'. This would prevent Beans from being deployed in multiple server regions.

Note: The SOAP IVP `.ear` file contains this specification, but it is not required for SOAP applications.

7. Ensure that the Application Assembly tool transaction attribute is set to TX_NOT_SUPPORTED.
The Application Assembly tool defaults to a transaction attribute of TX_REQUIRED. This introduces additional overhead and possible synchronization bottlenecks.
8. SOAP applications will perform better with smaller SOAP messages containing fewer XML elements and especially fewer XML attributes. Avoid passing empty attributes or empty elements in SOAP messages. Do not include unneeded data in SOAP messages. The contents of SOAP messages must be serialized and parsed. These are expensive operations and should be minimized.
9. You may need to increase the default java heap size.
SOAP and XML (DOM) are storage-intensive and small heap sizes may result in excessive java garbage collection. We found a heapsize of 256M (the default) was optimal for most test cases in the laboratory. You can monitor garbage collection using the verbosegc directive. Use the JVM_HEAPSIZE and JVM_MINHEAPSIZE parameters in the current.env file to specify Heapsize.
10. Make certain the TCP/IP send/receive buffers are large enough to hold the bulk of the xml messages that will be sent.
11. Do not use the TestClient.jsp generated by WSAD do do stress testing.
The proxy class it calls has all methods synchronized. The TestClient HTTP flows back into the server which is a fatal problem if MAX_SRS is set. You will encounter random errors if the server has multiple SRs associated with it due to the way the EndPointURL is set.
12. Consider using a Message Model rather than RPC model. It provides complete control over the format of the XML but requires additional programming effort.
13. When using RPC-style messages, try to send strings if possible.
14. Consider writing your own serializers and deserializers, avoiding reflection.
15. Consider writing a servlet to use a SAX parser rather than the SOAP Runtime if you require improved performance.

Chapter 10. WebSphere for z/OS performance monitoring procedures

This chapter describes general WebSphere for z/OS performance monitoring procedures for J2EE applications.

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: Please see Chapter 11, “Systems Management Facility (SMF) recording and monitoring” on page 133 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 85.

Steps for diagnosing a performance problem

WebSphere for z/OS is a large and complex application server. It relies on many different subsystems and components to accomplish its job. In order to get maximum performance from WebSphere for z/OS, it is important that each of these services be tuned to avoid unnecessary overhead. Perform the following steps to determine if you have a performance problem and to figure out the source of the problem.

1. Determine if you have a performance problem. Have you observed, for example, poor throughput or poor response time?

If . . .	Then . . .	Notes
Yes	Go to step #2	Check job log for CEE dumps, check for SVC dumps, etc. Figure out if the issue is underutilization, overutilization, overhead when unloaded, overhead when loaded, storage footprint.
No	You're done	

2. Is this a new WebSphere for z/OS installation?

If . . .	Then . . .	Notes
Yes	Go to step #3	
No	Go to step #4	

3. Have you read Chapter 9, "WebSphere for z/OS performance tuning guidelines" on page 73 and implemented the performance recommendations?

If . . .	Then . . .	Notes
Yes	Go to step #4	
No	Read Chapter 9, "WebSphere for z/OS performance tuning guidelines" on page 73 and implement the performance recommendations.	Once you have implemented the performance tuning recommendations, begin at step #1 again.

4. Have you checked to see if component tracing has been turned off?

If . . .	Then . . .	Notes
Yes	Go to step #5	

If . . .	Then . . .	Notes
No	a. On the MVS console use the following command to display the component traces: D TRACE,ALL b. Check that all traces are either OFF or MIN. c. If a particular component trace is ON, enter TRACE CT,OFF,COMP=xxxxxxx where xxxxxx is the name of the component.	If this solves the problem, you are done. If not, continue with step #5.

5. Have you checked key fields in your server region job log to see if WebSphere tracing has been turned off? Refer to “Server configuration example” on page 103.

If . . .	Then . . .	Notes
Yes	Go to step #6.	

If . . .	Then . . .	Notes
No	<ul style="list-style-type: none"> a. Look in the joblog of the control or server region b. Ensure that JAVA_COMPILER is not set. c. Ensure that JVM_DEBUG is not set d. Ensure that TRACEALL is set to either 0 or 1 e. If TRACEALL is set to 1, make sure TRACEBUFFLOC is set to BUFFER. f. Ensure that TRACEBASIC and TRACEDetail are not set. g. Make sure Production server is set to YES, and Debugger allowed is set to NO. 	If this solves the problem, you are done. If not, continue with step #6.

6. Have you checked to see if there are CEE dumps or SVC dumps in the server region? There will be a performance degradation if dumps are occurring.

If . . .	Then . . .	Notes
Yes	Go to step #7.	
No	<ul style="list-style-type: none"> a. Check the joblog of the server region for CEE dumps or SVC dumps. 	<p>If you are taking dumps, work with the application developers and the system center to resolve.</p> <p>Continue with Step #7.</p>

7. Have you checked to see if JRAS tracing is disabled?

If . . .	Then . . .	Notes
Yes	Go to step #8.	

If . . .	Then . . .	Notes
No	a. In the trace.dat file pointed to by the JVM properties file, look for the following: <pre>com.ibm.ejs.*=all=(enable/disable) com.ibm.ws390.orb=all=(enable/disable)</pre> b. Ensure that both of these are set to disable or that the two lines are deleted.	If this solves the problem, you are done. If not, continue with step #8.

8. Have you checked to see if a Java memory leak or a problem with Java garbage collection?

If . . .	Then . . .	Notes
Yes	Go to “Collecting performance diagnosis information”.	
No	a. Go to “Garbage collection and JVM_HEAPSIZE” on page 76 b. If you observe that the % of free space is declining or that you are spending more than 2% of the time in garbage collection, increase your Java heap size.	If this solves the problem, you are done. If not, continue with “Collecting performance diagnosis information”

Collecting performance diagnosis information

When you suspect that you have a performance problem in the runtime of WebSphere for z/OS, there are three reports we need to help you pinpoint the source of the problem:

- RMF report
- Workload activity (with the WLM service class definitions)
- Joblog for the server region

It is very important that these reports are generated during an interval in which the workload with the performance problem is running in a steady state. There should be no initialization or termination activities during the interval.

Steps for capturing RMF data

Before you begin:

1. Switch SMF datasets to ensure a clean SMF dataset for the run. To change SMF data sets, enter the following command on the MVS console:

```
I SMF
```

This will switch SMF datasets. You should see a message similar to this:

```
IEE388I SMF NOW RECORDING ON VOLSER SMFP71, DSN=SYS1.SP7.MANE
```

The important information in this message is the MAN data set that is now being used. In this example, it is the "E" SMF data set. You will need this information for the post-processing of the SMF data.

2. When the workload is running in a stable state, try to get the CPU percentage as high as you can.

Perform the following steps to generate an RMF report.

1. To start RMF, enter the following command on the MVS console:

```
S RMFXA.A  
R xxx,G0      [where xxx is the RMF reply number]
```

You should have a member of one of your PROCLIBs named RMFXA. It should look something like this:

```
1 //IEFPROC EXEC PGM=ERBFMFVC,REGION=0M,DPRTY=(10,10),  
2 // PARM='MEMBER(XS)',PERFORM=74,TIME=1440  
3 //IEFRDOR DD DSN=USER.PARMLIB,DISP=SHR  
4 //IEFPARM DD DSN=USER.PARMLIB,DISP=SHR  
5 //SYSABEND DD SYSOUT=A  
6 //MFMESSAGE DD SYSOUT=H
```

On line 2 in the example above, PARM='MEMBER(XS)' refers to the PARMLIB member which will control your RMF environment. The PARMLIB member will be named: ERBRMFxx, where xx is replaced with the 2 characters in the above MEMBER statement. In this case, the characters are "XS".

The following is a copy of the PARMLIB member ERBRMFXS: (parameters 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 */
|         REPORT(REALTIME)         /* REALTIME REPORTS YES     */
|         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)

```

Notice the parameter SYNC(SMF). This will allow RMF to be synchronized with SMF.

-
2. To check your active SMF member, enter the following command on the MVS console:

```
D SMF,0
```

This command will return a lot of information including your active SMF member.

```
MEMBER = SMFPRMP5
```

In this example, the active PARMLIB member SMFPRMP5 controls the SMF collection. The following is the contents of PARMLIB member SMFPRMP5.

```

| ACTIVE,
| DSNAME(SYS1.S&SYSCZONE..MANA,
| SYS1.S&SYSCZONE..MANB,
| SYS1.S&SYSCZONE..MANC,
| SYS1.S&SYSCZONE..MAND,
| SYS1.S&SYSCZONE..MANE,
| SYS1.S&SYSCZONE..MANF,
| SYS1.S&SYSCZONE..MANG,
| SYS1.S&SYSCZONE..MANH,
| SYS1.S&SYSCZONE..MANI,
| SYS1.S&SYSCZONE..MANJ),

```

```

NOPROMPT,
REC(PERM),
MAXDORM(3000),
STATUS(010000),
JWT(0510),

SID(SP5,SYSNAME(SP5)),
SID(SP6,SYSNAME(SP6)),
SID(SP7,SYSNAME(SP7)),
LISTDSN,                                /* SHOW THE DATASET NAMES */
SYS(TYPE(0,2:10,14,15,20,22:24,26,    /* ALL RECORDS IN THIS LIST */
30,32:34,40,42,47:48,58,64,70:83,    /* WERE BEING COLLECTED BY */
84,88,89,90,                          /* SYSTEM TEST. ALL WERE LEFT */
110,127:134,148:151,161,             /* AS IS. 100:102 ARE FOR DB2 */
244,245,253)),
INTVAL(05),
SYNCVAL(05),
SYS(DETAIL,INTERVAL(003000)),
SYS(EXITS(IEFACTRT,IEFUJI,IEFU83,IEFU84,IEFU85,IEFUJV,IEFUSI)),
SUBSYS(STC,EXITS(IEFU83,IEFU84,IEFU85,IEFUSI),
INTERVAL(SMF,SYNC))

```

Notice the parameters INTVAL(05), SYNCVAL(05), and INTERVAL(SMF,SYNC). These settings cause an RMF report to be cut every 5 minutes on the 5 minute boundary (0,5,10,15,20,25,30,35,40,45,50,55) after the hour.

The following parameters from the above definition determine which SMF records will be collected:

```

SYS(TYPE(0,2:10,14,15,20,22:24,26,
30,32:34,40,42,47:48,58,64,70:83,
84,88,89,90,
110,127:134,148:151,161,
244,245,253)),

```

-
3. To stop RMF, enter the following command on the MVS console:

```
P A
```

The output will go to the output queue specified in the SYSOUT parameter in the ERBRMF PARMLIB member. In this example, the output queue specified is the hold queue SYSOUT(H).

-
4. Use the SDSF print open dataset function to copy the report to a dataset.
-

When you are done, your RMF reports are in a dataset.

You should check the following RMF reports:

- Direct Access Device Activity
- Enqueue Activity
- Paging Activity

For information on interpreting these reports, consult Resource Measurement Facility Report Analysis, SC33-7991.

Steps for capturing a workload activity report

Before you begin: Make sure that the workload is running in a stable state.

Perform the following steps to capture a workload activity report.

1. Submit a JCL job with the following statements:

```
//SMFSTAGE JOB MSGLEVEL=1,CLASS=A,MSGCLASS=H
//STEP1 EXEC PGM=IFASMFDP
//DUMPIN DD DSN=SYS1.SP7.MANE,DISP=SHR
//DUMPOUT DD DSN=WALKER.BOSS.SMF.STAGING,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
        INDD(DUMPIN,OPTIONS(DUMP))
        OUTDD(DUMPOUT,TYPE(000:255))
//*****
//RMFSORT EXEC PGM=SORT
//EXITLIB DD DISP=SHR,DSN=SYS1.SERBLINK
//SORTIN DD DSN=WALKER.BOSS.SMF.STAGING,
//        DISP=SHR
//SORTOUT DD DISP=(,PASS,DELETE),UNIT=SYSDA,SPACE=(CYL,(500,100)),
//        DSN=&&SMFSORT//SORTWK01 DD
//        DISP=(NEW,DELETE),SPACE=(CYL,(35)),UNIT=SYSALLDA
//SORTWK02 DD DISP=(NEW,DELETE),SPACE=(CYL,(35)),UNIT=SYSALLDA
//SORTWK03 DD DISP=(NEW,DELETE),SPACE=(CYL,(35)),UNIT=SYSALLDA
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD *
        SORT FIELDS=(11,4,CH,A,7,4,CH,A),EQUALS
        MODS E15=(ERBPPE15,500,EXITLIB,N),
        E35=(ERBPPE35,500,EXITLIB,N)
//*
//* RMF POSTPROCESSING
//*
//*****
//* THIS STEP WILL PRODUCE A M1 WORKLOAD ACTIVITY REPORT IN GOAL MODE *
//* OR COMPAT MODE, DEPENDING ON THE INPUT RECORDS *
//*****
//WKLD@PGP EXEC PGM=ERBRMFPP,TIME=1440
//STEPLIB DD DISP=SHR,DSNAME=SYS1.SERBLINK
//MR400001 DD SYSOUT=*
//MFPMSGDS DD SYSOUT=*
//MFPINPUT DD DSN=&&SMFSORT,DISP=(OLD,DELETE)
//*SYSRPTS(WLMGL(SCLASS(TSO*),SCPER))
//SYSIN DD *
```

```

SYSRPTS(WLMGL(POLPAG,WGPER,SCLASS,RCLASS,SCPER,POLICY,WGROUP))
REPORTS(ALL)
SYSOUT(H)
/*

```

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. Use the SDSF print open dataset function to copy the report to a dataset.
-

When you are done, you have a workload activity report in a dataset.

Collecting additional performance diagnosis information

- 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, we 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 — refer to .
 - Jinsight — if you want to profile your application, you can download the Jinsight tool from the IBM alphaWorks web site at:
<http://www.alphaworks.ibm.com/tech/jinsight>

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 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.

Note: If you are looking for the System Management Facility Browser for z/OS and OS/390 source code, remember that it will not be shipped within the other WebSphere for z/OS samples. The source and the compiled version will be shipped via the WebSphere for z/OS Web site at <http://www.ibm.com/software/webservers/appserv/>.

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.

Eight records can be produced: the *Server Activity record*, *Container Activity record*, *Server Interval record*, the *Container 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 *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838.

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 server definition of the Systems Management User Interface by checking the checkbox: "Write Server Activity SMF Records."

Container activity record

The purpose of the container activity SMF record is to record activity that is running inside a container 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 inside a container located in a WebSphere transaction server. If the activity runs in multiple servers, then multiple records are written for the activity.

You can activate this record through the server definition of the Systems Management User Interface by checking the checkbox: "Write Container Activity SMF Records."

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 server definition of the Systems Management User Interface by checking the checkbox: "Write Server Interval SMF Records."

Container interval record

The purpose of the container interval SMF record is to record activity that is running inside a container located inside the WebSphere transaction server. This record is produced at regular intervals and is an aggregate of the activities running inside a 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 container located in a WebSphere transaction server within the interval being recorded. If there is more than one server instance associated with a server, there will be a record for the container from each server instance. To get a common view of the work running in the container during the interval, you must merge the records after processing.

You can activate this record through the server definition of the Systems Management User Interface by checking the checkbox: "Write Container Interval SMF Records."

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 J2EE server definition of the Systems Management User Interface by checking the checkbox titled "Write Container Activity SMF Records."

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 activate this record through the J2EE server definition of the Systems Management User Interface by checking the checkbox titled "Write Container Interval SMF Records."

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

Enabling SMF recording for the WebSphere Application Server can be achieved by following the steps outline below.

1. Using the Server Definition of the System Management User Interface Administrator application
2. Editing the SMFPRMxx parmlib member

Steps for using the System Management User Interface Administrator application

Before you begin: Ensure that you have proper access to the Administration application.

Follow these steps to enable SMF recording:

1. Open the Administration application and start a new conversation.

2. Select the server for which you want to enable SMF recording.

3. Select one or more check boxes for SMF recording. The available choices are:
 - **Server activity record**
Check the checkbox titled "Write Server Activity SMF Records."

- **Container activity record**
Check the checkbox titled "Write Container Activity SMF Records."
- **Server interval record**
Check the checkbox titled "Write Server Interval SMF Records."
- **Container interval record**
Check the checkbox titled "Write Container Interval SMF Records."
- **J2EE Container activity record**
Check the checkbox titled "Write Container Activity SMF Records" in the J2EE Server properties definition.
- **J2EE Container interval record**
Check the checkbox titled "Write Container Interval SMF Records" in the J2EE Server properties definition.
- **WebContainer activity record**
WebContainer activity records are automatically enabled with the enabling of J2EE Container activity records.
- **WebContainer interval record**
WebContainer interval records are automatically enabled with the enabling of J2EE Container interval records.

-
4. Set the interval length.
-
5. Save the server definition.
-
6. Validate the new conversation.
-
7. Mark all tasks complete.
-
8. Activate the new conversation.
-

The steps for using the System Management User Interface Administration application are completed when the the new conversation is successfully activated.

For additional information on enabling SMF recording for WebSphere Application Server through the Server Definition of the System Management User Interface Administrator application (see *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface, SA22-7838*).

Steps for using a DB2 SPUFI script to turn SMF recording on and off

Normally, you must use the Systems Management Administration application to change the SMF recording options of WebSphere for z/OS servers. Below is a SPUFI script to easily turn on or off SMF recording for the server/container activity/interval records.

Follow these steps to enable SMF recording:

1. Edit the following example script. This example turns Container Interval SMF recording ON for the WSIVP2 servers assuming the active conversation is 115:

```
UPDATE BBO.BB0MT06_ACT_SG
  SET WR_CON_INT_SMF_REC = X'01'
  WHERE SERVER_GRP_NAME = 'WSIVP2'
  AND SERVER_GRP_LEVEL = 115;
```

-
2. Substitute the appropriate generic server name for 'WSIVP2'.
-
3. Substitute the appropriate SERVER_GRP_LEVEL (or Conversation Level) which is usually the highest number in that column representing the current active conversation. (If there have been new conversations added since the active one, then you will have to figure it out or set all rows in the table for that server.)
-
4. Specify the type of records to be written.
 - To Write Server Activity SMF Records, use column 'WR_SRV_ACT_SMF_REC'
 - To Write Container Activity SMF Records, use column 'WR_CON_ACT_SMF_REC'
 - To Write Server Interval SMF Records, use column 'WR_SRV_INT_SMF_REC'
 - To Write Container Interval SMF Records, use column 'WR_CON_INT_SMF_REC'
-
5. To turn on SMF recording for a specific record, set the column to X'01'. For example, to write server activity SMF Records:
SET WR_SRV_ACT_SMF_REC = X'01'
-
6. To turn off SMF recording for a specific record, set the column to X'00'. For example, to turn off write server activity SMF Records:
SET WR_SRV_ACT_SMF_REC = X'00'

-
7. To display the SMF settings of a particular server (for example, WSIVP2) in all conversations and see what the highest level number is, you can run the script below. (The output has to be viewed in HEX because the fields either contain a X'00' for "no recording" or X'01' for "recording."):

```
SELECT SERVER_GRP_NAME,
       SERVER_GRP_LEVEL,
       WR_SRV_ACT_SMF_REC, WR_CON_ACT_SMF_REC,
       WR_SRV_INT_SMF_REC, WR_CON_INT_SMF_REC,
       SMF_INT_LENGTH
FROM BB0.BB0MT06_ACT_SG WHERE SERVER_GRP_NAME = 'WSIVP2';
```

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 Administration application, 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 SM User Interface
 - 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 by the SM EUI for the server or container definition). The default SMF recording interval is 30 minutes. See *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838, for details.

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.

SMF Record Interpreter for record type 120

The WebSphere for z/OS SMF Record Interpreter is a tool which enables the interpretation of complete output datasets from the IBM z/OS utility program IFASMFDP. It writes a header line for all record types and a detailed dump for record type 120.

Note: See the section “Formatting the output data set” on page 141 about how to use IFASMFDP to extract specified subsets of SMF data from the system SMF datasets into a sequential dataset.

The WebSphere for z/OS SMF Record Interpreter dumps all WebSphere for z/OS-relevant data into a printable output file. It is a Java utility, so it needs to be interpreted and executed by a Java Virtual Machine (JVM) under the z/OS or OS/390 UNIX environment.

Example of SMF Record Interpreter output: Data from the sequential file is produced record by record. Each record contains a number of triplets which are first described in the record’s header section (the first part of a record). The description is then followed by the triplet contents which are presented by the tool in the sequence of their appearance within the record.

Each triplet contains data for a section. Several types of sections are defined, such as:

- ProductSection
- ActivitySection
- CommSessionSection
- etc.

The WebSphere for z/OS SMF Record Interpreter interprets each section in its specific way and prints the interpreted data into the output file.

Note: Some sections may contain subsections that are also organized by the means of triplets.

The printable output of the WebSphere for z/OS SMF Record Interpreter will resemble the following example.

SMF file analysis starts ...

```
-----  
Record #2, Type: 120, Size: 372, Date: Mon Apr 23 09:13:39 EDT 2001  
  SystemID: SY1, SubsystemID: null, Flag: 94  
  Subtype: 1 (SERVER ACTIVITY)  
#Triplets: 3  
Triplet:   offset: 64   length: 32   count: 1  
Triplet:   offset: 96   length: 192  count: 1  
Triplet:   offset: 288  length: 84   count: 1
```

```
Triplet #1
ProductSection
Version: 2, Codeset: IBM-1047
Endian: 1, TimeStampFormat: 1 (S390STCK64)
IndexOfThisRecord: 1, Total#OfRecords: 1, Total#OfTriplets: 3
```

```
Triplet #2
ServerActivitySection
HostName: PLEX1
ServerName: BBOASR1
ServerInstanceName: BBOASR1A
ServerType: MOFW Server
#OfServerRegions: 1
ASID1: 49, ASID2: 0, ASID3: 0, ASID4: 0, ASID5: 0
UserCredentials: IBMUSER
ActivityType: 1 (method request)
ActivityID: * b5bad234 1f950520 000000dc 00000006 * &sup1;
           &#141;K..n.... &#245;.... *
           * 09263048 xxxxxxxx xxxxxxxx xxxxxxxx *
           ...&#171;..... *
WlmEnclaveToken: * 00000020 000001e0 xxxxxxxx xxxxxxxx *
                 .....\...... *
ActivityStartTime: * b5bad234 1f950520 xxxxxxxx xxxxxxxx *
                  &sup1;&#141;K..
                  n..... *
ActivityStopTime: * b5bad28b 219e3847 xxxxxxxx xxxxxxxx *
                  &sup1;&#141;K&#164;.
                  &#241;&#226;..... *
#InputMethods: 1, #GlobalTransactions: 1, #LocalTransactions: 0
```

```
Triplet #3
CommSessionSection
CommSessionHandle: * 25a48320 00000001 xxxxxxxx xxxxxxxx * .uc..... *
CommSessionAddress: jobname=BBOAX8 asid=0035
CommSessionOptimization: 1 (local optimization)
DataReceived: 402, DataTransferred: 1688
```

(...)

SMF file analysis complete.

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/d1/websphere20/zos390-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 -----
                  * ā... * p1047
  ActivityID     * b6c7a7b7 14e9bc85 000000b0 00000007
                  * æGxηZ7e 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 Application Server

Before you begin: Ensure that you have proper access to the Administration application.

Follow the step below to disable SMF recording for WebSphere Application Server through the server definition of the System Management User Interface Administrator application (see *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface*, SA22-7838).

- Disabling SMF recording can be achieved by the steps outlined in “Steps for using the System Management User Interface Administrator application” on page 137, and by deselecting the records checked.

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 149 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 181 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 (Version 2)**

- **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.
- **Subtype 2: Container activity record (Version 2)**
 - **Container activity section** (one section per record):
Contains information about each activity that occurred within one container.
 - **Class section** (multiple sections per record):
Contains information about all classes involved in this activity.
 - **Method section** (multiple sections per class section):
Contains information about all methods of this class involved in this activity.
- **Subtype 3: Server interval record (Version 2)**
 - **Server interval section** (one section per record):
Contains aggregated information about all activities that occurred within the specified server interval.
- **Subtype 4: Container interval record (Version 2)**
 - **Container interval section** (one section per record):
Contains aggregated information about all activities that occurred within one container in the specified interval.
 - **Class section** (multiple sections per record):
Contains information about all classes involved in this activity in the specified interval.
 - **Method section** (multiple sections per class section):
Contains information about all methods of this class involved in this activity in the specified interval .
- **Subtype 5: J2EE container activity record (Version 1)**
 - **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 (Version 1)**
 - **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 1)**
 - **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 1)**
 - **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 176 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 <i>WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface</i> , 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 <i>WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface</i> , 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 0cyydddF. See "Standard SMF record header" in <i>WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface</i> , 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

Individual header extension for subtype 2					
40	28	SM120CAS	4	binary	Offset to container activity section from RDW
44	2C	SM120CAL	4	binary	Length of container activity section
48	30	SM120CAN	4	binary	Number of container activity sections

The following triplet appears 0–n times; once for each class section.					
52	34	SM120CLS	4	binary	Offset to class section from RDW
56	38	SM120CLL	4	binary	Length of class section
60	3C	SM120CLA	4	binary	Number of class 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
----	----	----------	---	--------	------------------------------------

Individual header extension for subtype 4					
40	28	SM120CIS	4	binary	Offset to container interval section from RDW
44	2C	SM120CIL	4	binary	Length of container interval section
48	30	SM120CIN	4	binary	Number of container interval sections

The following triplet appears 0–n times; once for each class section.					
52	34	SM120CLS	4	binary	Offset to class section from RDW
56	38	SM120CLL	4	binary	Length of class section
60	3C	SM120CLN	4	binary	Number of class 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 (Version 2):

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.

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	SM120STY	4	binary	The WebSphere for z/OS server type. 0: MOFW server. 1: J2EE server.

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	<p>Communications session optimization</p> <p>1: Local communications session: The session is a local OS/390 optimized communications session.</p> <p>2: Remote communications session: The session is a remote communications session.</p> <p>3: Remote encrypted (SSL)</p> <p>4: Remote within sysplex.</p>
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 2: Container activity record (Version 2):

1. **Container activity section** (one section per record):
 Contains information about each activity that occurred within one container.
2. **Class section** (multiple sections per record):
 Contains information about all classes involved in this activity.
3. **Method section** (multiple sections per class section):
 Contains information about all methods of classes involved in this activity.

Subtype 2: Container 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	SM120ASR	4	binary	The specific WebSphere for z/OS transaction server instance server region where the request ran
84	54	SM120CNM	256	EBCDIC	WebSphere for z/OS container name
340	154	SM120CTP	4	binary	Container transaction policy Values: <ul style="list-style-type: none"> • 1: Transaction required • 2: Same-Server Hybrid Global • 3: Hybrid Global • 4: Supports Same-Server Hybrid Global.
344	158	SM120CSP	4	binary	Container security policy: <ul style="list-style-type: none"> • 000000001: DCE • 000000010: Userid password • 000000100: Userid passticket • 000001000: SSL type 1 • 000010000: non-authenticated clients • 000100000: SSL client certificates • 001000000: Kerberos • 010000000: SendAssertedId • 100000000: AcceptAssertedId
348	15C	SM120WLM	8	HEX	WLM enclave token
356	165	SM120ATY	4	binary	Type of activity that this record references: <p>1: Method request: This record refers to a method request that is not part of a global transaction.</p> <p>2: Transaction: This record refers to a transaction.</p>
360	168	SM120AID	20	HEX	Identity of the activity

Subtype 2: Class section

Offset	Offset	Name	Length	Format	Description
0	0	SM120CLN	256	EBCDIC	Name of a class activated by the container
256	100	SM120NIC	4	binary	Number of instances of this class that were created
260	104	SM120NIA	4	binary	Number of instances of the class that were activated
264	108	SM120NIR	4	binary	Number of instances of this class that were removed (deleted)
268	10C	SM120NIP	4	binary	Number of instances of this class that were passivated
272	110	SM120RMR	4	binary	Reserved
276	114	SM120RMW	4	binary	Reserved
280	118	SM120MN	4	binary	Number of method triplets in this class section
The following triplet appears 0–n times; once for each method section.					
284	11C	SM120MS	4	binary	Offset to method section from the beginning of this class section
288	120	SM120ML	4	binary	Length of method section
292	124	SM120MN	4	binary	Number of method sections

Subtype 2: Method section

Note: When a client invokes a method on an object instance in a WebSphere for z/OS server region and that method, in turn, invokes other methods within the same object instance, only the first method invoked by the client will be recorded to SMF. The subsequent methods it invokes within the same object instance will not be recorded.

Offset	Offset	Name	Length	Format	Description
0	0	SM120MNM	256	EBCDIC	Name of the method Note: The only way SMF recording distinguishes methods is by method name. SMF recording does not evaluate the complete method signature.

256	100	SM120NMI	4	binary	Number of times the method was invoked during the activity
260	104	SM120NEX	4	binary	Number of non-framework exceptions that were detected by the container
264	108	SM120ART	4	binary	Average response time: The response time is measured in micro seconds. If the value exceeds 2**31 micro seconds, this field becomes negative and the accuracy is changed to seconds. A positive value is the time in micro seconds. A negative value is the time in seconds.
268	10C	SM120MRT	4	binary	Maximum response time: The response time is measured in micro seconds. If the value exceeds 2**31 micro seconds, this field becomes negative and the accuracy is changed to seconds. A positive value is the time in micro seconds. A negative value is the time in seconds.

Subtype 3: Server interval record (Version 2):

1. Server interval section (one section per record):

Contains information about each activity that occurred within 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	SM120STY	4	binary	The WebSphere for z/OS server type. 0: MOFW server. 1: J2EE server.

Subtype 4: Container interval record (Version 2):

1. Container interval section (one section per record):

Contains information about each activity that occurred within one container in the specified interval.

2. Class section (multiple sections per record):

Contains information about all classes involved in this activity in the specified interval.

3. Method section (multiple sections per class section):

Contains information about all methods of all classes involved in this activity in the specified interval.

Subtype 4: Container interval section

Offsets	Offsets	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	SM120CNM	256	EBCDIC	WebSphere for z/OS container name
336	150	SM120CTP	4	binary	Container transaction policy. Values: <ul style="list-style-type: none"> • 1: Transaction required • 2: Same-Server Hybrid Global • 3: Hybrid Global • 4: Supports Same-Server Hybrid Global.
340	154	SM120CSP	4	binary	The container security policy <ul style="list-style-type: none"> • 000000001: DCE • 000000010: Userid password • 000000100: Userid passticket • 000001000: SSL type 1 • 000010000: non-authenticated clients • 000100000: SSL client certificates • 001000000: Kerberos • 010000000: SendAssertedId • 100000000: AcceptAssertedId
344	158	SM120SST	16	S390STCK	The time that the sample began in the server.
360	168	SM120SET	16	S390STCK	The time that the sample ended.

Subtype 4: Class section: (See Subtype 2: Class section)

Subtype 4: Method section: (See Subtype 2: Method section)

Subtype 5: J2EE container activity record (Version 1):

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

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

Subtype 6: Container interval record (Version 1):

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.
-----	-----	----------	----	----------	---------------------------------

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 1):

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.

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

Subtype 8: WebContainer interval record (Version 1):

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.

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

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

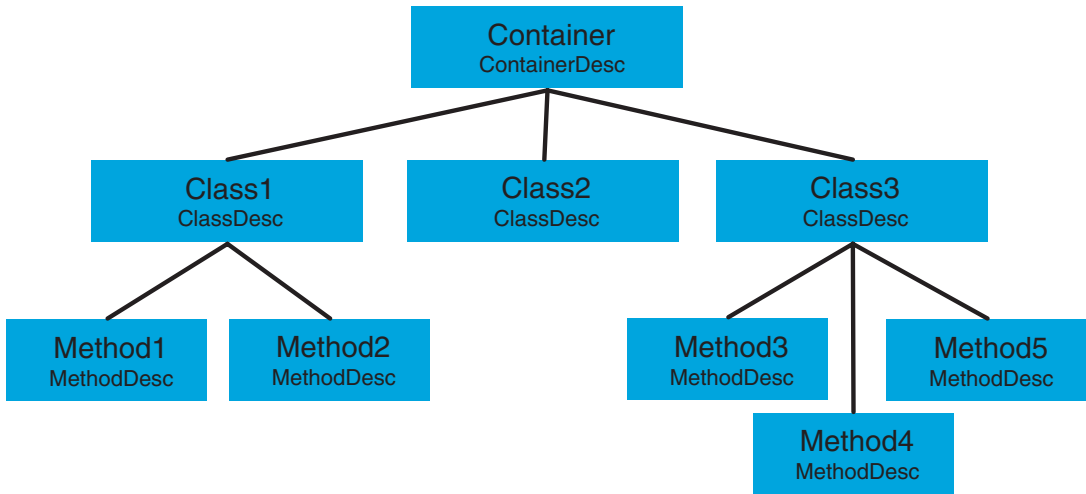


Figure 6. SMF records: logical records and split mechanism

Split between classes

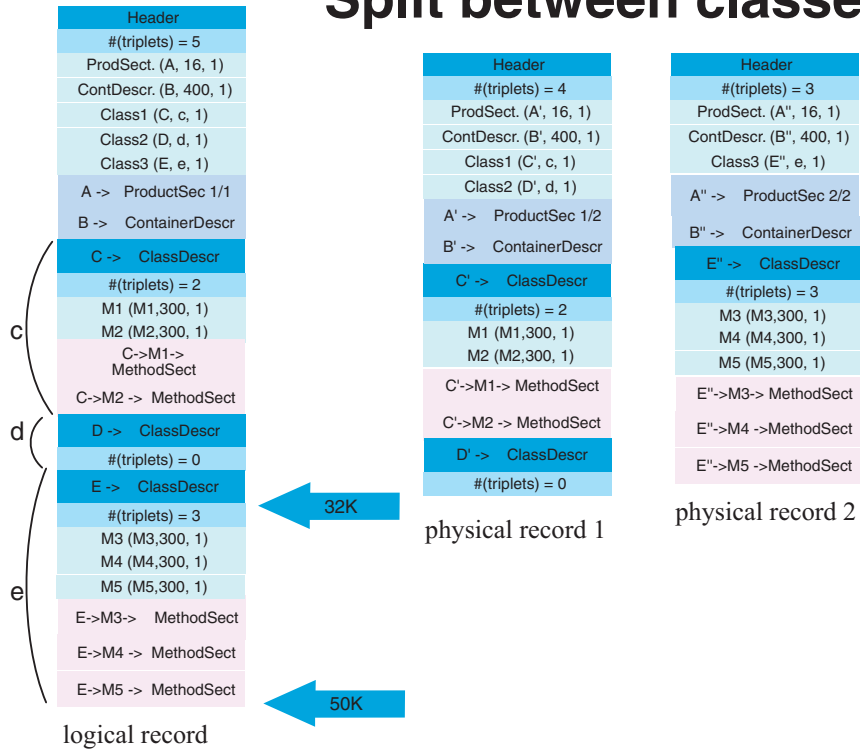


Figure 7. SMF record: split between classes

Split between methods

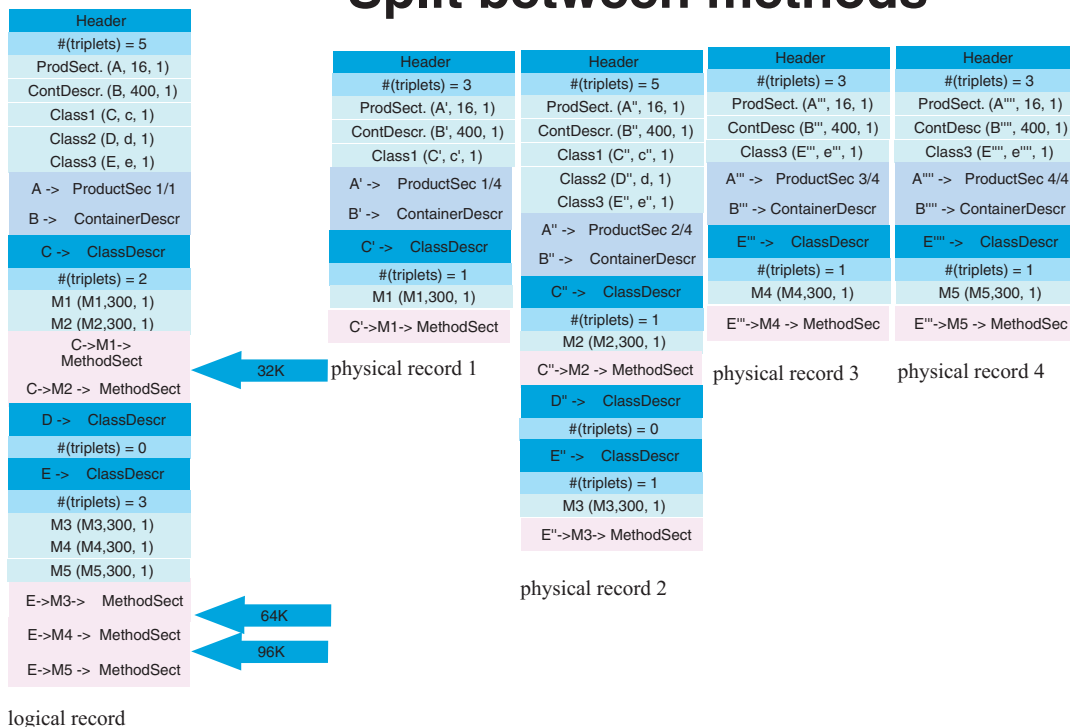


Figure 8. SMF record: split between methods

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 21. 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
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. MOFW tuning considerations

This chapter outlines tuning considerations specific to MOFW objects.

For MOFW objects, there are several additional transaction policies: `HYBRID_GLOBAL` and `SUPPORTS_HYBRID_GLOBAL`. These policies are similar to local transactions in the EJB world. The overhead of a transaction is reduced since they do not assume a full two-phase OTS mediated transaction. This policy is not standard and should be used carefully since it does affect the behavior of the application and possibly makes it nonportable.

Transient objects in WebSphere for z/OS are about twice as fast as persistent objects when running in a `HYBRID_GLOBAL` transaction. This is primarily because there is no interaction with any other resource manager, which eliminates the need to coordinate a transaction. Also, there is no need to do any logging to disk, so the latency of the transaction is improved.

If possible, read-only persistent data can be configured in a container which has a pinned policy. This means that the data will be read once from the database per server and kept in memory rather than being retrieved for each transaction.

To improve MOFW query performance, the environment variable `SOM00SQL=1` is set by default in the server configuration. When this is set, we can easily push queries down to DB2. This has a significant improvement in query performance. IBM recommends that you set this variable to on. When set to on, you lose some NLS support in Query.

You should also ensure that your query calls are being pushed down to DB2. This is most easily determined by looking at DB2PM reports where you look at the detail of the calls to DB2. You can review the resulting query statement that was issued to DB2 and see how many fetches are done from DB2. See "Tuning tips for DB2" on page 87 for more information.

Finally, the definition of each home (or container), has an option for whether you need method level access checking. If you don't need it, you can improve performance by shutting it off.

Appendix C. Naming conventions for application servers

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

Naming conventions for application servers

- 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
- ORBsrvmname 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.

Naming conventions for application servers

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
```

Naming conventions for application servers

```
// 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..SBBLOAD,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 IWSSNM='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 &IWSSNM'  
//STEPLIB DD DSN=&BBOLIB..SBBOLIB,DISP=SHR  
//* DD DSN=&BBOLIB..SBBOLD2,DISP=SHR  
//* DD DSN=&BBOLIB..SBBLOAD,DISP=SHR  
//* DD DSN=&LELIB..SCEERUN,DISP=SHR  
//* DD DSN=&DB2..SDSNLOAD,DISP=SHR  
//BBOENV DD PATH='&CBCONFIG/&RELPATH/&SYSPLEX/&IWSSNM/current.env'  
//CEEDUMP DD SYSOUT=**  
//SYSOUT DD SYSOUT=**  
//SYSPRINT DD SYSOUT=**
```

Appendix D. Notices

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

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

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

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

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will

be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

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

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

Licenses 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	RACF
DB2	RMF
IBM	S/390
IMS	VTAM
IMS/ESA	WebSphere
MVS	z/OS
OS/390	

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.

Glossary

For more information on terms used in this book, refer to one of the following sources:

- Sun Microsystems Glossary of Java Technology-Related Terms, located on the Internet at:
<http://java.sun.com/docs/glossary.html>
- *IBM Glossary of Computing Terms*, located on the Internet at:
<http://www.ibm.com/ibm/terminology/>
- The Sun Web site, located on the Internet at:
<http://www.sun.com/>

Index

A

- Administration
 - Overview 2
- Administration and Operations applications
 - adding a new administrator 69
- APPC automation and recovery 50
- Application Assembly tool 112
- application messages
 - issuing to the MVS master console 66
- application servers
 - naming conventions 183
- Auditing in WebSphere 149
 - SMF record type 120 149
 - SMF record type 80 181
- authentication 107
- automation and recovery scenarios 50

B

- Backup
 - Guidelines and procedures 45
- beans
 - enterprise Java 111
 - entity 111
 - session
 - statefull 111
 - stateless 111

C

- canceling application servers and server instances 30, 34
- CICS
 - tuning 101
- CICS automation and recovery 56
- CLIENT_TIMEOUT 21
- clients
 - local or remote 110
- CMP 115
- cold starting WebSphere for z/OS 16
- collecting additional performance diagnosis information 130
- collecting performance diagnosis information 125
- component management 112
- configuration
 - recommended hardware 74

- connection
 - CMP 115
- container activity record 134
- container configuration 111
- container interval record 135
- Container Managed Persistence (CMP) Connection and Prepared Statement Pooling 115
 - Database object
 - characteristics 116
 - Connections 116
 - Prepared statements 117
 - JVM properties related to CMP Connection Pooling 116
 - overview 115
 - Two methods of CMP Connection Pooling 115
 - "Across" transaction pooling 116
 - "Within" transaction pooling 115

D

- daemon
 - definition 11
- Daemon
 - canceling 30
- Daemon automation and recovery 61
- DB2
 - tuning 87
- DB2 automation and recovery 55
- DB2 for WebSphere
 - tuning 88
- DB2 for z/OS or OS/390
 - backing up 49
- definitions
 - daemon 11
 - server 11
 - server instance 11
 - sysplex 11
 - z/OS or OS/390 system 11
- determining if the Daemon is up 15
- determining where to issue messages 65
- display operations 31
- displaying control region and trace settings status 36

- displaying the status of ARM-registered address spaces 31

E

- EJBs 111
- Elements 2
- enterprise Java beans 111
- entity beans 111
- environment variables
 - CLIENT_TIMEOUT 21
- example
 - server configuration 103

F

- functions of SMF recording 137

G

- garbage collection
 - JVM_HEAPSIZE 76
- getting help for the modify command 34
- GRS

- tuning 99
- guidelines
 - automation and recovery 50

H

- handling WLM failures 40
- hardware
 - recommended configuration 74
- HFS
 - performance 85
- HFS directories 47
- hot starting WebSphere for z/OS 16
- HTTP transport 108
 - using 111

I

- IMS automation and recovery 57
- Interface Repository automation and recovery 62
- Introduction i
- issuing application messages to the MVS master console 66

J

- J2EE
 - programming tips 117

- J2EE applications
 - tuning 108
- J2EE container 111
- J2EE container activity record 135
- J2EE container interval record 136
- Java database connectivity 114
- Java performance
 - garbage collection 76
- Java virtual machine properties 114
- JDBC parameters 114
- JDBC topic 117
- JMS
 - tuning 92
- JSP topic 117
- JVM
 - tuning 99
- jvm properties file 114
- JVM_HEAPSIZE
 - garbage collection 76

L

- Language Environment
 - tuning 82
- LDAP
 - tuning 101
- LDAP automation and recovery 59
- LE
 - tuning 82
- LE Heap
 - fine tuning 82
- Lightweight Directory Access Protocol (LDAP)
 - access control list, updating 69
 - backing up 47
- location of executable programs 106
- logging messages and trace data for Java server applications 65

M

- message and trace data
 - system performance when logging 66
- messages
 - BBOU0003E 23
 - BBOU0009E 23
 - BBOU0739I 23
 - BBOU0747I 23
 - BBOU0748I 23
 - BBOU0749I 23
 - BBOU0750I 23
 - BBOU0751I 23
 - BBOU0752I 23
 - BBOU0753I 23
 - BBOU0754D 23

- messages (*continued*)
 - BBOU0755I 24
 - BBOU0756I 24
 - BBOU0757I 24
 - BBOU0761W 24
 - BBOU0762W 24
 - BBOU0763W 24, 26
 - BBOU0764W 24, 27
 - determining where to issue 65
 - issuing to the MVS master console 66
 - modifying trace variables 34
 - MOFW tuning considerations 183
 - MVS console
 - message and trace data 66

N

- naming automation and recovery 62
- naming conventions for application servers 183
- NFS automation and recovery 60

O

- operating system
 - tuning 75
- operating WebSphere for z/OS 11
- operations
 - ARM and restart 37
 - cancel operations 30
 - canceling application servers and server instances 30
 - canceling the Daemon 30
 - Daemon
 - canceling the Daemon 30
 - determining if the Daemon is up 15
 - display operations 31
 - displaying active address spaces 32
 - displaying active replies 32
 - displaying indoubt units of work (threads) for DB2 33
 - displaying the status of a server or server instance 31
 - displaying the status of ARM-registered address spaces 31
 - displaying units of work (threads) for DB2 32
 - displaying units of work (transactions) for IMS 33
 - displaying units of work for CICS 33

- operations (*continued*)
 - display operations (*continued*)
 - displaying units of work for RRS 34
 - modify operations 34
 - canceling application servers and server instances 34
 - displaying control region and trace settings status 36
 - getting help for the modify command 34
 - modifying trace variables 34
 - other
 - WebSphere for z/OS 37
 - restart operations
 - peer restart and recovery 17
 - setting up error log streams 38
 - start operations 12
 - cold start 16
 - determining if the Daemon is up 15
 - hot start 16
 - quick start 16
 - starting server instances 16
 - starting servers and server instances 15
 - starting the WebSphere for z/OS environment 12
 - warm start 16
 - stop operations 28
 - stop vs. cancel 29
 - stopping servers and server instances 30
 - stopping the WebSphere for z/OS environment 29
 - subsystems 43
 - DB2 Universal Database 43
 - IMS 44
 - RRS 44
 - Workload Management (WLM) 45
 - WebSphere for z/OS
 - operation 11
 - Workload Management for operations 38
 - changing states 41
 - handling failures 40
 - WLM application environment status 38
- Operations
 - Identifying where to perform 3
 - MVS console operation 9
 - Overview 2
 - Systems Management User Interface operation 9

- Operations and Administration
 - Introduction i
 - other WebSphere for z/OS
 - operations 37
 - overview
 - peer restart and recovery 17
- Overviews
 - Administration 2
 - Operations 2
 - Required elements and subsystems 2

P

- parameters
 - JDBC 114
- peer restart and recovery 17
 - BBOU0763W
 - steps for resolving indoubts 26
 - environment variables 21
 - failure 22
 - in-flight work 19
 - in-flight work versus indoubt work 19
 - indoubt work 20
 - new work 21
 - overview 17
 - recoverable communication manager 18
 - resolving indoubts
 - BBOU0761W 25
 - BBOU0762W 25
 - BBOU0763W 26
 - BBOU0764W 27
 - RRS panels 22
- peer restart and recovery prerequisites 17
- performance
 - diagnosis
 - collecting additional information 130
 - collecting information 125
 - steps for diagnosing a performance problem 121
 - HFS 85
 - system
 - when logging messages and trace data 66
 - tuning guidelines 72
 - Unix System Service 84
- performance
 - information and accounting 121
 - monitoring procedures 121
- prerequisites
 - peer restart and recovery 17

- procedures
 - administration
 - logging messages and trace data for Java server applications 65
 - WebSphere for z/OS 65
- programs
 - executable
 - location of 106
- properties
 - Java virtual machine 114

Q

- quick starting WebSphere for z/OS 16

R

- RACF
 - tuning 90
- RACF automation and recovery 52
- real storage
 - tuning 75
- recommended hardware
 - configuration 74
- recycling the server region 77
- Required elements and subsystems
 - Elements 2
 - Overview 2
 - Subsystems 3
- resource recovery services (RRS)
 - backing up 47
- RRS automation and recovery 53
- RRS panels
 - using in recovery 22
- run-time environment
 - backup 47

S

- scenarios
 - automation and recovery 50
 - APPC 50
 - CICS 56
 - Daemon 61
 - DB2 55
 - IMS 57
 - Interface Repository 62
 - LDAP 59
 - naming 62
 - NFS 60
 - RACF 52
 - RRS 53
 - Systems Management (SM) 63
 - TCP/IP 55
 - Unix System Services 54
 - WebServer 64

- scenarios (*continued*)
 - automation and recovery (*continued*)
 - WLM 51
- security 107
 - Lightweight Directory Access Protocol (LDAP) 69
- server
 - definition 11
 - server activity record 134
 - server configuration
 - example 103
 - server instance
 - definition 11
 - server interval record 134
 - server region
 - isolation policy 109
 - recycling 77
 - server replicas
 - setting 110
 - servers
 - single or multiple 108
 - servlet/EJB integrated runtime 108
 - session beans 111
 - setting the number of threads 110
 - setting up error log streams 38
 - Simple Object Access Protocol
 - tuning 118
- SMF
 - recording and monitoring 133
- SMF record type 120 149
- SMF record type 80 181
 - Overview 181
 - Preparing audit support 181
 - Setting up RACF for audit support 181
 - Setting up SMF for audit support 181
 - Using audit support 181
- SMF recording
 - functions of 137
- SMF records
 - container activity record 134
 - container interval record 135
 - disabling recording 147
 - enabling recording 137
 - formatting output 141
 - J2EE container activity record 135
 - J2EE container interval record 136
 - record types 134
 - server activity record 134
 - server interval record 134
 - SMF Browser 144

- SMF records (*continued*)
 - SMF Record Interpreter 142
 - SMF ViewTool 143
 - subtype 1
 - Communications session section 159
 - Server activity section 158
 - subtype 2
 - Class section 162
 - Container activity section 160
 - Method section 162
 - subtype 3
 - Server interval section 163
 - subtype 4
 - Class section 166
 - Container interval section 165
 - Method section 166
 - subtype 5
 - Bean method section 168
 - Bean section 167
 - J2EE container activity section 167
 - subtype 6
 - Bean method section 171
 - Bean section 171
 - J2EE container interval section 170
 - subtype 7
 - HttpSessionManager activity 171
 - Servlet activity 172
 - WebApplication activity 172
 - WebContainer activity 171
 - subtype 8
 - HttpSessionManager interval 174
 - Servlet interval 175
 - WebApplication interval 175
 - WebContainer interval 173
 - type 120
 - record environment 151
 - record mapping 152
 - type 120 (78) 147
 - type 80 147
 - viewing output 141
 - WebContainer activity record 136
 - WebContainer interval 136
 - SOAP 118
 - starting a server 15
 - starting servers and server instances 15
 - starting the WebSphere for z/OS environment 12
 - startup order for WebSphere and subsystems 49
 - stateful session bean 111
 - stateless session bean 111
 - steps for diagnosing a performance problem 121
 - stopping the WebSphere for z/OS environment 29
 - storage 75
 - real tuning 75
 - virtual tuning 75
 - subsystems
 - backup guidelines and procedures 49
 - startup order 49
 - tuning 87
 - Subsystems 3
 - sysplex definition 11
 - system logger tuning 80
 - System Management Facility recording and monitoring 133
 - system performance when logging messages and trace data 66
 - Systems Management (SM) automation and recovery 63
- T**
- tasks (*continued*)
 - Container Managed Persistence (CMP) Connection and Prepared Statement Pooling overview 115
 - disabling SMF recording for the entire MVS system 147
 - disabling SMF recording for WebSphere Application Server 147
 - display operations
 - displaying the status of a server or server instance 31
 - displaying the status of ARM-registered address spaces 31
 - editing the SMFPRMxx parmlib member 140
 - enabling recording 137
 - formatting the output data set 141
 - granting database authority steps for 71
 - modify operations
 - canceling application servers and server instances 34
 - displaying control region and trace settings status 36
 - getting help for the modify command 34
 - modifying trace variables 34
 - operations
 - checking and starting the workload management application environment 40
 - displaying WLM application environment status 38
 - getting out of the stopped state and back to the available state 41
 - SMF record type 120 overview 149
 - SMF record type 80 overview 181
 - start operations
 - determining if the Daemon is up 15
 - starting server instances 16
 - starting the WebSphere for z/OS environment 12
 - steps for starting a server 15
 - steps for diagnosing a performance problem 121

- tasks *(continued)*
 - stop operations
 - stopping servers and server instances 30
 - stopping the WebSphere for z/OS environment 29
 - using a DB2 SPUFI script to turn SMF recording on and off 138
 - using a message logger 66
 - using the SMF ViewTool
 - installation and invocation 143
 - using the System Management User Interface Administrator application 137
 - viewing the output data set 141
 - writing records to DASD 141
- TCP/IP
 - tuning 91
- TCP/IP automation and recovery 55
- threads
 - setting the number of 110
- topology 108
- trace data
 - issuing to the MVS master console 66
- transaction policies 112
- tuning
 - CICS 101
 - DB2 87
 - DB2 for WebSphere 88
 - GRS 99
 - J2EE applications 108
 - JVM 99
 - LDAP 101
 - LE 82
 - LE Heap 82
 - MOFW 183
 - operating system 75
 - RACF 90
 - real storage 75
 - SOAP 118
 - storage 75
 - subsystems 87
 - system logger 80
 - TCP/IP 91
 - virtual storage 75
 - WebSphere for z/OS 72
 - WebSphere for z/OS runtime 102
 - WebSphere MQ and JMS 92
 - WLM 85
 - z/OS or OS/390 78

U

- Unix System Service
 - performance 84
- Unix System Services automation and recovery 54
- using a message logger 66

V

- virtual storage
 - tuning 75

W

- warm starting WebSphere for z/OS 16
- WAS 4.x integrated runtime 111
- WebContainer activity record 136
- WebContainer interval record 136
- WebServer automation and recovery 64
- WebSphere for z/OS
 - operation 11
- WebSphere for z/OS administration procedures 65
- WebSphere for z/OS runtime
 - tuning 102
- WebSphere MQ
 - tuning 92
- WLM
 - tuning 85
- WLM automation and recovery 51
- Workload Manager
 - tuning 85, 92

Z

- z/OS or OS/390
 - tuning 78
- z/OS or OS/390 system
 - definition 11



Program Number: 5655-F31

Printed in the United States of America

SA22-7835-05

