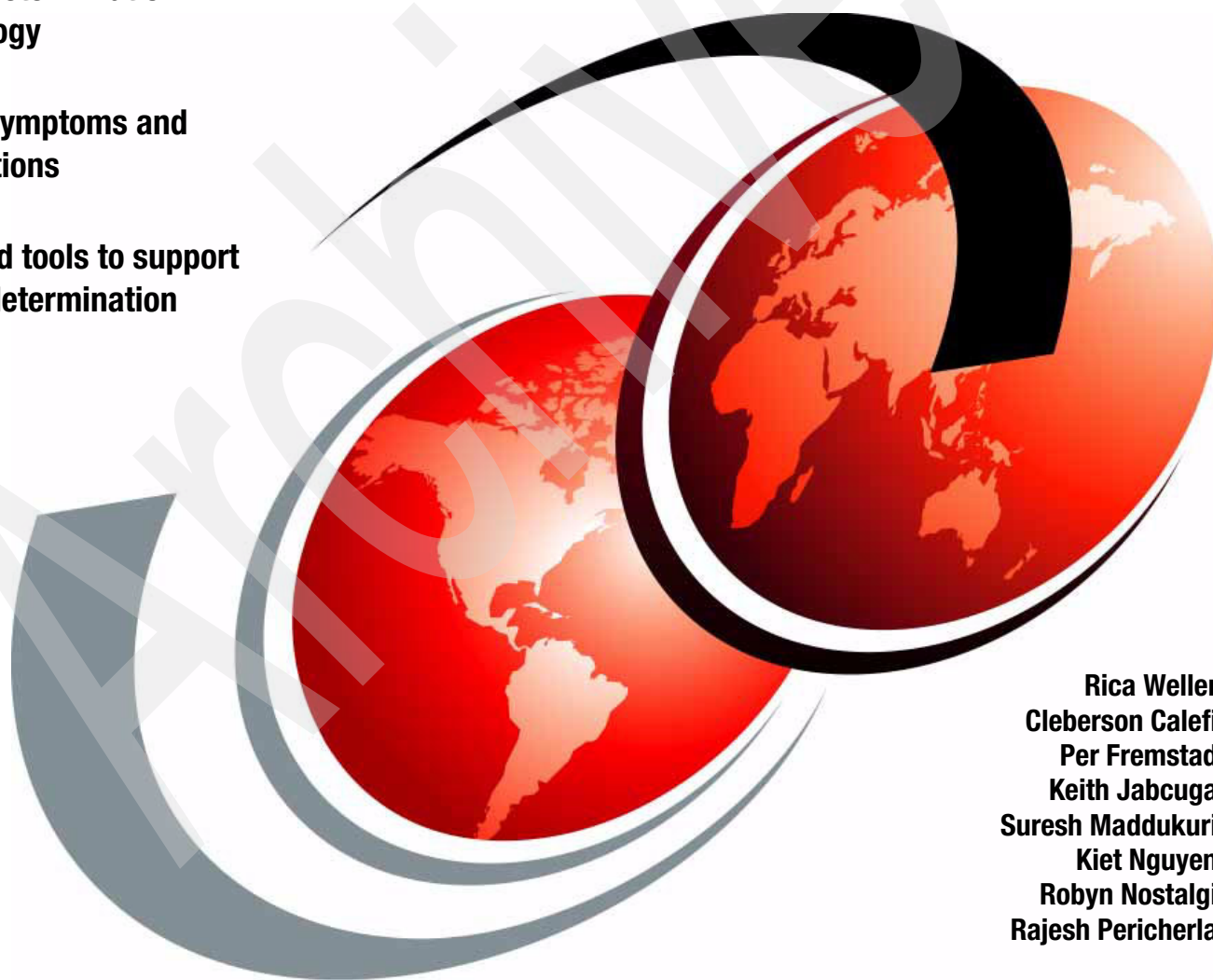


Problem Determination for WebSphere for z/OS

Problem determination
methodology

Problem symptoms and
their solutions

Means and tools to support
problem determination



Rica Weller
Clebertson Calefi
Per Fremstad
Keith Jabcuga
Suresh Maddukuri
Kiet Nguyen
Robyn Nostalg
Rajesh Pericherla

Redbooks



International Technical Support Organization

Problem Determination for WebSphere for z/OS

August 2006

Archived

Note: Before using this information and the product it supports, read the information in “Notices” on page xv.

Third Edition (August 2006)

This edition applies to Version 6, Release 0, Modification 1 of WebSphere Application Server for z/OS (program number 5655-N01).

© Copyright International Business Machines Corporation 2002, 2005, 2006. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	ix
Tables	xiii
Notices	xv
Trademarks	xvi
Preface	xvii
How the book is structured	xvii
What this book is about	xvii
Who this book is for	xviii
How to use this book	xviii
The team that wrote this redbook	xix
Become a published author	xxi
Comments welcome	xxi
Part 1. Problem determination methodology	1
Chapter 1. Problem determination methodology	3
1.1 What problem determination is	4
1.2 What problem determination is not	4
1.3 Problem determination approach	4
1.3.1 Problem determination flow chart	4
1.3.2 Problem determination process	5
1.3.3 Specific considerations for WebSphere for z/OS problem determination	8
1.3.4 The importance of a test environment	8
1.4 The skills needed for WebSphere for z/OS	9
1.4.1 System skills	9
1.4.2 Skills for deploying and running an application	10
Chapter 2. Contacting IBM: Information	13
2.1 Communicating with IBM	14
2.2 The IBM WebSphere support structure	14
2.3 Before you contact IBM support	15
2.3.1 Defining the problem	15
2.3.2 Determining whether this situation has already been reported	15
2.3.3 Gathering background information	16
2.3.4 Determining the business impact	18
2.4 How IBM Software Support handles your problem	19
2.4.1 The PMR	19
2.4.2 Investigating a PMR	19
2.4.3 How technical questions are handled by IBM	20
2.5 Exchanging data with IBM by FTP	21
2.5.1 Copying the job log into a z/OS data set	21
2.5.2 Compressing the data	21
2.5.3 Finding specific FTP instructions	22
2.5.4 Using naming conventions	23
2.6 IBM contacts	23

Chapter 3. Information sources	25
3.1 WebSphere for z/OS support pages	26
3.1.1 The WebSphere for z/OS home page	26
3.1.2 WebSphere support page	27
3.1.3 WebSphere for z/OS V6 product manuals	27
3.1.4 WebSphere for z/OS V6 Information Center	29
3.1.5 WebSphere for z/OS IBM services	30
3.1.6 Recommended reading list: WebSphere Application Server	31
3.2 Techdocs: White papers, hints, and tips	31
3.3 Redbooks and draft publications	31
3.4 Sources of information for developers	32
3.4.1 WebSphere Developer's Domain	32
3.4.2 The alphaWorks community	33
3.4.3 Java	33
3.5 Other helpful Web sites	33
3.5.1 zSeries support	34
3.5.2 z/OS home page	34
3.5.3 LookAt messages	34
3.5.4 All software products	35
3.5.5 IBM Software support guide	35
3.5.6 z/OS Internet library	36
3.6 Educational information	36
3.6.1 IBM Global Services	36
3.6.2 WebSphere for z/OS training and certification	36
3.6.3 IBM Education Assistant	36
Part 2. Problem symptoms and their resolutions	39
Chapter 4. Exceptions and error messages	41
4.1 What is an exception or error?	41
4.2 Symptom flow chart: Exceptions and error messages	42
4.3 Diagnosing an error or exception message	43
Chapter 5. Abend	49
5.1 What is an abend?	50
5.2 Symptom flow chart: Abend	50
5.3 Diagnosing an abend	51
Chapter 6. Hang	59
6.1 What is a hang?	60
6.2 Symptom flow chart: Hang	60
6.3 Diagnosing a hang	60
Chapter 7. Timeout	67
7.1 What is a timeout?	68
7.2 Symptom flow chart: Timeout	68
7.3 Diagnosing a timeout	69
Chapter 8. Does not stop	75
8.1 What is the does not stop symptom?	76
8.2 Symptom flow chart: Does not stop	76
8.3 Diagnosing the symptom	77
Chapter 9. Job failed	83
9.1 What is job failed?	84

9.2 Symptom flow chart: Job failed	84
9.3 Diagnosing the job failed symptom	85
Chapter 10. No response	89
10.1 What does “no response” mean?	90
10.2 Symptom flow chart: No response	90
10.3 Diagnosing the no response symptom	91
Chapter 11. No resource access	99
11.1 What is no resource access?	100
11.2 Symptom flow chart: No resource access.	100
11.3 Diagnosing no resource access	101
Chapter 12. High CPU utilization	109
12.1 What is high CPU utilization?	110
12.2 Symptom flow chart: High CPU utilization.	111
12.3 Diagnosing high CPU utilization	111
Chapter 13. WebSphere for z/OS performance analysis	117
13.1 Performance terminology	118
13.1.1 Response time	118
13.1.2 Throughput	118
13.1.3 Transaction	118
13.1.4 Hit rate	119
13.1.5 Page view rate	119
13.1.6 Number of clients and think time.	119
13.1.7 Resource.	119
13.2 Managing performance of WebSphere transactions.	120
13.2.1 Managing the number of servant regions	120
13.2.2 Managing the number of JVM threads	122
13.2.3 Classifying servant region enclaves (WebSphere transactions).	122
13.2.4 Classifying servant regions	124
13.2.5 Classifying controller regions	125
13.2.6 Special considerations for HTTP requests over multiple servants	125
13.3 Introduction to performance analysis and management	126
13.3.1 Setting your performance expectations.	126
13.3.2 What is a performance problem and how do you manage it?	127
13.3.3 What to do about a performance problem.	128
13.4 Diagnosing performance problems	129
13.4.1 Understanding the expectations	130
13.4.2 Quantify: Take a quick snapshot view of the system	131
13.4.3 Finding the cause	132
13.4.4 Analyzing a heap or memory problem	136
13.4.5 Analyzing a response time problem	136
13.4.6 Analyzing a high CPU usage problem	137
13.5 Related information sources	138
Part 3. Problem avoidance and best practices.	139
Chapter 14. Phase 1: Installation, configuration, and migration	141
14.1 Preparing the Installation	142
14.2 Installation and configuration	143
14.3 Migration	144
14.3.1 Migrating from V5.x to V6.0.x	144

14.3.2	Migrating from V4.0.1 to V6.0.x	145
14.3.3	Migrating from V3.5 Standard Edition to Version 6.0.x	145
14.3.4	Checklist for migration	145
14.4	Coexistence	147
14.5	Most common problems	147
14.6	Related references	151
Chapter 15.	Phase 2: Application deployment	153
15.1	Tools for the deployment phase	154
15.1.1	Installing and deploying application files	154
15.1.2	Logging and tracing	155
15.2	Problem avoidance checklist	156
15.2.1	Assembling an application	156
15.2.2	Deploying an application	156
15.3	Most common problems	157
15.4	Related references	163
Chapter 16.	Phase 3: Run applications	165
16.1	Request process overview	166
16.2	Model-view-control model for problem determination	167
16.2.1	Typical problems in the view tier	167
16.2.2	Typical problems in the control tier	170
16.2.3	Typical problems in the model tier	173
16.3	Problem avoidance	175
16.3.1	Designing, coding, and testing	175
16.3.2	Change control	176
Chapter 17.	Phase 4: System run time	177
17.1	The WebSphere for z/OS V6 runtime environment	178
17.2	Problem categories in the runtime phase	180
17.3	Understanding your own runtime configuration	182
17.4	Troubleshooting tips for the runtime environment	182
17.5	Security issues and problems	183
17.6	Problem avoidance checklist	183
17.7	Typical problems	186
Part 4.	Problem Determination Means and Tools	193
Chapter 18.	Commands	195
18.1	Commands for administering WebSphere for z/OS	196
18.2	z/OS MODIFY commands	196
18.2.1	z/OS DISPLAY commands	197
18.2.2	Basic TRACE commands	201
18.2.3	Dynamic Java TRACE	201
18.3	TCP/IP related commands	202
18.3.1	The netstat command	203
18.3.2	The nslookup command	204
18.3.3	The ping command	205
18.3.4	The tracert command	205
18.4	USS and OMVS commands	206
18.4.1	Display file system with df	206
18.4.2	Display disk space usage with du	207
18.4.3	Display thread information with ps	208
18.4.4	Display thread details with DISPLAY	209

18.4.5 Search string patterns with WASgrep.sh	210
18.5 Windows FTP command	211
Chapter 19. Logs for problem determination in WebSphere for z/OS	213
19.1 Job logs and system log	214
19.1.1 When to use system log and job logs	214
19.1.2 How to set up system log and job logs	214
19.1.3 System log and job log output and their interpretation	215
19.2 WebSphere error log (BBORBLOG)	216
19.2.1 When to use BBORBLOG	216
19.2.2 How to set up BBORBLOG	217
19.2.3 BBORBLOG output and its interpretation	218
19.3 First Failure Data Capture	219
19.3.1 When to use FFDC	220
19.3.2 How to set up the FFDC tool	220
19.3.3 FFDC output and its interpretation	221
19.3.4 Example: Using the FFDC tool for problem determination	225
19.4 The Java Logging API	227
19.4.1 What is the Java Logging API and when to use it	227
19.4.2 Setting up the Java Logging API	228
19.4.3 Java Logging output and interpretation	231
19.4.4 Java Logging API example	232
19.5 IBM HTTP Server logs and trace	232
19.5.1 Server error log	233
19.5.2 Server access log	234
19.5.3 Very verbose trace	235
19.5.4 HTTP plug-in log	237
Chapter 20. WebSphere for z/OS traces and dumps	241
20.1 CTRACE for WebSphere	242
20.1.1 Setting up and taking a CTRACE	242
20.1.2 Viewing CTRACE and JRas data through IPCS	242
20.2 JDBC trace	244
20.2.1 Setting up the JDBC trace	244
20.2.2 JDBC trace output and interpretation	245
20.3 SVC dumps	247
20.3.1 Capturing an SVC dump	247
20.3.2 Problems capturing an SVC dump	248
20.3.3 Formatting an SVC dump using IPCS	248
20.3.4 Related references	249
20.4 CEEDUMP	249
20.5 Java Transaction Dump (TDUMP)	250
20.6 Javadump	251
20.7 Heapdump	251
Chapter 21. Diagnostic tools for WebSphere for z/OS	253
21.1 Collector tool	254
21.2 JVM dump and heap analysis tools	254
21.2.1 Svcdump.jar	255
21.2.2 HeapRoots	259
21.2.3 Dumpviewer GUI and jformat	259
21.3 Memory Dump Diagnostic Tool for Java	260
21.4 Trace Analyzer for WebSphere Application Server	261
21.5 Java Garbage Collection Formatter	262

21.6 dumpNameSpace tool.	265
21.7 Rational Application Developer V6 Debug Perspective	266
21.7.1 When to use the Rational Application Developer Debug Perspective	266
21.7.2 Setting up the Rational Application Developer Debug Perspective	267
21.7.3 Rational Application Developer debugger output and interpretation.	269
21.8 Tivoli Performance Viewer	270
21.8.1 Setting up Tivoli Performance Viewer.	270
21.8.2 Tivoli Performance Viewer output and its interpretation	271
21.9 OMEGAMON XE for WebSphere	273
Chapter 22. Other handy tools	275
22.1 TCP/IP related tools	276
22.1.1 TCP/IP checkout program (InetInfo.java)	276
22.1.2 TCP/IP network packet tracing with Ethereal	277
22.1.3 TCP/IP for z/OS packet trace	279
22.2 MVS Extended Information	281
22.3 Resource Measurement Facility reports	283
22.3.1 Running the RMF post processor	284
22.3.2 Analyzing RMF reports	285
22.3.3 References	292
22.4 System Management Facility records and browser	293
22.4.1 Setting up SMF recording	293
22.4.2 WebSphere for z/OS SMF browser.	295
22.5 Stress test tools.	300
22.5.1 WebSphere Studio Workload Simulator for z/OS and OS/390	300
22.5.2 Microsoft Web Application Stress Tool	305
22.6 FTP, Telnet, and editors	306
22.6.1 TeraTerm Pro	306
22.6.2 WS_FTP Professional.	307
22.6.3 Directing SYSPRINT output to an HFS file	308
22.6.4 UltraEdit	308
Appendix A. Messages and codes	311
A.1 WebSphere for z/OS message codes	312
A.1.1 Specific Java component messages	312
A.1.2 Minor codes	314
A.1.3 Abends	315
A.2 System and component message table	315
Appendix B. Additional material	319
B.1 Locating the Web material	320
B.2 Using the Web material	320
Related publications	321
IBM Redbooks	321
Other publications	321
Online resources	322
How to get IBM Redbooks	323
Help from IBM	323
Index	325

Figures

1-1	General problem determination flow chart	5
1-2	Working together	9
1-3	Deploying applications	11
2-1	IBM support structure	14
3-1	IBM WebSphere Application Server for z/OS home page	26
3-2	IBM WebSphere for z/OS support Web site	27
3-3	IBM WebSphere Application Server library page	28
3-4	IBM WebSphere Application Server for z/OS Information Center page	29
3-5	Messages and codes in the Information Center	30
3-6	Recent IBM Redbooks and Redpapers	32
3-7	zSeries support page	34
3-8	LookAt messages	35
3-9	IBM Education Assistant WebSphere for z/OS: Problem determination	37
4-1	Flow chart for symptom: Exception and error	42
4-2	A Java stack trace with an exception	44
5-1	Flow chart for symptom: Abend	50
5-2	Example of IEA995I message with abend code	51
5-3	Output from IPCS ip st worksheet validate	55
5-4	IPCS ip summ format output showing RTM2WA SUMMARY	56
5-5	Browse dump storage using IPCS	56
5-6	Search for eye catchers in dump storage near PSW address	56
5-7	Traceback using IPCS data	57
6-1	Flow chart for symptom: Hang in the application server	60
7-1	Flow chart for symptom: Timeout	68
7-2	Session timeout	69
7-3	Active Jobs in DA panel	70
7-4	FFDC file information in trace	72
8-1	Flow chart for symptom: Does not stop	76
8-2	FFDC file information in trace	78
8-3	Output from IPCS ip st worksheet validate	79
8-4	IPCS ip summ format output showing RTM2WA SUMMARY	80
8-5	Browse dump storage using IPCS	80
8-6	Search for eye catchers in dump storage near PSW address	80
8-7	Traceback using ipcs ledata	81
9-1	Flow chart for symptom: Job failed	84
10-1	Flow chart for symptom: No response	90
10-2	Status of installed applications in Administrative Console	92
10-3	WebSphere for z/OS change log detail levels	94
11-1	Flow chart for symptom: No resource access	100
11-2	New JDBC provider for DB2	102
11-3	Properties for Data Sources	102
11-4	Changing database access scope	103
11-5	Test JDBC connection in Administrative Console	104
11-6	WebSphere for z/OS logging detail levels	105
12-1	Flow chart for symptom: High CPU utilization	111
13-1	WebSphere workload definition with Workload Manager	121
13-2	WLM definitions for servers and transaction classes in CB subsystem	123
13-3	WLM definition of Service Class WASHI	124

13-4	WLM definition of the servant regions, STC subsystem	125
13-5	Performance monitoring: An overview	130
13-6	Partition view from Partition Data Report	131
13-7	CP usage, response time, and throughput	133
13-8	CPU% and response time versus throughput	134
13-9	CP time per transaction	135
14-1	Problem areas in Phase 1	141
16-1	Request/response flow	166
16-2	Verifying the status of an application server	170
16-3	Hit Count application Web page	172
16-4	HTML source code instead of proper application	172
16-5	JDBC Data source configuration	174
17-1	WebSphere V6 for z/OS runtime structure (network deployment configuration)	179
17-2	Stand-alone application server configuration	180
18-1	USS command df display	207
18-2	Using DISPLAY OMVS to show thread information	209
18-3	FTP client delivered with Windows	211
19-1	FFDC tool architectural overview	220
19-2	ffdcRun.properties level values	221
19-3	Index and exception Logs	225
19-4	Java logging architecture	228
19-5	Enable log in Diagnostic Trace Service	229
19-6	WebSphere Change Log Detail Levels	230
19-7	IBM HTTP Server logs and trace	233
19-8	Server error log sample	233
19-9	Server access log sample	235
19-10	A plug-in trace record and some of the important fields	238
19-11	Plug-in traces	238
20-1	CEEDUMP sample	249
21-1	Memory Dump Diagnostic tool for Java screens	261
21-2	Trace Analyzer for WebSphere for z/OS	262
21-3	Enable GC Verbose in Advanced Java virtual machine settings	263
21-4	Diagram of garbage collection records	264
21-5	Enable Debugging Service in Administrative Console	267
21-6	Debug menu from the Debug icon	268
21-7	Remote Java Application Debug configuration	268
21-8	The Debug Perspective in Rational Application Developer	269
21-9	Start Tivoli Performance Viewer in Administrative Console	270
21-10	Tivoli Performance Viewer	271
21-11	Configuration panel for thread pool properties	272
22-1	InetInfo.java program output	276
22-2	Ethereal data analysis with Follow TCP Stream windows	278
22-3	IPCS CTRACE display parameters	280
22-4	TCP/IP network packet trace report	281
22-5	MXI Primary Option menu	282
22-6	CPU Activity Report (partial view)	285
22-7	Partition Data Report (partial view)	286
22-8	Partition Data report and processing weights (partial view)	287
22-9	Summary Report	288
22-10	Workload activity (part 1)	289
22-11	Workload activity (part 2)	290
22-12	Workload report for WebSphere server address space (partial)	291
22-13	Response time distribution (partial view)	292

22-14 Sample summary report from SMF Browser	297
22-15 WebSphere Studio Workload Simulator window	300
22-16 Pop-up window to start recording	301
22-17 WebSphere Studio Workload Simulator with scripts of captured sessions	301
22-18 WebSphere Studio Workload Simulator window: Web session elements	302
22-19 Variable elements through a filter	302
22-20 Various runtime parameters.	303
22-21 WebSphere Studio Workload Simulator Monitor GUI	304
22-22 Sample simulation graph	305
22-23 Microsoft Web Application Stress Tool	306
22-24 TeraTerm Pro.	307
22-25 Example of WS_FTP Professional)	308
22-26 UltraEdit	309

Archived

Tables

2-1	Problem severity levels	18
2-2	PMR numbers and what they indicate	23
5-1	WebSphere related codes	52
5-2	Abend reason code and explanation	52
18-1	Useful z/OS DISPLAY commands	200
18-2	TRACEBASIC/TRACEDETAIL codes	201
19-1	Parts of server log stream record output	219
19-2	Log Details Level	230
19-3	First line in trace sample	231
20-1	JDBC trace strings	244
20-2	Useful IPCS commands for formatting an SVC dump	248
21-1	Parameters and options for the dump utility	256
21-2	Modules and description to verify in Tivoli Performance Viewer	271
A-1	WebSphere for z/OS message formats	312
A-2	WebSphere for z/OS messages overview	312
A-3	BBOO0222I message components	313
A-4	WebSphere-related abend codes	315
A-5	Example abend code and related reason code	315
A-6	System and component messages	315

Archived

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 U.S.A.

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.

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.


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

COPYRIGHT LICENSE:

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

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

@server®	DB2®	Parallel Sysplex®
eServer®	DFS™	Rational®
Redbooks (logo)  ™	Informix®	Redbooks™
alphaWorks®	IBM®	RACF®
developerWorks®	IBMLink™	RETAIN®
z/Architecture™	IMS™	RMF™
z/OS®	Language Environment®	S/390®
zSeries®	Lotus®	System z™
AIX®	MQSeries®	Tivoli®
Cloudscape™	MVS™	WebSphere®
CICS®	OMEGAMON®	1-2-3®
DB2 Universal Database™	OS/390®	

The following terms are trademarks of other companies:

EJB, Java, JavaServer, JavaServer Pages, JDBC, JDK, JMX, JSP, JVM, J2EE, Solaris, Sun, Sun Java, SNM, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Excel, Microsoft, Visual Studio, Windows NT, Windows, Win32, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

Ethereal is a registered trademark of Ethereal, Inc.

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

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

Preface

This IBM® Redbook can help clients and IBM employees understand the different aspects of problem determination for IBM WebSphere® Application Server Version 6 for z/OS®. It is intended to be an additional resource to the Information Center at:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

How the book is structured

The authors have arranged the information in this book in four parts.

In Part 1, we provide an overview of problem determination methodology, list the skills that are needed to perform problem diagnosis, tell you where to find information about problem determination and related topics, and explain how to interact with IBM to obtain support when a problem occurs.

In Part 2, we present the most typical problem symptoms that you might encounter and the diagnostic flow charts that we have developed for each of these symptoms. These flow charts lead you through the problem determination steps that are required either to identify the problem, find the solution, or prepare the information that is required when you need to consult with IBM support.

In Part 3, we identify the possible problem areas that can be encountered in WebSphere for z/OS and we have arranged this information into four specific problem phases:

- ▶ Phase 1: Installation, configuration, and migration
- ▶ Phase 2: Application deployment
- ▶ Phase 3: Testing applications for the first time
- ▶ Phase 4: Production environment

We also discuss typical problems for each phase and provide a problem avoidance checklist.

In Part 4, we identify the problem determination means and tools that can assist you with day-to-day tasks and prevent problems. We introduce and explain:

- ▶ Especially helpful WebSphere commands
- ▶ Logs, traces, and dumps
- ▶ Diagnostic tools for problem determination

What this book is about

In this book:

- ▶ We provide you with a general approach to problem determination and problem source identification and show you how to use this approach for WebSphere Application Server for z/OS V6.
- ▶ We offer information to help you to work through problems that you might experience with WebSphere for z/OS V6 from the initial problem symptom through to the solution of the problem.
- ▶ We introduce you to the tools and resources available to help you identify problems and find solutions in the WebSphere for z/OS V6 environment.

Our goal is to help you become more self sufficient at diagnosing and resolving your WebSphere for z/OS problems.

We have collected actual problem scenarios that have been reported to IBM support and presented these in a standard format to help you find problem solutions. Even when a solution to your specific problem is not found in these scenarios, we hope that you can use the symptom flow charts, references, and tools in this book to help you work through your problem and identify the cause and possible solution.

If you are unable to find a solution to your problem, we also provide you with useful information to assist you when you communicate with IBM and WebSphere support teams.

Who this book is for

This book is intended for system programmers and administrators working with WebSphere for z/OS V6. It is targeted towards those who need to identify problems, analyze them, and fix them efficiently to deliver sound support for the WebSphere environment, including IBM support and technical professionals and Java™ developers who work in this environment.

How to use this book

This book is intended to be used as a reference. We have structured it so that each part or chapter can stand on its own. We have assumed that you will come across this book when searching for specific information about a problem that you have encountered in your WebSphere on z/OS environment.

For that reason, we researched symptoms and created flow charts for those symptoms to help you analyze problems and fix them or to prepare the documentation that is required to report a problem to IBM support. From the symptom chapters, you can use the links to the other parts in this book that direct you to related information.

You can also use this book to:

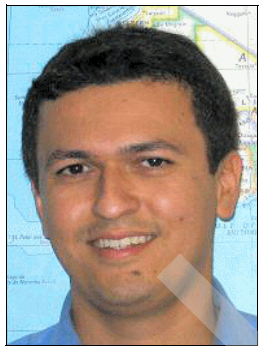
- ▶ Learn about new supporting means and tools or learn new tricks for performing problem determination for WebSphere for z/OS.
- ▶ Discover what other problems are likely in your particular phase of the WebSphere life cycle and how to tackle them.
- ▶ Learn about general problem determination methodology, other information resources, and how to contact IBM.

The team that wrote this redbook

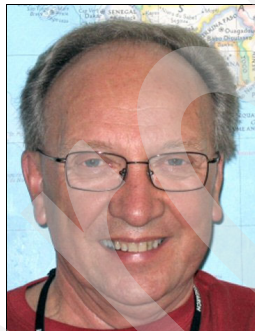
This IBM Redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.



Rica Weller is a Project Manager at the International Technical Support Organization (ITSO), working in New Zealand and the U.S. She was a Systems Engineer for IBM S/390® for two years and a Senior Consultant for IBM WebSphere Business Integration on z/OS in the Competence Center with IBM Germany for three years. She also taught classes, presented at several conferences, and coauthored several IBM Redbooks™ about WebSphere for z/OS and textbooks about zSeries® basics. Rica holds a degree in Business Administration from TU Dresden, Germany, and a Master Degree in Business from Massey University, New Zealand.



Cleberson Calefi is an IBM WebSphere consultant at Bank of Brazil. For the last three years, Cleberson has worked extensively with the WebSphere Application Server environment, advising clients about problem solving, tuning, and implementation of fail-safe runtime environments. His areas of expertise include J2EE™ application development and WebSphere Application Server administration for z/OS, z/Linux®, and Windows®. He holds a degree in Information Systems from the University Alvorada, Brazil.

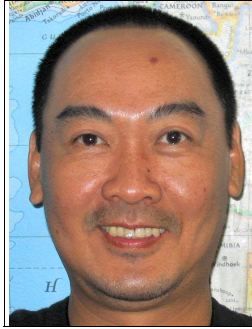


Per Fremstad is an IBM certified IT specialist from IBM Systems and Technology Group, Norway. He has worked for IBM since 1982 and has extensive experience with zSeries and z/OS. His areas of expertise include the Internet, the WebSphere product family, and Web enabling applications on z/OS. He teaches frequently about WebSphere and Java topics and about zSeries and z/OS at several universities. He holds a Bachelor of Science degree from the University of Oslo, Norway.



Keith Jabcuga is Software Support Specialist working at the ITSO in Poughkeepsie, New York. He has been on the WebSphere for z/OS support team for four years and his areas of expertise include defect support and application diagnostics. Keith holds a Master of Science degree in Computer Science from the University of Buffalo.

Suresh Maddukuri is an IT Specialist who assists customers in the United States. He worked as an administrator for WebSphere Application Server on z/OS and distributed platforms. His main responsibilities are troubleshooting problems, performance monitoring, and application server tuning. His areas of expertise include IBM WebSphere MQ and WebSphere Business Integration Message Broker. He holds a post-graduate diploma in Computer Applications and a degree in Mechanical Engineering from Nagarjuna University, India.



Kiet Nguyen is an IT Specialist with IBM Global Services/AMS CRM Siebel Development in North Carolina. He has more than 20 years of experience that ranges from MVS™ systems/application programming to building component software and end-to-end applications on distributed platforms for worldwide customers. He holds a degree in mathematics from Georgetown University in Washington, D.C. His areas of expertise also include J2EE Development and WebSphere Application Server Administration.



Robyn Nostalg is an IT Software Support Specialist working in the IBM Support Center in Sydney, Australia, and she has been in this role for more than 10 years. She has specialized in supporting customers that run WebSphere Application Server for z/OS. She has also worked on the zSeries Software Support team, providing defect and non-defect support for all software components related to the z/OS operating system.



Rajesh Pericherla is a system tester and lead strategist in the WebSphere for z/OS SVT in the WQCoC organization. He has been working with this group for the last eight years. He is responsible for planning the system tests for the latest WebSphere releases on all supported platforms with the main focus on z/OS. He holds a Master of Science degree in Computer Engineering from Walden University.

Special thanks to Bob St. John of zSeries Performance, IBM Poughkeepsie, for his additional chapter about performance problem analysis.

Thanks to all the contributors to the previous IBM Redbooks about WebSphere for z/OS Problem Determination, and the support teams of the IBM International Technical Support Organization, in particular (and in no specific order):

Ash Venkatramen, Keith Kopycinski, Tamas Vilaghy, Patrick C. Ryan, Andrew Lam, Youn Chin Mah, Ralph Schipani Jr., Brent Watson, Ron Allan, Dave Clarke, James Bai, Paola Bari, DongJune Choi, Mike Cox, Alberto González Dueñas, John Hutchinson, Wilhelm Michel, Theresa Tai, Egon Terwedow, Ella Buslovitch, Rich Conway, Don Bagwell, Keith Kopycinski, Nancy Trent, Michael Stephen, Hany Salem, Peter Bertolozzi, Melinda Carter, Mike Schwartz, Dave Griffiths, Christopher Vignola, Timothy Spewak, Stephen J Kinder, Mark Dinges, David Follis, Timothy Kaczynski, Teddy J Torres, Scott Kurz, Louis Wilen, Maria Clarke, Edward McCarthy, Kenneth Irwin, Al Schwab, Forsyth Alexander, Don Brennan, Tessa Nguyen, and many more working in z/OS and WebSphere for z/OS Development and Technical Support worldwide.

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners, and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

ibm.com/redbooks

- Send your comments in an email to:

redbook@us.ibm.com

- Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Archived



Part 1

Problem determination methodology

Businesses today are becoming increasingly dependent on IT. When a business experiences a problem with their IT systems, the impact can be devastating. It therefore becomes critical to have the information, tools, and support available to help identify the type, source, cause, and solution.

In this part of this book, we discuss a general problem determination methodology and how it applies to the IBM WebSphere Application Server for z/OS environment. We review the skills that are necessary for performing problem diagnosis for WebSphere for z/OS and provide information that can assist you when you communicate with IBM support. We also provide you with additional information and resources all aimed at helping you find a solution to your problem.

Archived

Problem determination methodology

Problem determination is not unique to the software industry. A doctor, for example, uses a problem determination process with a sick patient. After identifying initial symptoms and asking the patient questions to gain a better understanding of the problem, the doctor orders tests, analyzes results, and sometimes consults with a specialist.

Similar to the human body, IBM WebSphere Application Server for z/OS can experience problems that must be diagnosed and corrected. The steps that you take during the problem determination process can help you define and identify the problem and find a solution.

In this chapter, we discuss a general approach to problem determination methodology and how it applies to a WebSphere for z/OS environment. We explain how to analyze a problem and what steps can be taken to find the cause and solution. We discuss the skills needed for WebSphere for z/OS problem determination.

1.1 What problem determination is

The term *problem determination/problem source identification* is used to describe the broad topic of finding the answers to the questions “what went wrong?” and “why did it go wrong?” when there is a problem with a system. This means identifying which component of the system is responsible for causing the problem.

The goal of problem determination/problem source identification, in its most basic sense, is to get to the root of a problem. It is similar to what a programmer might call debugging, but on a much larger scale. Problem determination/problem source identification includes debugging applications, but it also involves diagnosing the system at large by investigating product configurations and verifying the means by which all of the system components interact.

When a problem or system is complex, then you need to adopt a more structured and systematic approach to determine what caused the problem. You can follow the process shown in the general problem determination flow chart that we describe in 1.3.1, “Problem determination flow chart” on page 4.

1.2 What problem determination is not

There are many symptoms that an enterprise system can show that are often classified as problems. Poor system performance, for example, can definitely be a problem. However, the process of checking and solving performance problems is often referred to as tuning. Tuning a system involves its own separate set of tools and processes.

Understanding the difference between problem determination/problem source identification and tuning is very important and knowing when to use which can save you a lot of time. Problem determination/problem source identification fixes functional problems. Tuning means adjusting the system and the application; it handles problems that are associated with slow processes.

Even though tuning falls outside the area of problem determination, we have provided basic information about performance issues and where to find links and information about it to make this book as complete and helpful as possible. See Chapter 13, “WebSphere for z/OS performance analysis” on page 117, and for more information about performance monitoring, refer to *Monitoring WebSphere Application Performance on z/OS*, SG24-6825.

1.3 Problem determination approach

In this section, we describe the steps involved in the problem determination process in general terms. First, we use a flow chart to graphically track our problem determination process. Then, we describe each step in the process, what is involved, and what should be considered at each step.

1.3.1 Problem determination flow chart

The flow chart in Figure 1-1 on page 5 shows the steps, decisions, and flow of the problem determination process.

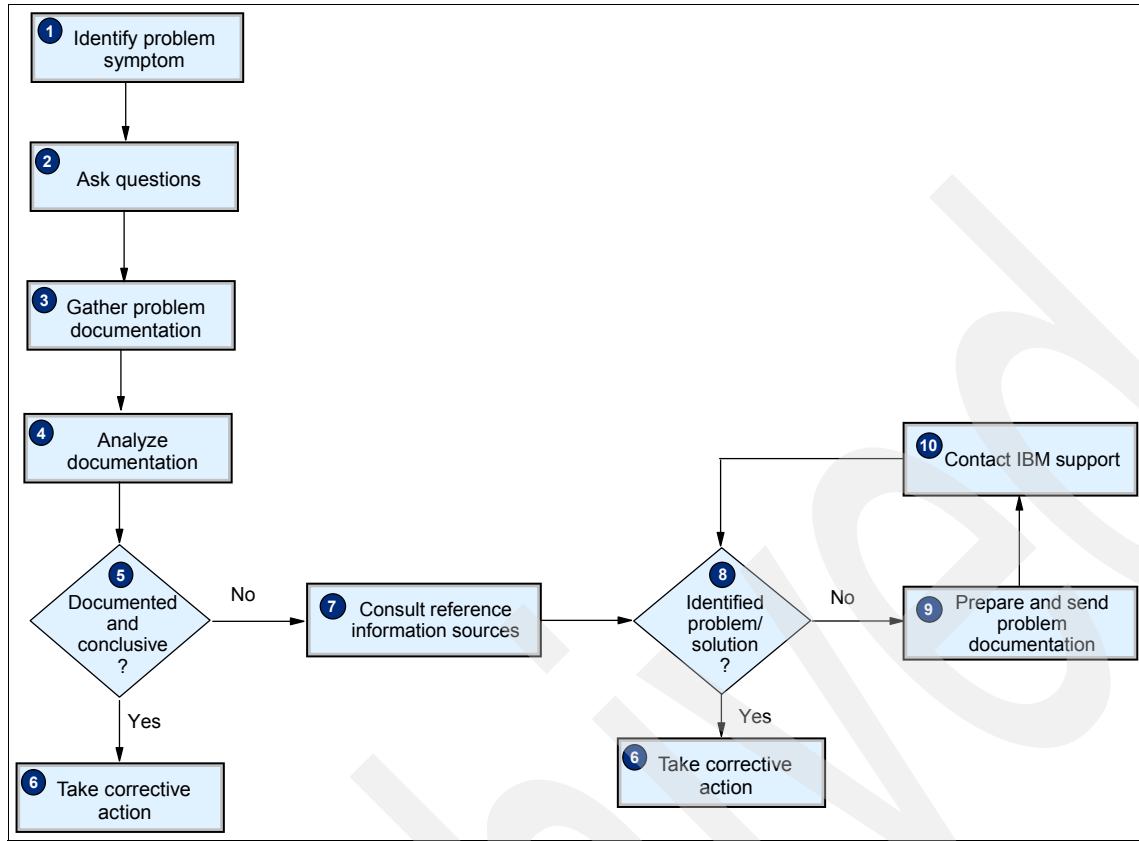


Figure 1-1 General problem determination flow chart

1.3.2 Problem determination process

The tasks outlined in the flow chart in Figure 1-1 are expanded as follows:

1. Identify the problem symptoms.

Every time a software problem occurs, some kind of indication is given about it. It might be an error message, a wrong output, an abend, no response or bad response times, or a message that is returned by the browser. These are called *symptoms*.

Before you can begin to solve a problem, you need to know what type of problem you have. To start the problem determination process, you identify the symptoms of the problem by describing the symptoms. If there are multiple problems, try to separate them and deal with them independently. Be careful about assumptions.

It is very common for more than one condition to exist for each problem. The more complicated the scenarios are, the more likely it is that a combination of problems lead to the symptoms that you are experiencing. Therefore, always keep an open mind when performing problem determination.

2. Ask questions.

During this step, you identify the background or supporting information for the problem and you do this by asking questions. A good place to start is to identify whether any recent changes have taken place in the system. The questions to consider are:

- Has there been a change to the software operating system or the application server software?
- Has there been an upgrade, new maintenance level, or initial program load (IPL)?

- Have there been any changes to the environment such as network topology, hardware configuration, or increase in the number of users?
- Have we made changes to the back-end systems that we are connecting to?
- Have any new applications been deployed, changed, or upgraded?
- Have we run this application, server, or system successfully before?
- When did the symptoms first appear? When the system was under peak load? After backup jobs?
- Can we reproduce the error?

Asking these types of questions can help you eliminate potential causes early in the process. The answers to the questions form part of your symptom data.

3. Gather the documentation.

The type of information that you gather depends on the type of symptoms that you are experiencing, but essentially what you are doing at this stage is collecting evidence of the problem. So, if the symptom is an error message, then you must obtain and examine the log or trace that shows the message.

We recommend that you complete the following tasks as part of the gathering process:

- Document your problem determination steps. Keep a log of symptoms, messages, files, tests, results of tests, and conclusions.
 - Retrace the steps to recreate the problem and see the results yourself.
 - Understand the meaning behind the request that has created or that has induced the problem. This helps you isolate the problem.
 - Use a controlled test environment when possible.
 - Use the MustGather documentation for information about what data to collect.
- MustGather* is a term that is used to describe the essential or minimum problem documentation that is required to analyze a problem. See “MustGather” on page 16.

Knowing what data to collect and how to collect it can be difficult. We guide you through this in Part 2, “Problem symptoms and their resolutions” on page 39 where we analyze symptoms in detail with the use of symptom-specific flow charts.

Sometimes your problem is very serious and your expertise in the product area is limited. Then you might choose to go directly to the step of calling IBM support rather than try to analyze your own trace or dump.

4. Analyze the documentation.

The documentation that you obtain will depend on the type of symptom and what is enabled in your system. Some output is available by default; others, like traces, might have to be enabled. How to enable the different output received from an error is covered in Part 4, “Problem Determination Means and Tools” on page 193.

Symptoms such as *abend*, *loop*, and *incorrect output* are often accompanied by messages, or you can find indications in traces and logs. Check the data that you have collected for messages or other indications. Analyze the messages, logs, and traces.

Part 2, “Problem symptoms and their resolutions” on page 39 discusses what output is available given a particular symptom and how to analyze it.

5. Determine whether the problem is recorded in product documentation and, if so, what corrective action is recommended.

In this step, check the product documentation, such as the product reference manuals and product Web sites, to determine whether your problem is documented. In the case of an

error message, the product documentation might describe the reason for the error and offer possible corrective action.

6. Take corrective action.

The action you take depends on the cause and the recommended solution to your problem. The typical outcome or action that you can take falls into the following categories:

- The product works as designed.

In this case, you can accept the design and adjust your system accordingly, or you can request a design change. This is an official request to change the product design that is assessed by technical staff, usually the product developer.

- You find a workaround for your problem.

This means that changes must be made to your WebSphere for z/OS system to circumvent the problem. In some cases, this workaround is the solution. In other cases, it is a temporary solution until a permanent fix is found.

- You find a problem scenario or symptom that is described in an authorized program analysis report (APAR).

In this case, you apply the fix (program temporary fix, or PTF) associated with the APAR to correct the problem.

- If your problem scenario or symptom is not found on the WebSphere for z/OS IBM support page, consider these possibilities:

- This is a new WebSphere Application Server for z/OS problem, which should be reported to IBM so that they can produce a fix.
- It is a user error. This includes configuration, setup, or procedural error. This must be corrected by the user.
- It is an application problem. This should be presented to the application owner or developer to correct.

7. Consult reference information sources.

Information sources come in many forms: a product manual, a Web site that contains links to product fixes, a colleague with specialized skills, an online technical forum, and IBM software support. Refer to Chapter 3, “Information sources” on page 25, where we outline what information sources are available to you and how you can get access to them.

If your symptom is an error message, check the meaning of the message in the product manuals because this might point you to the exact cause of the error and tell you what is required to fix it. If not, you can access IBM support data and search for your symptoms in hopes of finding other, similar problems reported by customers. These problem records can tell you what was done or recommended to fix the problem.

8. Identify the problem and solution.

Using the information sources, you might have identified the problem and found a solution and can now take corrective action. If you have not been able to identify the problem or find a solution, you need to prepare and gather the problem determination documentation.

9. Prepare and send problem documentation.

If, after consulting your information sources, you are unable to determine the exact problem or to identify the cause, then you should forward all problem documentation to IBM support.

10. Contact IBM support.

Refer to Chapter 2, “Contacting IBM: Information” on page 13 for the options that are available when you must contact IBM. We also explain the WebSphere support teams and structure.

1.3.3 Specific considerations for WebSphere for z/OS problem determination

Keep these points in mind when you are working with WebSphere for z/OS problems:

- ▶ WebSphere for z/OS is a complex software product that involves many z/OS components and therefore requires intensive system administration.
- ▶ Many of the environment parameters and variables for WebSphere for z/OS components must be set to a specific value.
- ▶ User-set WebSphere for z/OS components require consistency throughout your environment.
- ▶ Not all problems are related to WebSphere for z/OS. Consider all variables in your specific z/OS environment to eliminate those that are not relevant.

We suggest that you always keep detailed, up-to-date versions of the following items when you are working on a problem:

- ▶ Your network topology records
- ▶ A high-level application description
- ▶ A detailed model of your application
- ▶ A detailed model of how your WebSphere for z/OS application interacts with other IBM products, tools, or third-party software
- ▶ A log of your WebSphere for z/OS setup
- ▶ Fixes that have been installed for WebSphere for z/OS and other components that interact with your WebSphere for z/OS system, such as WebSphere MQ and others
- ▶ A log of your hardware specifications

Keep in mind that a history log of changes works much better than a simple log of the current environment conditions. It is always best to retrace your steps up to the point of failure.

1.3.4 The importance of a test environment

There are two major types of environments where problems can be identified:

- ▶ Test environments
- ▶ Production environments

Test environments are usually easier to debug because they can be changed easily without any business impact. If possible, we recommend that you try to use your test environment for problem determination/problem source identification and to test the solutions.

Debugging production environments has an impact on your business and is therefore much more difficult. Introducing change to this environment can create interruptions or disturbances in your business. Therefore, if you must debug a production environment, it is important to apply thoughtful problem determination/problem source identification methodology to your components and configurations that is based on symptoms, causes, and analysis to find logical solutions for your problems.

1.4 The skills needed for WebSphere for z/OS

Properly managing and administering a WebSphere for z/OS system requires expertise in many different areas (Figure 1-2). When everyone involved works closely together, you can avoid many problems and save time and money.

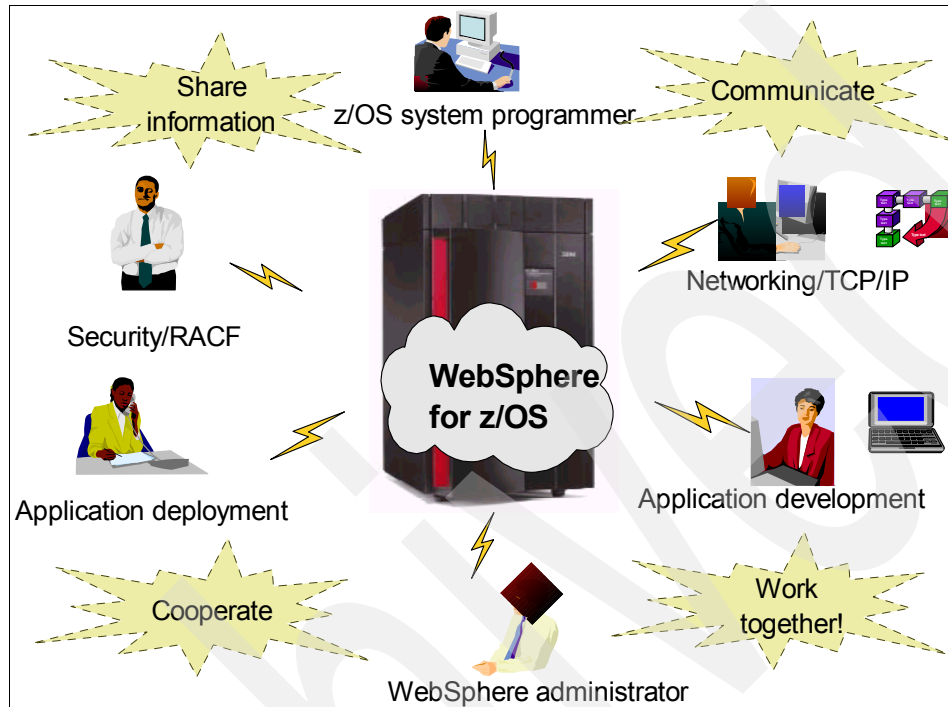


Figure 1-2 Working together

You should also ensure that there are sufficient systems programming and application deployment skills and experience, because WebSphere for z/OS utilizes most of the advanced features and functions of the operating system. A list of these functions is available in *WebSphere Application Server for z/OS Version 6.0.1: Migrating, coexisting, and interoperating*, SA23-2207. You must have systems programming skills in all of these areas. If you try to set up the WebSphere run time without good skills or assistance in these areas, you are likely to experience many frustrating problems and delays.

See 3.6, “Educational information” on page 36 for resources that can help you improve your skills for WebSphere for z/OS problem determination.

1.4.1 System skills

The following “traditional” skill areas are critical for successfully installing and establishing a WebSphere for z/OS run time:

- ▶ **z/OS:** To install software products and related prerequisites and to set up required OS resource definitions and settings.
- ▶ **UNIX® System Services:** To set up a functional Hierarchical File System (HFS) and UNIX environment.
- ▶ **TCP/IP:** To configure connectivity for WebSphere clients and servers.
- ▶ **Resource Access Control Facility (RACF®) or equivalent:** To authenticate WebSphere clients and servers and authorize access to resources.

- ▶ **Logger:** To set up log streams for Resource Recovery Services (RRS) and the WebSphere error log.
- ▶ **Parallel Sysplex®:** To implement multi-system configurations.
- ▶ **RRS:** To implement RRS and support two-phase commit transactions.
- ▶ **Automatic Restart Manager (ARM):** To set up an automation process for stopping and starting the WebSphere runtime environment. Although deemed to be optional, it is crucial to have all operational processes automated in a multiple logical partitions (LPARs), multiple application servers environment.
- ▶ **WebSphere:** To customize and set up WebSphere administrative servers and application servers and configure WebSphere resources as required by your application.

1.4.2 Skills for deploying and running an application

Beyond the traditional skills, you need people with skills in developing, deploying, and running the applications, although the development is usually done by third-party vendors.

The people who are deploying and running the application should:

- ▶ Understand the WebSphere for z/OS structure.
- ▶ Understand the J2EE architecture (for example, understand how the deployment descriptor fits into their WebSphere for z/OS configuration).
- ▶ Be able to use Application Server Tool Kit (ASTK) or WebSphere Studio to modify deployment descriptors; to view the layout of Java Archive (JAR) files, Web Archive (WAR) files, and more; and to create Enterprise Archive (EAR) files.
- ▶ Be familiar with and be capable of setting trace settings for WebSphere to trace various components (for example, the Web container, EJB™ container, class loader, and so forth).
- ▶ Be able to use the Administrative Console or WSAAdmin scripts to create J2EE server instances and install J2EE applications.
- ▶ Understand the output of the job log, the WebSphere for z/OS error log, and the trace output of runtime and J2EE servers as they relate to troubleshooting deployment problems.
- ▶ Work closely with system programmers for IBM Workload Manager (WLM) configuration, and creating procedures for J2EE server instances.
- ▶ Work closely with the security administrator to define user ID and groups for J2EE controller regions and servant regions as well as the EJB security environment.
- ▶ Work closely with application developers for defining data sources, resolving external EJB references, understanding application logic, and so forth.
- ▶ Be familiar with UNIX System Services (USS) and comfortable using the shell.
- ▶ Be familiar with TCP/IP setup and be able to use commands such as **tracert**, **netstat**.

Optionally, they should:

- ▶ Work closely with the DB2® administrator, WebSphere MQ administrator, and others when there are problems with data sources.
- ▶ Be familiar with Lightweight Directory Access Protocol (LDAP), be able to understand LDAP traces, know how to use the LDAP browser, and so forth.

Figure 1-3 on page 11 illustrates application development and deployment. You should always be aware of all these dependencies.

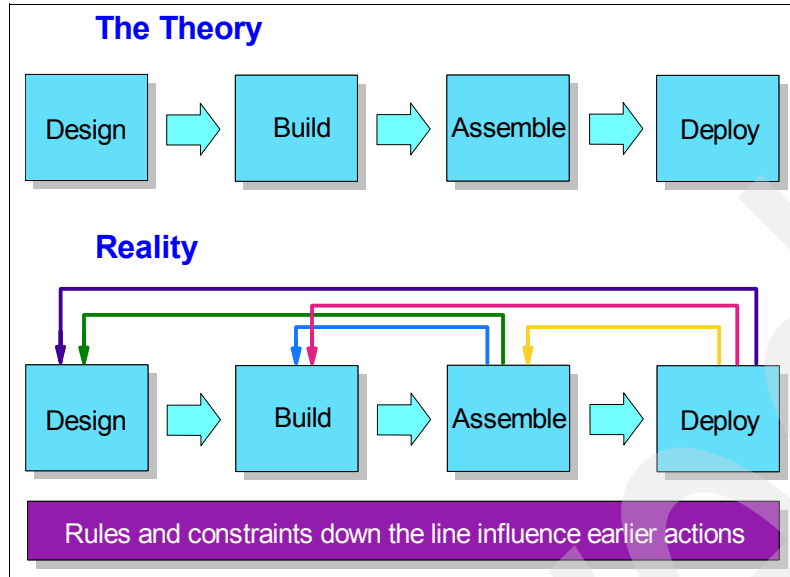


Figure 1-3 Deploying applications

It is unlikely that any one person can possess all these skills. It takes a team of specialists to set up the WebSphere run time and run the server. For specific courses and an organized view of the curricula, see the class catalogs at:

<http://www.ibm.com/services/learning/>

Archived



Contacting IBM: Information

This chapter provides information that can assist you when you communicate with IBM Support. We describe the WebSphere Application Server for z/OS support team structure and show you how to place a Problem Management Record (PMR) with IBM support. We also show you how to exchange information and send documentation to IBM support teams.

2.1 Communicating with IBM

Customer experiences with IBM support departments show that some critical situations in handling and solving a software problem are results of a lack of communication and information. We want to contribute to filling this gap.

How is IBM support structured? What should you do first if a software problem is encountered? What information and data are needed? How do you report this problem to IBM? These are the questions we answer in this section.

Many of these items are an excerpt of what you can find in the IBM Support Guide where much more information is available, such as:

- ▶ Software Support Handbook Overview
- ▶ Enhanced Support
- ▶ Contacting IBM
- ▶ No Support Contract
- ▶ Preventing Problems
- ▶ Support Contacts (worldwide)
- ▶ Additional Offerings
- ▶ Acronyms and Other Terms (really helpful for finding IBM terms)

You can find the Support Guide at:

<http://techsupport.services.ibm.com/guides/handbook.html>

2.2 The IBM WebSphere support structure

Figure 2-1 is a high-level view of how IBM support is structured.

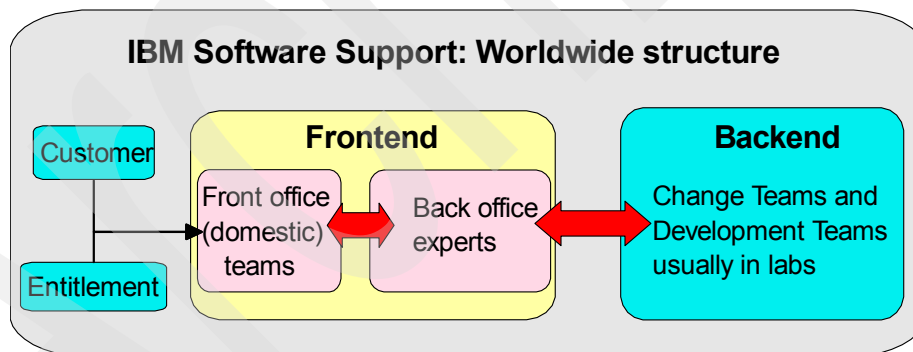


Figure 2-1 IBM support structure

If a problem is reported in a Problem Management Record (PMR), this usually happens through the problem-entry help desk or front-office teams. The next level is the front-end support personnel, who usually have broader skills with IBM software products and a national language approach.

If more in-depth skills are needed, the back-end becomes involved (for example, IBM software laboratories), where the software is developed and necessary code changes are made.

Communication between front-end and back-end support works very well because of the worldwide IBM network and the fact that the teams usually know each other well and use all communication vehicles.

2.3 Before you contact IBM support

There are various means for obtaining support for your specific problem. You can benefit immediately from the extensive IBM self-help support Web site, where you can download fixes, use keywords for searches, find how-to information, and possibly solve your problem, all before contacting IBM support directly.

The latest information about getting support for WebSphere z/OS can be found at:

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

Tip: Use component ID 5655I3500 as one of the keywords for your search. This reduces the search results to only WebSphere Application Server for z/OS problems and APARs.

The following sections describe actions that you should take before contacting IBM Support Center. Most of this information is also described in *Steps to getting support for WebSphere Application Server*, available at:

<http://techsupport.services.ibm.com/guides/handbook.html>

2.3.1 Defining the problem

Being able to articulate a problem and symptoms before contacting software support will expedite the problem-solving process. It is very important for you to be as specific as possible when you explain a problem to or ask IBM software specialists a question. The specialists want to be sure that they provide you with exactly the right solution, so the more they understand your specific problem scenario, the more likely they are to resolve it.

You should first try to recreate the problem. Document the following information:

- ▶ The steps you took to recreate the problem and any symptoms or error messages that you observe such as:
 - Date and time
 - User name or user ID involved
 - LPAR name, server name, job name, and so on
- ▶ Recent changes that have been made to your processing environment, such as hardware or software that has been added or removed
- ▶ System configuration updates

Be aware that you should report only one problem or question per PMR. This avoids confusion and misunderstandings about the case that is reported in the PMR.

2.3.2 Determining whether this situation has already been reported

The problem might be documented and resolved already, so check these product support resources to see whether the answer you are looking for is available:

- ▶ Information Centers and release notes

Information Centers provide fast, centralized access to WebSphere Application Server product information that is available in multiple languages and updated regularly. The problem might also be documented in the release notes and in the readme file that is packaged with the product.
- ▶ Software and hardware prerequisites

Verify the product release and major update requirements for the software that you are running (WebSphere Application Server for z/OS and its requirements).

- ▶ WebSphere Application Server related product support

Access APARs, Technotes, and PTFs; register to receive e-mail notifications about technical alerts or new downloads; and use an advanced search feature that searches all IBM knowledge bases, such as redbooks and Information Centers.
 - ▶ MySupport

Register to receive e-mail notification about critical issues, IBM product updates, and items of interest.
- 11.Link2000
- For IBM @server® zSeries users with installations that have access to Link2000 (previous IBMLink™), an interactive online database program, you can:
- Search for an existing authorized program analysis report (APAR) that is similar.
 - Search for an available program temporary fix (PTF) for the existing APAR.
 - Order the PTF if it is available.
- ▶ developerWorks® WebSphere

This gateway to WebSphere technical information for developers and administrators features:

 - Zones and road maps for specific products, in-depth technical articles, tutorials, white papers, and links to downloads, technical previews, steps to getting support for WebSphere Application Server, and plug-ins
 - Latest news about WebSphere products and offerings

2.3.3 Gathering background information

This section introduces sources that can help you gather background information.

MustGather

MustGather documents help with problem determination and save time when you are resolving PMRs. These documents, which are located on the product support site, include instructions about what documentation to gather for specific problems. You can find MustGather documents by searching for the word *mustgather* at the support Web site:

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

These are some of the MustGather documents for WebSphere for z/OS that might help you:

- ▶ MustGather: Read first for WebSphere Application Server for z/OS
- ▶ MustGather: High CPU causing hang or loop running V5 for z/OS
- ▶ MustGather: Plug-in regeneration problems for V5.0 and V5.1
- ▶ MustGather: Plug-in problems in V5.0 and V5.1 on z/OS
- ▶ MustGather: System management for synchronization failures
- ▶ MustGather: System management discovery problems
- ▶ MustGather: wsadmin problems in V5
- ▶ MustGather: Administrative console problems
- ▶ MustGather: A hang occurs when running WebSphere Application Server for z/OS
- ▶ Mustgather: Security problems with WebSphere Application Server z/OS V5
- ▶ MustGather: ABENDEC3 RC=413000x for 4.0, 5.0 and 5.1 for WebSphere Application Server for z/OS

Collector Tool

IBM WebSphere Application Server, Version 6.0.x on AIX®, HP-UX, Linux, Sun™ Solaris™, and Microsoft® Windows provides a collector tool that you can use for z/OS as well. Run it for all application servers and the deployment manager. The collector tool gathers extensive information about your WebSphere Application Server environment and packages it in a JAR file that you can send to IBM support to help determine and analyze your problem.

Information in the JAR file includes logs, property files, configuration files, operating system and Java data, and the absence or level of each software prerequisite. The collector program runs to completion despite any errors that it might find. Errors might include missing files or commands. The collector tool collects as much data in the JAR file as possible.

The collector tool has two phases. The first phase runs the collector tool on your WebSphere Application Server and produces the JAR file. The IBM support team performs the second phase, which analyzes the JAR file that the collector program produces. For more information about the collector tool and how to run it, search for “Gathering information with the Collector tool” in the WebSphere Application Server, Version 6.0.x, Information Center at:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

Tip: Entering “Collector-summary” output into an electronic service request (ESR) eliminates waiting on the phone to provide general information to IBM support level 1.

Other necessary information

To solve a problem effectively and efficiently, the software specialist must have all of the relevant information. Being able to answer the following questions can help IBM support in their efforts:

- ▶ What levels of software were you running when the problem occurred? Include all relevant products, including the operating system. Typically, this information should include:
 - Operating System version
 - WebSphere Application Server (Base or ND) version
 - Java run time (JDK™ or JRE) version
 - Host Security software (RACF, ACF2 or TopSecret) version
 - Optionally: DB2, MQ, HTTPD, and others
- ▶ You should mention major configurations such as the following, if relevant:
 - Monplex or Sysplex
 - Global security
 - Clustered or non-clustered application
- ▶ Has the problem happened before, or is this an isolated instance?
- ▶ What steps led to the failure?
- ▶ Can the problem be recreated? If so, what steps are required?
- ▶ Have any changes been made to the system (hardware, network, or software)?
- ▶ Were any messages or other diagnostic information produced? If yes, what were they (for example, trace record or dump output)?

It is often helpful to have a printout of the message ID numbers for any messages that you received before calling IBM.

The most common data that IBM requests includes:

- ▶ **System log (SYSLOG):** The z/OS system log, which has assorted system error messages and a few WebSphere error messages.

- ▶ **WebSphere Server job logs:** Application server job logs contain most of the configuration settings, stderr and stdout messages, and CEEDUMP and snap dumps. Job logs for deployment manager, node agent, and daemon servers might be required, depending on the nature of the problem.
- ▶ **WebSphere error log:** Target for WebSphere error messages.
- ▶ **Dump data sets:** If a system abend occurred, a JVM™ transaction dump and SAN Volume Controller (SVC) dump might be captured. In most cases, IBM asks you to compress (terse) and FTP them to designated FTP site. In situations where you must cancel a servant region to overcome a problem, be sure to request an SVCdump when you cancel.
- ▶ **Component trace (CTRACE) message log:** If more detailed information is required, IBM support asks you to have the component trace writer and debug turned on to display more detailed trace information in the message log.

We describe how to obtain this data in Chapter 19, “Logs for problem determination in WebSphere for z/OS” on page 213 and Chapter 20, “WebSphere for z/OS traces and dumps” on page 241. You can also find more information at the WebSphere for z/OS Information Center:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

Define your technical question in specific terms and provide the version and release level of the products in question.

2.3.4 Determining the business impact

You assign a severity level to the problem when you report it, so understanding the business impact of the problem is important. Table 2-1 shows descriptions of the severity levels.

Table 2-1 Problem severity levels

Severity Level	Definitions	Examples
1	Critical business impact. This indicates that you are unable to use the program, resulting in a critical impact on operations in a production environment. This condition requires an immediate solution.	<ul style="list-style-type: none"> ▶ No users of Tivoli® Problem Management can register a call. ▶ An application server running business-critical applications crashes and cannot be started again.
2	Significant business impact. This indicates that the program is usable but is severely limited.	All users of Tivoli Problem Management receive a database manager error when they try to view open problems.
3	Some business impact. This indicates that the program is usable but less significant features (not critical to operations) are available.	A client cannot connect to a server.
4	Minimal business impact. This indicates that the problem causes little impact on operations or that a reasonable circumvention has been implemented.	Documentation is incorrect.

2.4 How IBM Software Support handles your problem

Several options are available for submitting your problem to IBM support:

- ▶ You can telephone your local IBM support directly, and the person who takes your call creates a PMR with the information that you provide. You can find a list of IBM service numbers at:
<http://techsupport.services.ibm.com/guides/contacts.html>
- ▶ You can use the Web problem submission tool to submit an ESR at the following site:
<http://www.ibm.com/software/support/probsub.html>
- ▶ You can use IBMLink 2000 to submit your own electronic version of the PMR. You must be a registered IBMLink 2000 user to use this option. IBMLink 2000 is also referred to as servicelink. For more information, go to:
<https://www.ibm.com/ibmlink/link2/logon/logonPage.jsp>

Regardless of which option you choose, all software support calls for z/OS software products are recorded in the IBM Remote Technical Assistance Information Network (RETAIN®) system. This system, which is used worldwide by all of the support teams, is a very effective communication tool for IBM support teams.

The advantage of placing an electronic call using ESR or IBMLink 2000 is that you can view the updates in the record and monitor the status of your request.

2.4.1 The PMR

When your problem is logged, a unique PMR or Incident/Support Case is created. This record number is allocated regardless of how the problem is submitted.

Make note of this PMR number, Incident number, or Support Case number, and use it in any future communication about this issue with the support center.

Your PMR, Incident, or Support Case is routed to a resolution team for handling. You might be transferred directly to the resolution team, or your issue might be placed in a queue for a return call.

To open a PMR, you must provide the following information:

- ▶ IBM customer number
- ▶ Contact name, phone number, and e-mail address
- ▶ Operating system name and version
- ▶ The product name and the component ID of the product

Tip: The component ID for WebSphere Application Server for z/OS V6 is 5655-N01.

2.4.2 Investigating a PMR

At the resolution team level, your call is researched, resolved, or escalated as appropriate.

Due to the level of specialization that is required to maintain superior technical expertise at the team level, it is sometimes necessary to involve more than one support team to resolve a particular software problem. IBM support teams are all networked and work as one to resolve whatever problems or issues arise.

To investigate the issue, IBM support might require access to information about your system that is related to the failure. IBM support might also try to recreate the failure to obtain

additional information or the support team might request other problem data or material during the life of the problem. If the problem is very difficult, to rule out possible causes, it might be necessary for IBM support to gather different data, such as traces, to isolate the problem.

If your problem is related to configuration, you might have to recreate the problem to obtain necessary information.

During this investigation process, the resolution team determines whether your defect issue falls into one of three categories:

- ▶ It is the result of a software defect that has been reported previously.
A fix or workaround is provided to circumvent or correct the issue. If none is available and it is determined that one is required, the resolution team works with you to find the best feasible workaround. The resolution team advises you when the defect APAR is closed, assists with the implementation of the fix, and updates your problem record.
- ▶ It is the result of an IBM software defect that has not been reported before.
The Resolution Team works with you to create an APAR or Software Problem Report (SPR) to track the resolution of the defect. These APARs and SPRs are routed to the appropriate development teams.
Because of the complexities of the environments supported and the development, verification, and testing resources required, defect fixes might require an extended period of time for resolution. For high impact problems, the resolution teams make every effort to develop a workaround that you can use until your APAR or SPR has been resolved.
- ▶ It is a problem that is not related to a defect.
If the problem is not related to a software defect in supported IBM code, then the Resolution Team might seek a solution only at the request of the customer under a separate service agreement.

2.4.3 How technical questions are handled by IBM

With technical question support, you can receive answers from IBM to product-specific, task-oriented questions that are related to the installation and operation of currently supported IBM software. In the course of providing answers to your technical questions, support personnel might refer you to product documentation or publications or they might be able to provide a direct answer to your questions about:

- ▶ Installation
- ▶ Usage (how-to)
- ▶ Documented functions
- ▶ Product compatibility and interoperability
- ▶ Technical references to publications (Redbooks, manuals, and so on)
- ▶ Publication interpretations
- ▶ Configuration samples
- ▶ Planning information for software fixes
- ▶ IBM database searches

IBM Software Maintenance and Support Line are not structured to address questions about performance, consulting, or extensive configuration. Additional telephone and on-site support services are available to meet these needs.

For further information about these services, contact your IBM representative, who can help direct you to the persons who can discuss your needs. This could be handled, for example, by consultants and architects in the IBM Technical Support Offerings area.

2.5 Exchanging data with IBM by FTP

The most common and usually preferred way to exchange data with IBM is to send it to an FTP server. This section describes the tasks for ensuring smooth data transfer.

2.5.1 Copying the job log into a z/OS data set

If you obtained a dump or trace, this is written to a z/OS data set or to the job log. There are two ways to copy a job log into a data set. You can use a normal TSO command such as:

```
TSO OUTPUT jobname(jobnumber) PRINT(TEST.DATASET) KEEP
```

Unfortunately, this function is often limited to your own user ID, and you can only print jobs that are in the HELD OUTPUT queue. Therefore, it might be better to use the System Display and Search Facility (SDSF) based function.

You can either use the XDC action line command on any SDSF panel (**x - output, d - dataset, c - close after print**), or you can go into the actual system output (SYSOUT) file and use the **print** command on the command line.

Open the print with the **print odsn datasetname** command first. Afterwards, you can use the **print** command without any options to print the complete file, or use **print 200 400** to print a specific range of lines. Type **help print** and press **Enter** for more details and options.

Use **print close** or leave SDSF to close the print file. Otherwise you can amend the file with other **print** commands.

2.5.2 Compressing the data

You should make sure that large files or data sets are compressed before sending them to any FTP server. Compression decreases the size of the file and the amount of time necessary for sending the file to an FTP server. Be conscious of the binary/ASCII settings and send your file/data set in the proper format. You should always send compressed data in binary instead of ASCII format.

There are three possible ways to compress data:

- ▶ TRSMAIN/Packlib
- ▶ The tar command
- ▶ ZIP files

TRSMAIN/Packlib

Using TRSMAIN (also known as Packlib) for compressing data is the most common method of compressing files for the z/OS environment. A big advantage of data sets in *tersed* (packed) format is that the data set attributes are stored and the file can easily be uploaded and *untersed* (unpacked) in another z/OS system without guessing the DCB parameters.

You can download TRSMAIN from:

<ftp://ftp.software.ibm.com/s390/mvs/tools/packlib/>

If you send a data set using TRSMAIN, be sure to provide IBM support with values such as LRECL, RECFM, BLKSZ, and space requirements.

After installing TRSMAIN, use the sample job control language (JCL) shown in Example 2-1 on page 22 as a basis for creating your own job with proper modification to *PACKLIB_PDS*, *&input_dataset*, *&tersed_output* to compress *&input_dataset* into its compressed format.

Example 2-1 Job to invoke TRSMAIN

```
//PACKIT JOB 'ACCOUNTING INFORMATION',NOTIFY=&SYSUID.  
//*****  
//* *  
//* TRSMAIN with PACK option *  
//* *  
//*****  
//JOB LIB DD DISP=SHR,DSN=&PACKLIB_PDS  
//STEP EXEC PGM=TRSMAIN,PARM=PACK  
//SYSPRINT DD SYSOUT=H  
//INFILE DD DISP=SHR,DSN=&input_dataset  
//OUTFILE DD DISP=(NEW,CATLG),UNIT=SYSDAL,  
// DSN=&packed_output,  
// SPACE=(CYL,(ppp,sss),RLSE)
```

JOB LIB DD can be eliminated if *&PACKLIB* is included in the LNKLIST concatenation. The *&input_dataset* in *INFILE DD* must be modified with the proper name of the data set that needs to be compacted, and *&packed_output* in *OUTFILE DD* must be modified with the data set name accordingly. The *ppp* and *sss* are the primary and secondary spaces for the output data set. You can also do that using an ISPF dialog by entering the program name in the command line and entering information in the fields of the related panels.

The tar command

Files in any HFS directory in USS can be compressed using the **tar** command. Here are two examples of how to use it:

- ▶ This command takes a directory and places it in an archive in compressed format:

```
tar -cvzf archive directory
```
- ▶ To identify all files that have been changed in the past week (seven days) and to archive them to the /tmp/posix/testpgm file, enter:

```
find /tmp/posix/testpgm -type f -mtime -7 | tar -cvf testpgm.tar -  
-type -f
```

-type -f tells **find** to select only files. This avoids duplicate input to **tar**.

Read more about this command in *z/OS V1R3.0 UNIX System Services Command Reference*, SA22-7802-03.

ZIP file

This format is especially relevant for personal computer files. You can also put multiple files in a so-called ZIP archive. The most common programs used for this approach are Winzip and PKZip. These can be found at:

<http://www.winzip.com>
<http://www.pkzip.com>

Pack the files into a ZIP archive that is executable (EXE file) so that the recipient of the file can extract it properly without having Winzip or PKZip installed.

2.5.3 Finding specific FTP instructions

The different IBM geographies and regions provide you with specific information about where and how to send your data. IBM support personnel are most likely to request that data be sent to the Enhanced Customer Repository (ECuRep) FTP site:

<ftp://ftp.emea.ibm.com>

The following site provides additional information about file upload and download procedures:

<http://www.ibm.com/de/support/ecurep/mvs.html>

2.5.4 Using naming conventions

When you send data to an FTP server at IBM, you should use naming conventions that indicate the type of file or data set. You should also indicate the PMR number to identify the file or data set and where it is coming from. Table 2-2 shows some examples of naming conventions.

Table 2-2 PMR numbers and what they indicate

File/data set name	Comments
PMR12345.CEEDUMP	Belongs to PMR #12345 and contains a CEEDUMP
PMR12345.CEEDUMP.TERSED	Belongs to PMR #12345, contains a CEEDUMP, and is compressed using TRSMAIN
PMR12345.CONFIGFILES.ZIP	Belongs to PMR #12345, contains configuration files, and is compressed on the PC using a ZIP program
PMR12345.CONFIGFILES.EXE	Belongs to PMR #12345, contains configuration files, is compressed on the PC using a ZIP program, and is then converted to an EXE file
PMR12345.CONFIGFILES.TAR	Belongs to PMR #12345, contains configuration files, and is compressed using the USS <code>tar</code> command

These conventions might be slightly different in various IBM geographies and regions. IBM support personnel can advise you about how and with which naming conventions data should be sent to the FTP server.

2.6 IBM contacts

There are various possibilities for contacting IBM support personnel.

Usually the person working on the problem that you reported provides you with an e-mail address and phone number. This information exchange allows you to discuss the problem or to give additional information directly.

It is very helpful for IBM support to know who your backup is (with phone number and e-mail address) in case IBM support needs additional information and you are not available. Also, you can give your contact this information or you can update the PMR by yourself to provide additional information.

You can use e-mail (for small files or configuration data), FTP, or other tools to provide IBM support with problem documentation. The most common and preferred way is FTP, as described in 2.5, “Exchanging data with IBM by FTP” on page 21.

Many support teams have their own team-related e-mail address that allows all team members access to your documentation.

Refer to the IBM Directory of worldwide contacts if you are contacting IBM support for the first time:

<http://www.ibm.com/planetwide>

Archived



Information sources

In addition to this IBM Redbook, other documentation, such as books and Web sites, is available for WebSphere for z/OS and supporting components.

This chapter describes some of the resources available that the authors have found very helpful for solving problems in the WebSphere for z/OS environment.

3.1 WebSphere for z/OS support pages

Traditionally, the most commonly referenced sources of information about WebSphere for z/OS are the product manuals. These product manuals are important, but your main source for WebSphere Application Server for z/OS should be the home page. From there you can access other pages with important information about WebSphere on z/OS, which we describe in this section.

3.1.1 The WebSphere for z/OS home page

The WebSphere for z/OS home page (Figure 3-1) is a central entry point for a wide variety of information about WebSphere for z/OS. You can find the page at:

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

The screenshot shows the IBM WebSphere Application Server for z/OS home page. The top navigation bar includes the IBM logo, a search bar, and links for Country/region, Terms of use, Home, Products, Services & solutions, Support & downloads, and My account. A left sidebar contains a 'WebSphere Application Server for z/OS' section with links to Features and benefits, System requirements, Library, Success stories, News, Trials and betas, How to buy, Events, Training and certification, Services, and Support. Below this are sections for Related software, Related solutions, Related hardware, and Related services. The main content area is titled 'WebSphere Application Server for z/OS' and includes an 'Overview' section with a bulleted list of features and benefits. On the right, there is a 'We're here to help' section with contact options (Call me, E-mail us) and a phone number. Below that is a 'WebSphere Application Server V5.0.2' banner with a 'Trial Available' button. At the bottom right, a 'Highlights' section mentions '40 years of mainframe innovation' and provides links to 'WebSphere for z/OS V6.0.1 Webcast' and 'WebSphere for z/OS ISV Enablement'.

Figure 3-1 IBM WebSphere Application Server for z/OS home page

Click the links in the gray navigation bar on the left side of the page for specific information categories such as system requirements, the library (manuals and Information Center), and services.

3.1.2 WebSphere support page

Click the support link on the WebSphere for z/OS home page to access the WebSphere for z/OS product support Web site (Figure 3-2) at:

http://www.ibm.com/software/webervers/appserv/zos_os390/support/

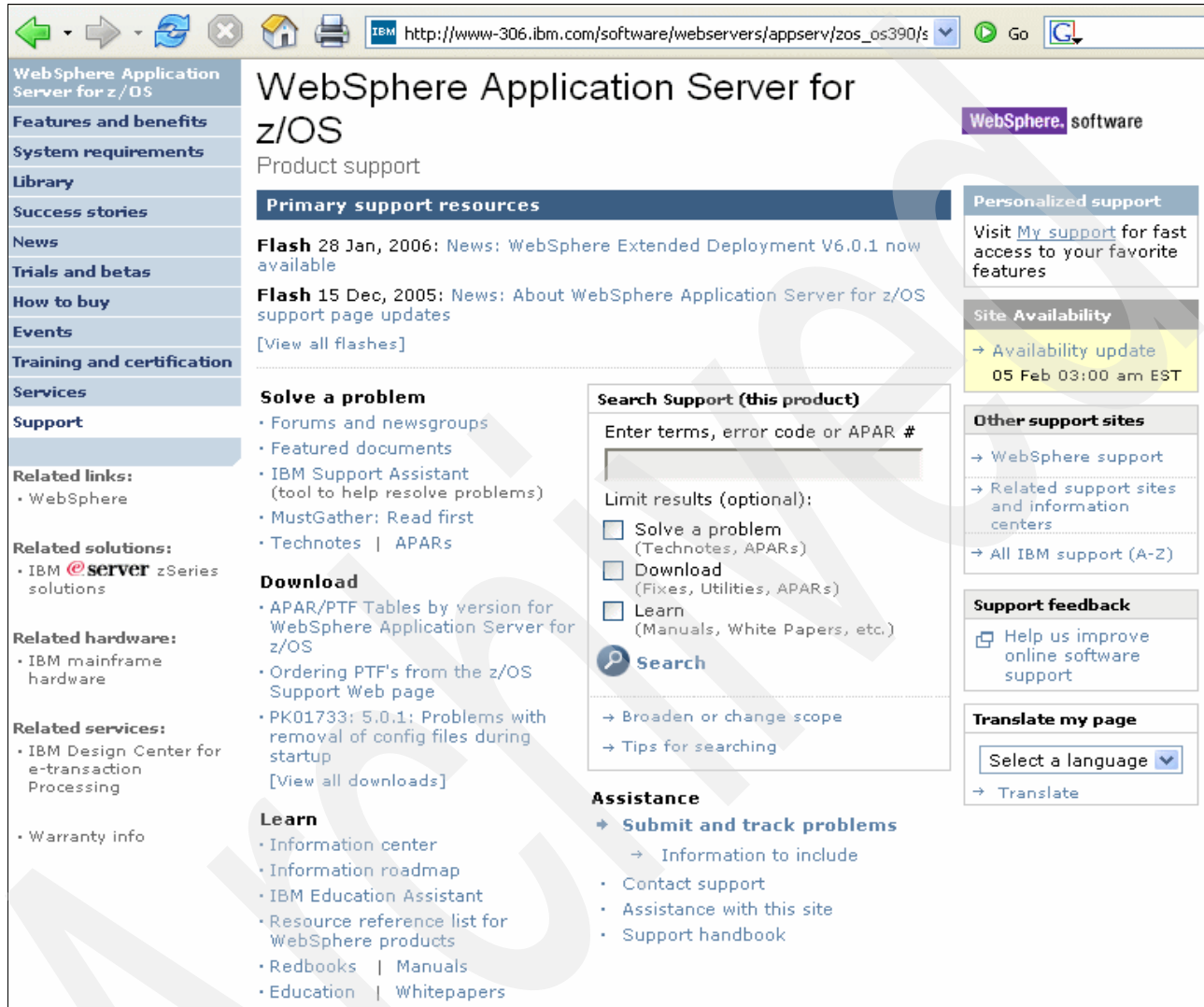


Figure 3-2 IBM WebSphere for z/OS support Web site

3.1.3 WebSphere for z/OS V6 product manuals

Click the Library link in the left navigation bar of the WebSphere for z/OS Web site to access the WebSphere Application Server library Web site:

<http://www.ibm.com/software/webervers/appserv/was/library/>

As shown in Figure 3-3 on page 28, this site is not z/OS-specific, which means that you can use it as reference for WebSphere Application Server material for all platforms.

WebSphere Application Server		WebSphere software
WebSphere Application Server	<div> V 6.0.x V 5.1.x V 5.0.x V 4.0.x V 3.5.x </div>	
Features and benefits		
System requirements		
Library		
Success stories		
News		
Trials and betas		
How to buy		
Events		
Training and certification		
Services		
Support		
Support for z/OS		
Related software	<div> WebSphere Application Server - Express WebSphere Application Server WebSphere Application Server Network Deployment WebSphere Application Server for OS/400 WebSphere Application Server for z/OS WebSphere Extended Deployment Rational Application Developer for WebSphere Software Rational Web Developer for WebSphere Software Macromedia ColdFusion MX for WebSphere </div>	
Related solutions	<div> WebSphere Application Server Version 6.0.x Information Center Express - distributed platforms Show Hide WebSphere Application Server - distributed platforms Show Hide Network Deployment - distributed platforms Show Hide WebSphere Application Server - z/OS Show Hide </div>	
	<div> Documentation - Information center format WebSphere Application Server for z/OS, Version 6.0.1 IBM WebSphere Help System plug-in: WebSphere Application Server for z/OS IBM WebSphere Help System plug-in: Javadoc Documentation - Book format Program directory (PDF) Migrating, coexisting, and interoperating (PDF) Installing your application serving environment (PDF) Administering applications and their environment (PDF) Setting up the application serving environment (PDF) Using the administrative clients (PDF) Securing applications and their environment (PDF) Developing and deploying applications (PDF) Troubleshooting and support (PDF) Tuning guide (PDF) License information Software license agreements Release notes Release notes Supported hardware and software Supported hardware and software Supported hardware </div>	
	<div> Related resources Download IBM WebSphere Help System [instructions] Get Adobe® Reader® Subscribe to My Support e-mail to learn of documentation updates </div>	
	<div> Related libraries WebSphere Application Server for OS/400 library WebSphere Application Server - Express library IBM HTTP Server library WebSphere Application Server - Edge Components library WebSphere Extended Deployment library WebSphere Business Integration Server Foundation library WebSphere Business Integration Server </div>	

Figure 3-3 IBM WebSphere Application Server library page

This page lists the WebSphere product manuals. Select **Show** from the WebSphere Application Server - z/OS section to link to the Information Center. This action also displays a list of the product manuals for z/OS that are available in PDF format. At the time of publication, the following guides were available:

- ▶ *Program Directory*, GI11-2825
- ▶ *Migrating, Coexisting, and Interoperating*, SA23-2207
- ▶ *Installing Your Application Serving Environment*, GA22-7957
- ▶ *Administering Applications and Their Environment*, GA22-7962
- ▶ *Setting Up the Application Serving Environment*, GA22-7958
- ▶ *Using the Administrative Clients*, SA23-2208
- ▶ *Securing Applications and Their Environment*, SA22-7961
- ▶ *Developing and Deploying Applications*, SA22-7959
- ▶ *Troubleshooting and Support*, GA22-7964
- ▶ *Tuning Guide*, SA22-7963

Attention: There is no longer a *Messages and Codes* manual. For messages and codes, see the Information Center or Appendix A, "Messages and codes" on page 311.

3.1.4 WebSphere for z/OS V6 Information Center

The Information Center lists documentation for several WebSphere Application Server products. To access this list, go to:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

To download your own copy of the Information Center go to:

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

Check the Techdoc *How can I put a local copy of the WebSphere Information Center on my workstation*, FQ102912.

In the Information Center navigation menu, select **WebSphere Application Server for z/OS V6** to display the welcome page, which contains information that is specific to this product.

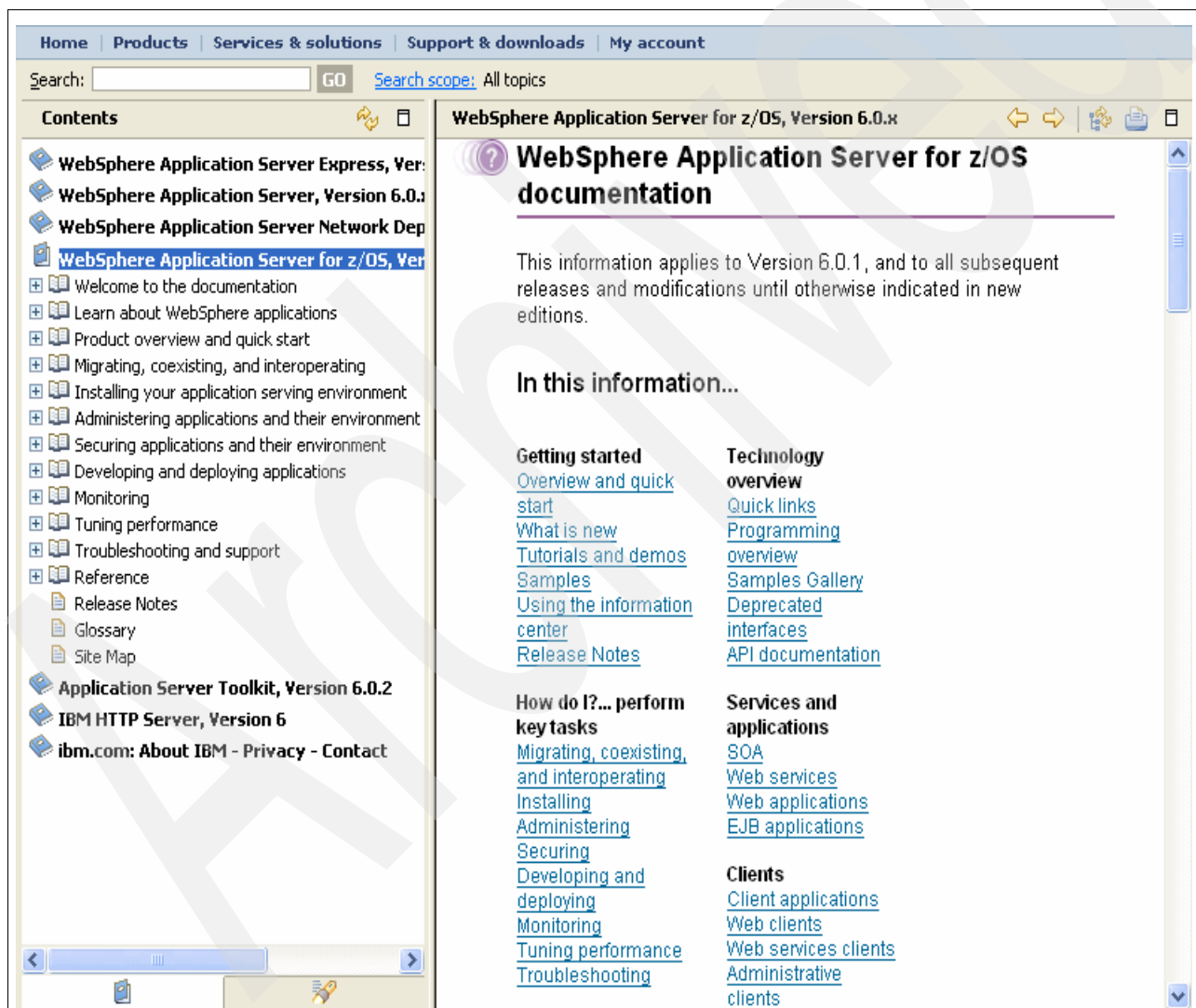


Figure 3-4 IBM WebSphere Application Server for z/OS Information Center page

To limit the search scope:

1. Click **Search scope** at the top of the page.
2. In the window that opens, click **New**.
3. Define a list name.
4. Select **WebSphere for z/OS**.

To look up messages and codes, either search for the particular message or code in the Information Center or go to the Contents panel and select **Reference** → **Troubleshooter** → **Messages** (Figure 3-5). Then, select a tab according to the first few letters in your message code. See Appendix A, “Messages and codes” on page 311 for WebSphere for z/OS messages and their code explanations.

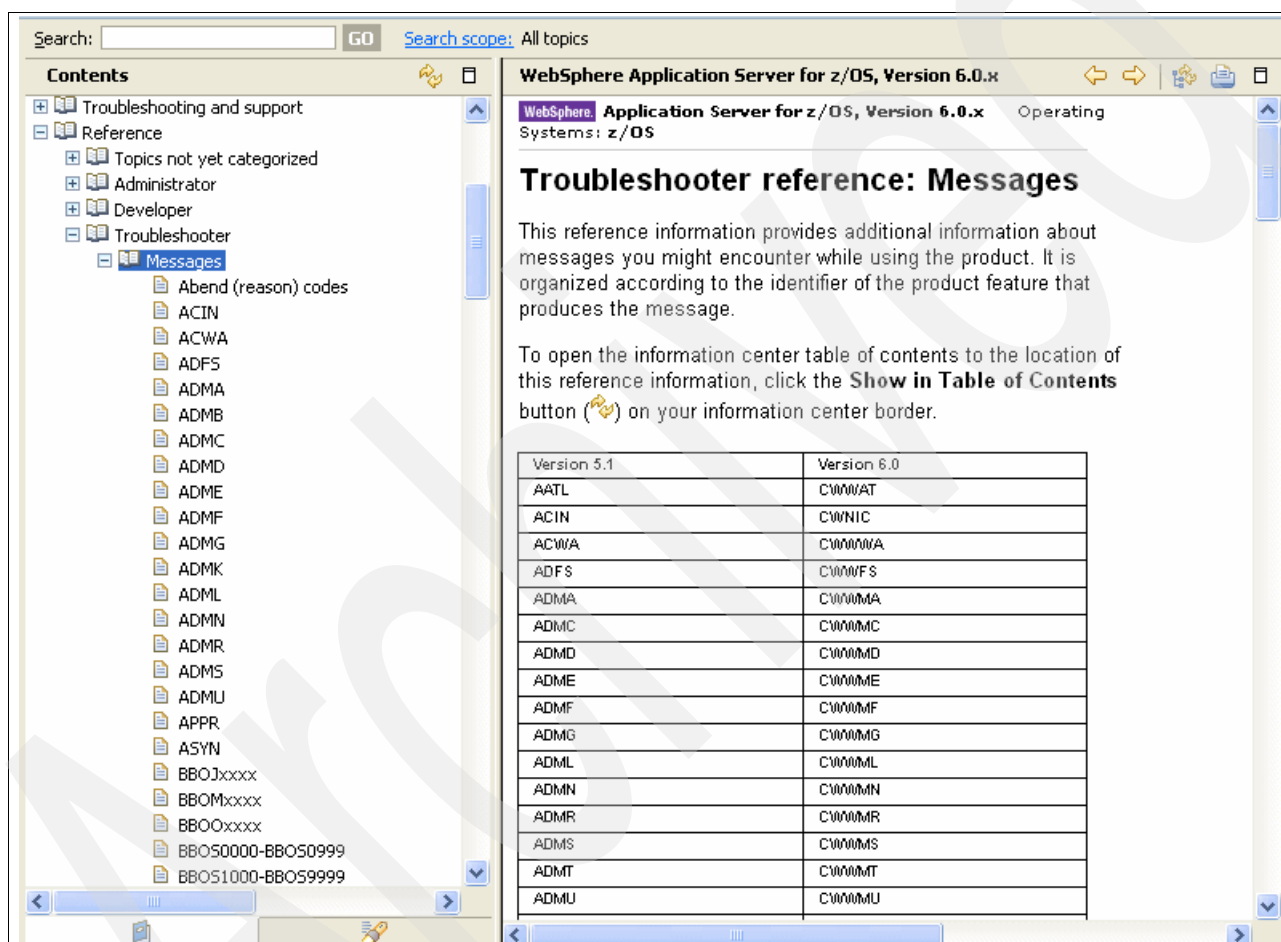


Figure 3-5 Messages and codes in the Information Center

3.1.5 WebSphere for z/OS IBM services

Consider using WebSphere Application Server for z/OS IBM services. For example, during an Architecture and Design Workshop, using preconfigured IBM hardware and software, IBM consultants can design and implement a working solution to your business problem, or a “WebSphere Proof of Concept for z/OS.” This allows you to implement a WebSphere production environment without interfering with your day-to-day business functions. For more information, see:

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

3.1.6 Recommended reading list: WebSphere Application Server

IBM Software Services for Software has compiled a reading list from a variety of sources for customers, consultants, and other technical specialists. Many of these documents focus on critical areas that should be understood before you start any Web application design and implementation. Others illustrate different stages of the project life cycle and should be reviewed before proceeding with each progressive phase.

http://www-128.ibm.com/developerworks/websphere/library/techarticles/0305_issw/recommendedreading.html

3.2 Techdocs: White papers, hints, and tips

At the Techdocs site, you can find guides for configuration, installation tips, operational recommendations, and tuning and debugging documents. There are also links to:

- ▶ PTFs and APARs
- ▶ Flashes
- ▶ Presentations and Tools
- ▶ FAQs
- ▶ White papers

To access this information, go to:

<http://www.ibm.com/support/techdocs/>

3.3 Redbooks and draft publications

The number of IBM Redbooks that address WebSphere for z/OS is growing.

The list in Figure 3-6 on page 32 is a small sample of the WebSphere IBM Redbooks and Redpapers that are available.

To preview books and papers that are scheduled for publications, click **Drafts**.

Default is "Exact Phrase." Use AND, OR, NOT to separate keywords.

New search:

Sort by: ☐ Fuzzy search

[search tips](#)

The following Redbook Domains relate to your search request:

→ [WebSphere Redbooks Domain](#)

250 results found in Redbooks, Redpapers, Drafts and Technotes

To learn more about Redbooks and Redpapers, [click here](#).

- 56. [Using WebSphere Extended Deployment V6.0 to build an On Demand production environment](#), SG24-7153-00
Draft Redbook, last update 12 January 2006
- 59. [Highly Available Architectures and Capacity Planning with WebSphere Remote Server V5.1.2.1](#), SG24-7184-00
Draft Redbook, last update 12 January 2006
- 80. [WebSphere for z/OS V6 Connectivity Handbook](#), SG24-7064-02
Redbook, published 28 December 2005
- 84. [System Integrators Guide: Integrating WebSphere Everyplace Mobile Portal with WebSphere Voice Products](#), REDP-4116-00
Redpaper, published 27 December 2005
- 86. [Publishing IMS and DB2 Data using WebSphere Information Integrator: Configuration and Monitoring Guide](#), SG24-7132-00
Redbook, published 27 December 2005
- 87. [IBM WebSphere Voice Server for Multiplatforms V5.1.1/V5.1.2 and WebSphere Voice Response: An Interoperability Guide](#), REC
Draft Redpaper, last update 27 December 2005
- 88. [Build a business process solution using Rational and WebSphere tools](#), SG24-6636-00
Draft Redbook, last update 22 December 2005
- 105. [WebSphere Information Integrator Content Edition: Planning, Configuration, and Monitoring Guide](#), SG24-6784-00
Redbook, published 8 December 2005

Figure 3-6 Recent IBM Redbooks and Redpapers

3.4 Sources of information for developers

There are various sources of information for developers that also help administration and support staff when trying to solve WebSphere for z/OS problems.

3.4.1 WebSphere Developer's Domain

At the WebSphere Developer Works Web site, you can access platform-independent information, best practices, hints and tips, documentation, tools, and other technical information:

<http://www.ibm.com/developerworks/websphere/>

At the site, there are links to:

- ▶ New to WebSphere
- ▶ Products
- ▶ How to buy
- ▶ Downloads
- ▶ Technical library
- ▶ Training
- ▶ Support
- ▶ Services
- ▶ Forums and community
- ▶ News

Important: These documents are not z/OS-specific and some of the tools might not be available for all platforms.

3.4.2 The alphaWorks community

Through alphaWorks®, developers around the world have the unique opportunity to experience the latest innovations from IBM. Emerging technologies are available for download at the earliest stages of development before they are licensed or integrated into products, so that users can evaluate and influence IBM research and development. In addition, early adopters have access to a virtual collaborative community to learn more about the uses of a particular technology, and opportunities for commercial use of alphaWorks' technologies.

<http://www.alphaworks.ibm.com/>

3.4.3 Java

At the Java community process Web site, you can access many hints and tips for coding J2EE applications:

<http://jcp.org/>

The Web site includes a reference section with specifications, white papers, and other Java-related information.

At the SUN Java technology home page, you can access first-hand information directly from the founder of Java technology:

<http://java.sun.com>

Sun Java™ Technology still is a major contributor to the Java community. The Web site contains technical information, specifications, examples, and references for J2EE.

There is also a Java documentation Web site:

<http://java.sun.com/j2se/download.html>

For information and downloads about J2EE, see

- ▶ J2EE Software Development Kits (SDK)
- ▶ J2EE Application Programming Interface (API) documentation
- ▶ J2EE platform specification

See the IBM developerWorks and alphaWorks domains for more Java-related information.

3.5 Other helpful Web sites

When you encounter problems with WebSphere for z/OS, consider that the application server is implemented in z/OS, a complex and function-rich operating system on a platform that provides a lot of special features. Problems that appear in an application or in the application server might actually stem from the underlying operating system or platform, or from not exploiting both of them efficiently. The Web sites in this section can help you find the problems related to the underlying technology of WebSphere on z/OS.

3.5.1 zSeries support

Figure 3-7 shows the zSeries support-related links found at the IBM System z™ support Web site.

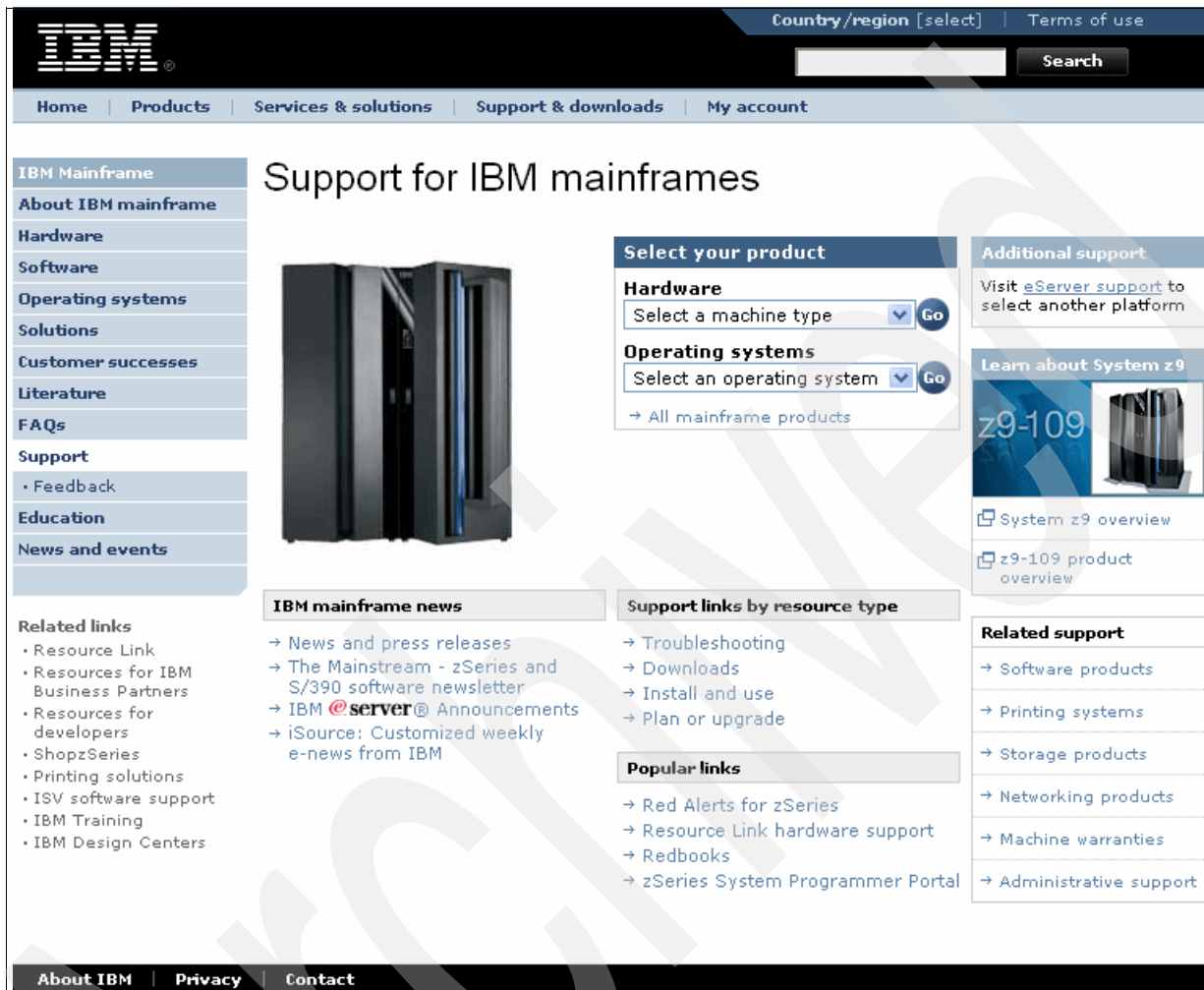


Figure 3-7 zSeries support page

To access these links, go to:

<http://www.ibm.com/servers/eserver/support/zseries>

3.5.2 z/OS home page

The z/OS home page covers the complete range of software products and technologies for this operating system:

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

3.5.3 LookAt messages

The LookAt site (see Figure 3-8 on page 35) provides a great search mechanism for z/OS messages:

<http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/>

z/OS

About z/OS

Software

How to Buy

Migration & Installation

News

Support

Downloads

Education

Library

Ease of Use

Contact z/OS

Related links

- Resources for business partners
- Resources for developers

LookAt

List Books

Download

Feedback

Help

1. Enter a Message ID

2. Select an area to LookAt!

Note: Platform selections below also include any IBM application software messages enabled for LookAt.

z/OS® and z/OS.e™

☐ V1R7
☒ V1R6
☐ V1R5
☐ V1R4
☐ V1R3

z/OS® only

☐ V1R2

Clusters for

☐ AIX™ and Linux™

z/VM®

☐ V5R1
☐ V4R4
☐ V4R3

VSE/ESA™

☐ V3R1
☐ V2R7
☐ V2R6
☐ V2R5

3. Click

Go

[What's New](#)
[LookAt Mobile Edition](#)

Or, to view messages in IBM books that have not been enabled for LookAt, select an area to search using BookManager®:

APAR and ++HOLD Docs

z/OS®

☐ All releases

For more information about LookAt...

- Do you want to know which books are enabled for LookAt? To display a list of the books that are currently enabled for LookAt, click [List Books](#).
- Host LookAt users: do you need to download LookAt code for TSO/E and Windows? To download LookAt code that will run on either platform, click [Download](#).

Figure 3-8 LookAt messages

Message tables are also provided in Appendix A, “Messages and codes” on page 311.

3.5.4 All software products

On this page you can search for help and information about all IBM software products:

<http://www.ibm.com/software/sw-atoz/>

3.5.5 IBM Software support guide

The purpose of this document is to provide guidelines and reference materials that customers might need when they require IBM service and support:

<http://techsupport.services.ibm.com/guides/handbook.html>

This site has links to:

- Overview
- Enhanced support
- Contacting IBM
- No support contract
- Preventing problems
- Support contacts
- Additional offerings
- Acronyms and other terms

3.5.6 z/OS Internet library

From this Web site, you can view, print, and order all available IBM manuals:

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

You can view the books in HTML format or you can download them in PDF format for easy reference and printing.

For System z and zSeries soft copy information, see:

<http://www.ibm.com/servers/eserver/zseries/softcopy/>

3.6 Educational information

IBM offers a comprehensive portfolio of technical training and education services that are designed for individuals, companies, and public organizations that wish to acquire, maintain, and optimize IT skills.

3.6.1 IBM Global Services

Go to the IBM Global Service Web site:

<http://www-1.ibm.com/services/us/index.wss/home>

Click **Training** to browse the:

- ▶ Course catalog
- ▶ e-Learning
- ▶ Blended learning
- ▶ Save money
- ▶ On-site training

3.6.2 WebSphere for z/OS training and certification

Click **Training and Certification** in the left navigation bar on the WebSphere for z/OS Web site or go to:

<http://www-306.ibm.com/software/info1/websphere/index.jsp?tab=education/index>

This site offers the latest information about upcoming training events, e-learning courses, and online tutorials that IBM offers for WebSphere and Java, for example:

WBSR6	WebSphere for z/OS V6 Implementation Workshop
WSW06	Security Workshop: WebSphere Application Server for z/OS

3.6.3 IBM Education Assistant

This site integrates narrated presentations, Show Me demonstrations, tutorials, and resource links to help you successfully use IBM software products:

<http://www-306.ibm.com/software/info/education/assistant/>

You can go directly to the education modules and select **WebSphere Application Server for z/OS** to obtain a list of education materials. For example, after selecting **Problem Determination** (see Figure 3-9 on page 37), you can download related presentations and audio material and work with them at your own pace.

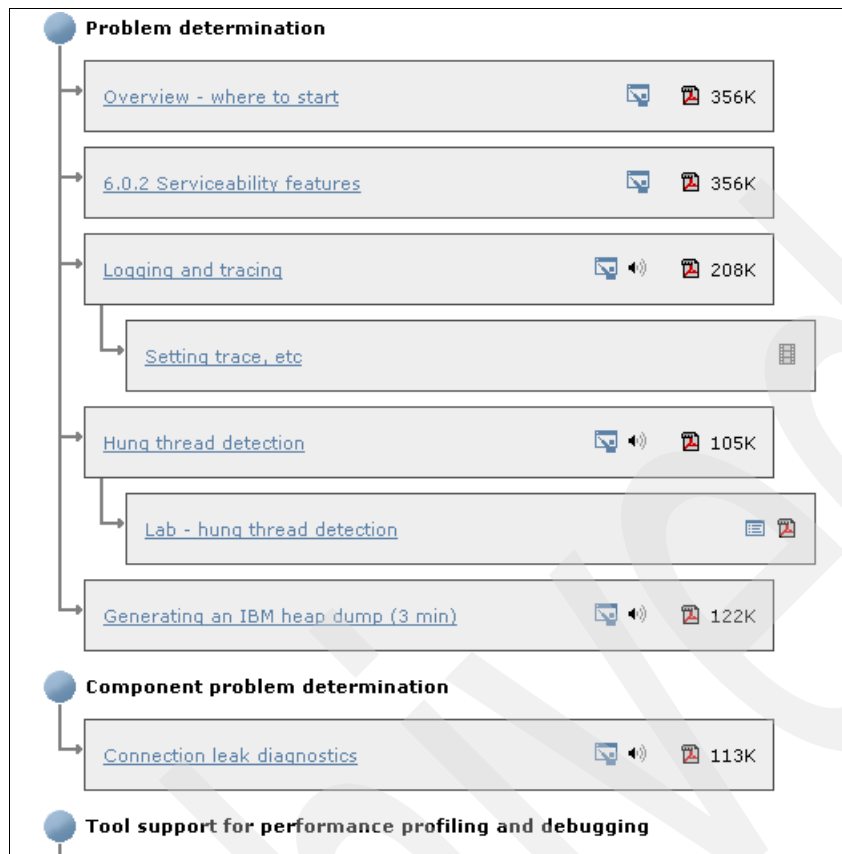


Figure 3-9 IBM Education Assistant WebSphere for z/OS: Problem determination

Archived

Problem symptoms and their resolutions

This part of the book is intended for system programmers and administrators who are working with WebSphere Application Server for z/OS V6 and who must identify problems, analyze them, and fix them efficiently so that they can deliver good support for the WebSphere environment. Another objective of this part is to assist IBM support and technical professionals and Java developers who work in this environment.

If a WebSphere for z/OS problem occurs, the symptom chapters can help you to determine what happened and why it happened. We show you how to diagnose various problem symptoms that are encountered in a WebSphere for z/OS environment.

Flow charts guide you through each step of the analysis process. Based on tasks that you must perform and answers that you give to a series of questions that were designed to filter out irrelevant facts, we try to help you:

- ▶ Determine the problem area.
- ▶ Identify the type, source, and cause of the problem.
- ▶ Discover a solution.

We also mention analysis tools and information sources that are related to the symptoms. When no feasible solution is apparent, you are directed to prepare documentation and contact IBM.

Archived

Exceptions and error messages

The goal of any deployed system or application is to reach and maintain steady state. The most critical task in support of that goal is to monitor your application for symptoms of potential problems that can compromise that goal and address them as early as possible. This is usually the most taxing task in the information technology life cycle. This chapter shows you how to diagnose the symptom of exceptions and errors.

The flow chart in this chapter can help you find possible fixes to the exceptions or errors that you encounter in the daily operations of your WebSphere system. It does so based on answers you give to a series of questions that are designed to filter out irrelevant facts and take you directly to the problem area. If solutions exist to your errors or exceptions, they are suggested. We also mention the analysis tools and information sources that are related to this symptom. When no feasible solution is apparent, you are directed to prepare documentation and contact IBM.

4.1 What is an exception or error?

An exception or error is an indication of a condition that is outside the normal operation of a WebSphere process. We use *exception* and *error* interchangeably in this section. Runtime errors are exceptions in J2EE parlance. Both reflect irregularities from the required functionality of end-to-end solutions. The job is to narrow the problem down to a specific root cause from the exception or error symptom given. There are cases where many root causes contribute to one symptom and vice versa.

There are two types of WebSphere errors: native code errors and Java exceptions.

- ▶ *Native code errors* happen when the base code layer that supports WebSphere components encounter irrecoverable conditions.

Native code errors have message IDs that are accompanied by descriptions. By design standards, the first four letters of a z/OS WebSphere message ID identifies the module or submodule belonging to a component.

- ▶ *Java exceptions* happen when errors arise in either the code that makes up WebSphere components or in the application that runs in the WebSphere application server.

Java exceptions follow the package, class, and method name convention, meaning library, component, and module.

Tip: Inspect your system logs for potential problems or early warnings as a daily routine. This habit makes you more familiar with your WebSphere environment because you can see small changes in behavior and warning messages early in the process, which might save you from having to fix big problems later on.

4.2 Symptom flow chart: Exceptions and error messages

Figure 4-1 shows the flow chart for an exception or error message.

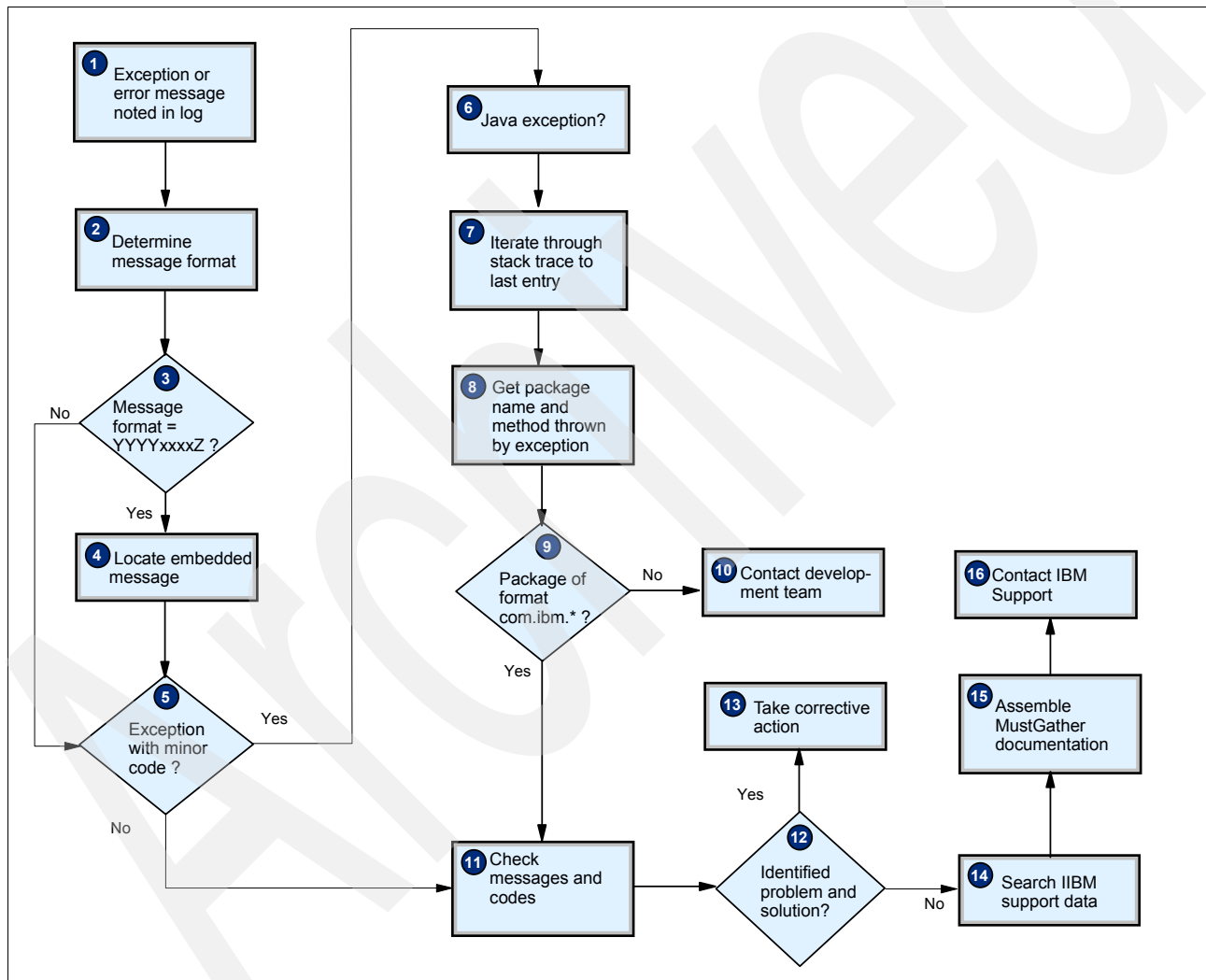


Figure 4-1 Flow chart for symptom: Exception and error

As you can see, each box has a number referring to the analysis in more detail.

4.3 Diagnosing an error or exception message

Follow these steps to analyze and solve error or exception symptoms:

1. Exception or error noted in syslog.

This can be a generic symptom that appears in any of the logs, traces, or the user interface. Pick up the major tokens (keyword, context, or eye catcher), which can be *error*, *exception*, *failed*, *not available*, or *not found*, and locate a number associated with this word. For example, suppose you notice this message in the syslog because it has the word *failed*:

```
16.18.08 STC06476 BB000038E Function CTRACE-DEFINE failed with RC=12,  
REASON=00001901, EXTENDED REASON=00000000
```

Several numbers are associated with this message; the ones that you should note are BB000038E, STC06476, RC=12, REASON=00001901, with BB000038E as the first priority.

2. Determine message format.

Analyze the message format to classify the message.

A Java exception always uses the keyword *exception* but does not have a message number. Other native code problems show very specific behavior that allows you to classify their origin.

A message from an IBM component (WebSphere for z/OS or any z/OS system components) usually has a format of YYYYxxxxZ, where YYYY stands for a letter combination that indicates the module name of the failing component, xxxx denotes the message number, and Z indicates the severity of the problem.

3. Is the message format YYYYxxxxZ?

Did you find a message number with the format YYYYxxxxZ associated with the error message or the problem? If yes, then locate the embedded message; see step 4. If no, then look for the exception minor code as described in step 5.

4. Locate embedded message.

Sometimes you can have a second number in a YYYYxxxxZ message with the same format. This indicates that the component is native code that acts as a wrapper for other code.

For example:

```
BB000220E SECJ0222E: An unexpected exception occurred when trying to ...
```

The BB000220E message has an embedded message number: SECJ0222E. When you look up the message number, use the second message ID because this is more specific and more likely to help when determining the root cause.

5. Do you see exception with minor code?

Does the message include the term *exception* and have a minor code associated with it (often in the format of C9C2xxxx)? If yes, then it is a Java exception. See step 6. If no, then look up the message at the WebSphere Information Center as described in step 11.

6. Is it a Java exception?

You can tell Java exceptions from other languages by the token names such as *java*, *servlet*, and the term *exception* that is usually part of the class or method name that handles exceptions for a particular library. Java exceptions can be from either the WebSphere product code or the application code that is running on the application server.

If the exception has a function name with a minor code, usually in the C9C2xxxx format, then it is a native code component that is wrapped by a Java class. Often, a Java wrapper class throws an exception on behalf of a native component and that is why you follow the stack trace to the original point of error.

7. Iterate through the stack trace.

When you read the trace (in the log), you see a chain of method calls, from class to class, leading to the one that takes the exception. Stack traces are in life order, so the last entry is always on top of the stack. Go to the last entry in the stack trace.

Figure 4-2 is a sample of a stack trace that was received in the log. The token that you should notice is A, **Exception**. This keyword indicates that it is a Java exception.

8. Get package name and method.

The last entry in the stack trace always has the last method that was in memory when the error occurred. The method name, its class, and its package name tell you the owner or provider of the package and are usually descriptive enough to hint at the root cause of the problem.

Item A in Figure 4-2 is the object type (kind) of the exception caught. It says what kind of work was being performed when the error occurred. Our example shows a Structured Query Language (SQL) call to the database that took the exception.

```
SQL Exception: Schema 'TRADER' does not exist at
db2j.ai.j.generateCsSQLException(Unknown Source) at
db2j.ai.g.wrapInSQLException(Unknown Source) at
itso.j2ee.trader.servlet.TraderSuperServlet.handlePerformLogon(TraderSuperServlet.java:
427) at itso.j2ee.trader.servlet.TraderSuperServlet.performTask(TraderSuperServlet.java
:303) at itso.j2ee.trader.servlet.TraderSuperServlet.doPost(TraderSuperServlet.java:78) at
javax.servlet.http.HttpServlet.service(HttpServlet.java(Compiled Code)) at
com.ibm.ws.webcontainer.servlet.ServletWrapper.handleRequest(ServletWrapper.jav
a(Compiled Code)) at
com.ibm.ws.webcontainer.servlet.CacheServletWrapper.handleRequest(CacheServletWra
pper.java(Compiled Code)) at
com.ibm.ws.webcontainer.WebContainer.handleRequest(WebContainer.java(Compiled
Code)) at
com.ibm.ws.webcontainer.channel.WCChannelLink.ready(WCChannelLink.java(Compiled
Code)) at
com.ibm.ws.http.channel.inbound.impl.HttpInboundLink.handleDiscrimination(HttpI
nboundLink.java(Compiled Code)) at
```

Figure 4-2 A Java stack trace with an exception

Item B tells you more about the class library and package where that code came from; it is application code in this example. Item C shows what a WebSphere component Java package looks like. It starts with “com.ibm.ws,” which represents the WebSphere code libraries. It does not cause the exception in our example but shows the difference in structure to the application code packages.

9. Does the package have a format of com.ibm.*?

The package name of a system application (WebSphere product code) starts with com.ibm.ws.* (item C in Figure 4-2). A message with the format of com.ibm.* indicates an IBM product package. All other package names come from third-party applications that are running in WebSphere for z/OS.

Does your exception method show a package format of `com.ibm.*`? If no, then contact your application developer; see step 10. If yes, then it is a WebSphere system exception; go to step 11.

10. Contact development team.

If the package name is not from IBM, then it is from a third-party or your in-house application. In that case, you can contact your development team for support.

The roles in the WebSphere environment overlap. By development, we mean your developers, system administrators, and systems programmers. Provide them with the trace (the method name, the package, and the class that threw the exception) so that they can determine the owner of the application that is causing the problem and what application component needs to be fixed.

11. Check messages and codes.

With either the message number, the embedded message number, the minor code, or the exception method and class, search for more information at the WebSphere Information Center:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

If your (embedded) message number starts with BBOO (component ID), then it is a WebSphere for z/OS product component message. See A.1, “WebSphere for z/OS message codes” on page 312. Embedded messages in a BBOO0222I message are Java component messages. For identification of the specific component, see A.1.1, “Specific Java component messages” on page 312.

If you have a minor code error, such as one that starts with C9C2, you are dealing with WebSphere for z/OS product code. See A.1.2, “Minor codes” on page 314. In the case of a WebSphere component (`com.ibm.ws.*`) exception, you can copy the class, method names, and verbiage from the exception and use them as your search token at the Information Center.

The Information Center has many more messages and codes and the explanations of them, along with hints and tips for solving specific problems. Use it as your first point of reference when you encounter a problem. For example, assume that you received this message in your syslog:

```
16.18.08 STC06476 BB000038E Function CTRACE-DEFINE failed with RC=12,
REASON=00001901, EXTENDED REASON=00000000
```

The BBOO prefix indicates a WebSphere for z/OS component failure. Search for BBOO0038E and the specific return and reason codes in the Information Center at:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v5r0/index.jsp?topic=/com.ibm.websphere.zseries.doc/info/zseries/ae/bboo02.htm>

The result is a description of the error (Example 4-1).

Example 4-1 Description of BB00038E search

BB000038E Function string failed with RC=dstring, REASON=hstring, EXTENDED REASON=hstring.

Explanation: WebSphere for z/OS failed as indicated and that function completed with a decimal return code indicated by RC, a hexadecimal reason code indicated by REASON, and an extended hexadecimal reason code indicated by EXTENDED REASON.

User Response: Consult the function indicated in the OS/390 C/C++ Library Reference, OS/390 MVS Programming: Assembler Services Reference, OS/390 MVS Programming: Authorized Assembler Services Reference, or other appropriate z/OS reference book for a description of this error.

If you search the general IBM support Web site, you can find an entry (Figure 4-2) with a similar description.

Example 4-2 Result of searching the general IBM support Web site

How to manage operator message routing in WebSphere for z/OS Version 5
SYSC SERVER= none ./bbortbuf.cpp+513 ... BB000038E Function CTRACE-DEFINE failed
with RC BossLog ... SERVANT PROCESS THREAD COUNT IS 6. HRDCPYDD BB000038E Function
CTRACE-DEFINE failed with R BBOJ0011I

This technical document described the problem and the solution for the sample error. The authors were able to use the document to fix the problem. We routed hardcopy and default messages to specific data sets instead of sending them to the console (SYSLOG), which creates a lot of extraneous messages. We had to configure another data set (file) to hold the messages.

12. Have you identified the problem and solution?

After searching the WebSphere Information Center, did you identify the problem? Have you found information that matches your symptom data? Have you found a fix for your problem? If yes, then take the corrective action that is described in step 13.

If no, then you need to do more research at the IBM support Web sites as described in step 14, and if your research is not successful, prepare to contact IBM support. See step 15.

13. Take corrective action.

The information you find using the IBM support data might provide the following solutions:

- An existing APAR and PTF fix for your problem is available for you to apply.
- Other reports of your symptoms provide a procedure for fixing the problem.

In such cases, follow the instructions that are provided or apply the information to your specific problem to solve it.

14. Search IBM support data.

Search IBM support Web sites and databases, specifically the WebSphere for z/OS support site at:

http://www-306.ibm.com/software/webservers/appserv/zos_os390/support/

From this site, you can follow several links to other support sites related to WebSphere Application Server, its components, and z/OS.

When you search problem databases for information or fixes related to an error or exception, keep in mind that they are reported in many formats (sometimes with return codes or reason codes). Therefore, you might alter your search keyword to find a match.

For example, you might search for “EC3 abend” on the WebSphere for z/OS support site and receive a list with a number of documents associated to “EC3 abend”, including:

PK04379: SERVANT REGION ABENDS WITH EC3, REASON CODE 04060012 WITH SMF AND HIGH VOLUME ENVIRONMENT

Click that particular link to access this Web site:

http://www-1.ibm.com/support/docview.wss?rs=404&context=SS7K4U&dc=DB500&q1=EC3&uid=swg1PK04379&loc=en_US&cs=utf-8&lang=en

This document describes a problem, explains the reason, and recommends applying maintenance, while specifying the service level and APAR number to download. After you follow these recommendations, you restart the application server, and your server runs successfully without issuing another abend. Your problem is solved. Document the

problem and the fixes that you have applied in your system change documentation for your specific WebSphere for z/OS environment for later reference.

Any other messages with the YYYYxxxxZ format are usually documented and maintained in IBM documentation. See A.2, “System and component message table” on page 315 to identify other IBM products (such as z/OS components or subsystems) that might have created your particular error message.

Go to the specific product manuals as indicated in the Appendix, or search the IBM Software support Web site at:

<http://www-950.ibm.com/search/SupportSearchWeb/SupportSearch?pageCode=SPS>

See also Chapter 3, “Information sources” on page 25, which provides many valuable links and resources for solving problems in WebSphere for z/OS.

After you have exhausted all resources and can find no apparent fix for your problem, proceed with step 15 to prepare to contact IBM.

15. Assemble MustGather documentation

Prepare the problem documentation, referred to as *MustGather*, for IBM support. For more information about MustGather, see “MustGather” on page 16.

Read *MustGather: Read first for WebSphere Application Server for z/OS* for help with assembling the appropriate documentation. You can find this at:

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

The minimum information that is necessary is:

- Problem description
Include information that is related to when the problem first started to occur.
- Software version and maintenance (build) level
You find this information in the job log of your application server. Search for “build level,” and you obtain a line similar to this:

```
BBOM0007I CURRENT CB SERVICE LEVEL IS build level cf10515.05 release  
WAS601.ZNATV date 04/15/05 12:55:41.
```
- Operating system version and maintenance (PUT) level
- The job log of the abending address space, including both controller and servant region job logs
- Any dumps or traces that were triggered by the problem

See also 2.3, “Before you contact IBM support” on page 15. Then proceed with step 16 to contact IBM Support.

16. Contact IBM support

To contact IBM support, refer to Chapter 2, “Contacting IBM: Information” on page 13, for instructions.

Provide the information that is outlined in the MustGather documentation step.

Archived

Abend

This chapter explains what an abend is and provides a flow chart and step-by-step descriptions that can help you analyze the problem and find its cause. We also mention the analysis tools and reference information sources that are related to this symptom.

5.1 What is an abend?

An abend refers to abnormal termination. The term identifies the event that is issued when a system abnormally ends (abend) a task or address space that cannot continue processing and produce valid results. An abend is given a code to identify the type of error that occurred.

When a program runs into an unexpected and unhandled condition, it might call the Recovery Termination Manager (RTM) to issue an abend. In most cases, the operating system captures a dump for the task before it is automatically purged from the system. This dump can help you obtain details about the runtime state and the last active modules. Most abends are accompanied by symptoms, such as messages and codes, that offer hints about the problem. An abend code is perfect for searching for known problems.

Abends are mostly system related, and the codes are described in the product-related messages and codes manuals. See Appendix A, “Messages and codes” on page 311, for messages and codes related to a WebSphere for z/OS environment.

5.2 Symptom flow chart: Abend

Figure 5-1 shows the flow chart for this symptom. Each box has a number that refers to the analysis in more detail.

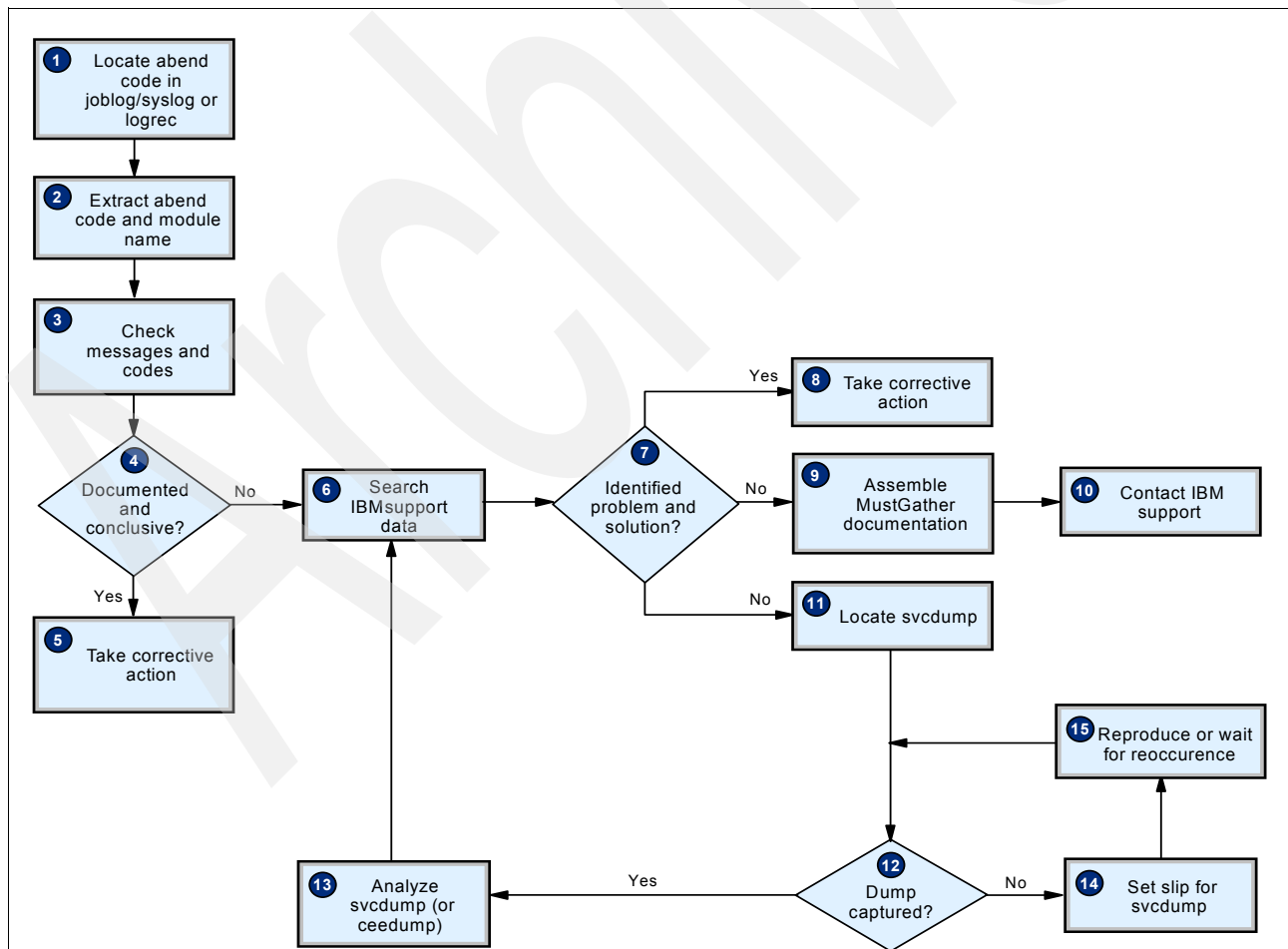


Figure 5-1 Flow chart for symptom: Abend

5.3 Diagnosing an abend

Follow these steps to analyze and solve an abend problem:

1. Locate abend code in job log, syslog, or logrec.

The first step is to locate the abend code. The abend message is found in the job log and syslog and is issued as part of z/OS message IEA995I as shown in Figure 5-2.

```
10.39.40 S0103633 IEA995I SYMPTOM DUMP OUTPUT 598
SYSTEM COMPLETION CODE=0D5 REASON CODE=00000021
TIME=10.39.40 SEQ=23671 CPU=0000 ASID=0077
PSW AT TIME OF ERROR 072C1000 AE86B6C4 ILC4 INTC 21
ACTIVE LOAD MODULE ADDRESS=2E863000 OFFSET=000086C4
NAME=BBOCMM
DATA AT PSW 2E86B6BE - D1B458F0 5098B218 F0005820
GR 0: 00000018 1: 7F1041B8
2: 30EBCDE0 3: 00FF8C90
4: 00000101 5: 7EEEE5F0
6: 30EBCDE0 7: 7EEEE5F0
8: 00000001 9: 0000000C
A: 00000000 B: AE86AF80
C: 2E86C0E8 D: 7F103F28
E: 0000030F F: 0001A80D
END OF SYMPTOM DUMP
```

Figure 5-2 Example of IEA995I message with abend code

The abend is also recorded in the sys1.logrec file and is extracted into a report using the Environmental Record, Editing, and Printing (EREP) program utility. The EREP report provides more information than is contained in the IEA995I message. The EREP report is particularly helpful when you have a series of abends. This is because it assigns a sequence number to each abend that makes it easier to identify what the first abend was. For more information about EREP, refer to the z/OS Internet Library at:

<http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/>

On the PDF line under **z/OS elements and features publications**, click your z/OS version. Under **Elements and features**, click **EREP** to find links to:

- *EREP V3R5 Reference*, GC35-0152
- *EREP V3R5 User's Guide*, GC35-0151

If WebSphere for z/OS issues a user completion code (abend), the way that the abend is recorded in the job log varies. If you are unable to find the abend message, you should try searching for other keywords such as “completion,” “code,” or “interrupt”. An example of an abend EC3 can be seen in Example 5-1.

Example 5-1 Example of EC3 abend found in job log

```
BPXP018I THREAD 24CAD40000000023, IN PROCESS 83886126, ENDED
WITHOUT BEING UNDUBBED WITH COMPLETION CODE 4FEC3000,
AND REASON CODE 04130007.
```

An abend or user completion code might have a Return code and a Reason code associated with it.

2. Extract abend code and module name.

After you have located the abend message, record the abend code and module name. The same abend code can be issued for many components, so the abend code alone is

usually not conclusive. From the module name or prefix, identify the component, subsystem, or product. In the case of Java code you might see a class path name.

When the module name that is recorded is not helpful or shows as unknown in the job log, you can find it in the EREP report. If you are unable to determine the module name, then verify it using an SVCDUMP. See Step 11.

3. Check message and codes manuals.

An abend code is either a z/OS system completion code or a user completion code:

- z/OS system completion code

This code is documented in the MVS System Codes manuals. The manual for this case is *z/OS V1R6.0 MVS System Codes*, SA22-7626-10. You can also consult *z/OS MVS Diagnosis: Procedures*, GA22-7587, a white paper with flow charts and step-by-step help that you can use to find a problem in the MVS operating system.

- User completion (abend) code

This code is documented in the specific manuals of the IBM component, subsystem, or product that issues the user completion codes. To determine the component, you can consult a table in A.2, “System and component message table” on page 315, which lists the message prefix and the issuing component.

The current abend codes specific to WebSphere Application Server for z/OS are CC3, DC3, and EC3 (Table 5-1). The full code descriptions are documented in A.1, “WebSphere for z/OS message codes” on page 312, and at the WebSphere Application Server for z/OS Information Center:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

Table 5-1 WebSphere related codes

Abend Code	Issuer
CC3	daemon processing failure
DC3	controller region processing failure
EC3	servant region processing failure

If a reason code is passed along with these abend codes, you can refer to the WebSphere Application Server for z/OS Information Center to obtain an explanation and determine the course of action. Table 5-2 shows an example of what you can find for abend reason 0001000.

Table 5-2 Abend reason code and explanation

Abend code	Abend reason	Explanation	Suggested action
CC3	0001000	BBORFRR routine was loaded into the wrong address. The routine should be in common.	The product was built or installed incorrectly. BBORFRR should reside in LPA and not be included in the STEPLIB/JOBLIB of the WebSphere for z/OS daemon address space.

Search for “abend (reason) codes.” If no explanation is given in the reason code and no indication is found in any information source, report the problem to IBM.

4. Is the information documented and conclusive?

Did you find the information in the messages manuals? Was the information adequate to identify the problem? Do you have enough information to correct the problem? If yes, then take corrective action; see step 5. If no, then search the IBM support pages; see step 6.

5. Take corrective action.

The messages and codes should provide the explanation of the abend code and a hint about how to respond. Processing this information should be sufficient to resolve the problem. You must restart the application server and check the logs for information about successful startup or other problems that you might encounter. In you have another abend, start to analyze it by going back to step 1.

6. Search IBM support data.

Search IBM support Web sites and databases, specifically the WebSphere for z/OS support site at:

http://www-306.ibm.com/software/webservers/appserv/zos_os390/support/

From this site, you can follow several links to other support sites that are related to WebSphere Application Server, its components, and z/OS. See also Chapter 3, “Information sources” on page 25, which provides many valuable links and resources for solving WebSphere for z/OS problems.

When you search problem databases for information or fixes related to an abend, keep in mind that abends, return codes, and reason codes are reported in many formats and you might have to alter your search keyword to find a match.

For example an EC3 abend might be reported as:

- Abend: ABENDEC3 or SEC3 or EC3 or 4FEC3000
- Return Code: RET4 or RC04 or RC4
- Reason Code: REASON4130004 or RSN04130004

7. Have you identified the problem and solution?

After searching the IBM support data, have you identified the problem? Have you found information that matches your symptom data? Have you found a fix for your problem? If yes, then take corrective action as described in step 8.

If no, prepare to contact IBM support or analyze the problem data further; see step 9.

8. Take corrective action.

The information that you have found using the IBM support data might have provided the following solutions:

- An existing APAR and PTF fix for your problem that is available for you to apply.
- Other reports of your symptoms that have provided a procedure for fixing the problem.

In those cases, follow the instructions that are provided or apply the information to your specific problem to solve it.

For example, you might have searched for an “EC3 abend” at the WebSphere for z/OS support site and received a list with a number of documents associated with EC3 abend, including:

PK04379: SERVANT REGION ABENDS WITH EC3, REASON CODE 04060012 WITH SMF AND HIGH VOLUME ENVIRONMENT

If you click that particular link, you gain access to this Web site:

http://www-1.ibm.com/support/docview.wss?rs=404&context=SS7K4U&dc=DB500&q1=EC3&uid=swg1PK04379&loc=en_US&cs=utf-8&lang=en

This document describes a problem, explains the reason, and recommends maintenance to apply, while specifying the service level and APAR number to download. After you follow these recommendations, restart the application server, and your server should run successfully without issuing another abend. Be sure to document the problem and the fixes that you have applied in your system change documentation for your specific WebSphere for z/OS environment for later reference.

9. Assemble MustGather documentation for abend.

MustGather documents can assist you with problem determination and save time resolving PMRs. For more information about MustGather, see “MustGather” on page 16.

For an abend, you should provide the following material:

- Problem description

Include information related to when the problem first started to occur.

- Software version and maintenance (build) level

Find this information in the job log of your application server. When you search for “build level,” you obtain a line similar to:

```
BBOM0007I CURRENT CB SERVICE LEVEL IS build level cf10515.05 release  
WAS601.ZNATV date 04/15/05 12:55:41.
```

- Operating system version and maintenance (PUT) level
- The job log of the abending address space (both controller and servant region job logs)
- The SVCDUMP triggered by the abend

10. Contact IBM support.

If you must contact IBM support, refer to Chapter 2, “Contacting IBM: Information” on page 13, for instructions about how to do this. Provide the information outlined in the MustGather documentation step.

11. Locate dump.

Usually the name of the SVCDUMP data set is recorded in the syslog. If you are not sure whether a dump was captured or which data set the dump was written to, then search for the word “dump” in the syslog and locate any messages pertaining to the dump.

Sometimes searching for “dumpid” can help find dumps when the word *dump* is too generic for a certain sysplex. Searching for dumpid results in messages such as this:

```
DUMPID=009 REQUESTED BY JOB (WT3DMGS )  
DUMP TITLE=COMPON=WEBSPPHRE Z/OS, COMPID=5655I3500,ISSUER=BBOR  
LEXT,ABEND IN CEEPLPKA/CEEOPCT
```

If there was a problem with capturing the dump, you see an IEAxxx message, such as:

```
IEA911I PARTIAL DUMP ON MVS.01MP.DMP00056 678  
IEA043I SVC DUMP REACHED MAXSPACE LIMIT - MAXSPACE=xxxx MEG
```

In such cases, you should fix the dump problem first before you attempt to analyze the dump because crucial information might not be written to the dump. Also ensure that your WebSphere for z/OS servers have the authority to create and write to the dump data sets.

12. Was a dump captured?

Was there a dump? Were you able to locate the dump? If so, then prepare to analyze the dump as described in step 13. If not, prepare to set a SLIP and contact IBM support. Go to step 14.

13. Analyze the SVCDUMP.

To analyze the SVCDUMP, invoke the z/OS MVS Interactive Problem Control System (IPCS). There are several methods for analyzing an abend using IPCS and data from the SVCDUMP.

The following steps are for only one of these methods; For more information about IPCS, see *z/OS V1R6.0 MVS IPCS Commands*, SA22-7594-05 and *z/OS V1R2.0 MVS IPCS User's Guide*, SA22-7596-01:

- a. Invoke IPCS and verify that you have the correct dump by checking the dump title, date, and time. To display this information, issue this command:

```
ip st validate worksheet
```

Figure 5-3 shows an example of output from this command.

I

```
MVS Diagnostic Worksheet

Dump Title: COMPON=WEBSPHERE Z/OS,  COMPID=5655I3500,ISSUER=BBORLEXT,
          ABEND IN BB0ORB  /UNKNOWN

CPU Model 2084 Version 00 Serial no. 012345 Address 01
Date: 02/18/2005      Time: 12:38:22.102475 Local

Original dump dataset: SYSPRD1.PLXA.SVCD.D050218.H123818.C2.N00011

Information at time of entry to SVCDDUMP:

HASID 04EC  PASID 04EC  SASID 04EC  PSW 070D1000 9C038948
```

Figure 5-3 Output from IPCS *ip st worksheet validate*

- b. Go to the diagnostic data report section and verify the abend code, reason code, and module name.
- c. Locate the Program Status Word (PSW) address of where the abend condition occurred and verify the module name in the summary format report, which can be obtained with:

```
ip summ format
```

- d. Scroll to the bottom of the report.
- e. Use **find previous** to locate the RTM2WA SUMMARY and control block data:

```
f 'rtm2wa summary' prev
```

The RTM2WA SUMMARY shows Recovery Termination Manager (RTM) data. This is the time-of-error information (see Figure 5-4 on page 56). Note the PSW address.

RTM2WA SUMMARY					

+001C	Completion code	840C4000			
+008C	Abending program name/SVRB address	007C2070	00000000		
+0094	Abending program addr	00000000			
GPRs at time of error					
0-3	00000000	215D3F00	00000000	21D126A0	
4-7	34326EB0	273C2250	7CA9E300	A667FA00	
8-11	A667F7AA	00000000	267908E8	221BDAA0	
12-15	33FE6C50	00000000	267918E0	0405A00C	
+007C	EC PSW at time of error	072C2000	A667FA42	00040004	00000000
+00DC	SDWACOMP	00000000			
+00E8	Return code from recovery routine-00				
	Continue with termination-implies percolation				
+00E0	Retry Address returned from recovery exit 00000000				
+00E4	RB Address for retry 00000000				
+000C	CVT Address	00FCB018			
+0038	RTCT Address	00FB24E0			
+00C8	SCB Address	007C4AC0			

Figure 5-4 IPCS ip summ format output showing RTM2WA SUMMARY

To determine the thread control block (TCB) address, you must scroll up a little to find the RTM2WA control block data and note the TCBC value.

In this example, the PSW is 072C2000 A667FA42. The second word is a 31-bit address.

For information about the format of the PSW, refer to *z/Architecture Principles of Operation*, SA22-7832-03.

- f. Locate the address in the dump storage. This is done from the IPCS main menu. In our example, we located 2667FA42 as shown in Figure 5-5.

ASID(X'04EC') ADDRESS(2667FA42.) STORAGE -----					
Command ==>					
2667FA42	A784	000A181B	9856E0D8	0D764700	xd...q.\Q...
2667FA50	FFA95800	D00018B0	A7F40005	18DB58B0	.z..}...x4.....
2667FA60	B00012BB	A774FF76	5820487C	BF1F201Cx.....@....
2667FA70	A784000D	A7AA0FF8	9856A0D8	0D764700	xd..x..8q..Q....

Figure 5-5 Browse dump storage using IPCS

- g. From the PSW address, try to determine the module name using the eye catchers in the dump (Figure 5-6).

2667E3B0	F2F0F0F5	F0F1F1F2	F2F0F3F1	F5F2F0F1	2005011220315201
2667E3C0	F0F2F0F0	0010E6F5	F1F0F2F0	F44D8282	0200..W610001(bb
2667E3D0	9696A299	815D0000	00C300C5	00C500F1	oosra)...C.E.E.1
2667E3E0	00005EF0	00000080	90684788	A74AFF80	..;0.....hx¢..
2667E3F0	0D805870	50485860	504C4100	00005810&..-&<.....

Figure 5-6 Search for eye catchers in dump storage near PSW address

The eye catchers are the ASCII characters that are found to the right of the storage. In this example, you can see the bboosra module name.

- h. Often obtaining a module name is sufficient, but when WebSphere for z/OS is involved, it is sometimes necessary to go further and find the related method name.

Examine the traceback data. Using the TCB from the RTM2WA information, enter the following command:

```
ip verbx ldata 'tcb(007c07f8) nthreads(*)'
```

When the output is displayed, locate the traceback information as shown in Figure 5-7.

Traceback:									
PU Addr	PU Offset	Entry	E Addr	E Offset	Statement	Load Mod	Service	Stat	
2667F7A0	+000002A2	SRAggregator::refresh(JNIEnv*,_jobject*,_jobject*)							
			2667F7A0	+000002A2		SUBP00L2		Call	
266127B0	+00000072	Java_com_ibm_ws390_orb_ORBEJSBridge_refreshSRAggregator							
			266127B0	+00000072		SUBP00L2		Call	
36422E28	+0000013C	com/ibm/ws390/orb/ORBEJSBridge.refreshSRAggregator(ILjava/la							
			36422E28	+0000013C		SUBP00L0		Call	
36426388	+00000052	com/ibm/ws390/orb/SRAggregator.getSRObjectElementHT()Ljava/u							
			36426388	+00000052		SUBP00L0		Call	
36414658	+000000A4	com/ibm/ws390/management/ServantMBeanInvoker.invokeSpecified							
			36414658	+000000A4		SUBP00L0		Call	
7C200830	+000002BC	INVFRMMI	7C200830	+000002BC		*PATHNAM		Call	
7C5DA5B0	+00000030	c_invokerFromMMI							
			7C5DA5B0	+00000030		*PATHNAM		Call	
7CCF1788	+0000026E	mmipSelectInvokeJavaMethod							
			7CCF1788	+0000026E		*PATHNAM		Call	
7CCE4618	+00005C6A	mmipExecuteJava							
			7CCE4618	+00005C6A		*PATHNAM		Call	
7CCFD298	+000002DC	mmijExecuteJavaFromJIT							
			7CCFD298	+000002DC		*PATHNAM		Call	

Figure 5-7 Traceback using IPCS data

The information found in the traceback might be sufficient to find the module or method name. When the traceback provided by IPCS does not go far enough, a tool called svcdump.jar can be used. Refer to 21.2, “JVM dump and heap analysis tools” on page 254, for more details on how to download and run the svcdump.jar tool.

- i. With the information obtained from the svcdump.jar tool, such as abend code, module, and method name, determine in which component the abend was taken. You can use this information to debug the module or search IBM support data for related information and possible fix. After searching IBM support data on the Web, we found the PK06080 APAR to address our problem.

- j. If you cannot find a solution, prepare MustGather documentation and contact IBM.

Refer to 20.3, “SVC dumps” on page 247, for more information about how to use SVCDUMP for problem analysis in WebSphere for z/OS.

14. Set SLIP for SVCDUMP.

Dumps can be suppressed by the dump analysis and elimination process. When this is the case, you must set a SLIP to capture a dump when the abend occurs. This is done using the SLIP SET z/OS command.

Example 5-2 on page 58 shows a SLIP that was used to capture a dump for an EC3 abend. It uses a wild card for the reason code so that any of the 0413000* abend reason

codes that occur are allowed. The ASIDLIST is a list of address space IDs (ASIDs) for current, home, primary, secondary, and other address spaces in the dump, should you be in cross memory with them at the time.

Example 5-2 Example for setting a SLIP

```
SLIP SET,A=SVCD,COMP=EC3,REASON=0413000x,ID=WEC3,MATCHLIM=20,  
SDATA=(ALLNUC,CSA,GRSQ,LPA,LSQA,PSA,RGN,SQA,SUM,SWA,TRT),  
ASIDLST=(0,H,I,P,S)
```

An example of a SLIP that is used to capture a dump for a 0C4 abend is:

```
SLIP SET,A=SVCD,COMP=0C4,ID=R0BS,JOBNAME=(WASR0BS),END
```

Refer to the *z/OS MVS System Commands*, SA22-7627-11, for a full description and syntax of the SLIP command.

15. Reproduce or wait for reoccurrence.

Sometimes you must have an SVCDUMP to determine the cause of the abend. Therefore, you must set a SLIP and try to reproduce the error. If you cannot reproduce the error, then wait for the problem to reoccur with the SLIP in place.

If you are unsure about the most appropriate SLIP, contact IBM support for assistance.

Hang

This chapter explains what a hang is. The flow chart and step-by-step descriptions that we provide can help you analyze the problem and find its cause. We also mention the analysis tools and refer to information sources that are related to this symptom.

6.1 What is a hang?

A WebSphere Application Server hang is when an address space for the application server on z/OS is not processing work and appears to be idle. A hang can be noticed when the server no longer processes requests. The scope of this symptom is restricted to a hang in the WebSphere for z/OS environment.

6.2 Symptom flow chart: Hang

Figure 6-1 shows the flow chart for this symptom. Each box has a number that refers to the analysis in more detail.

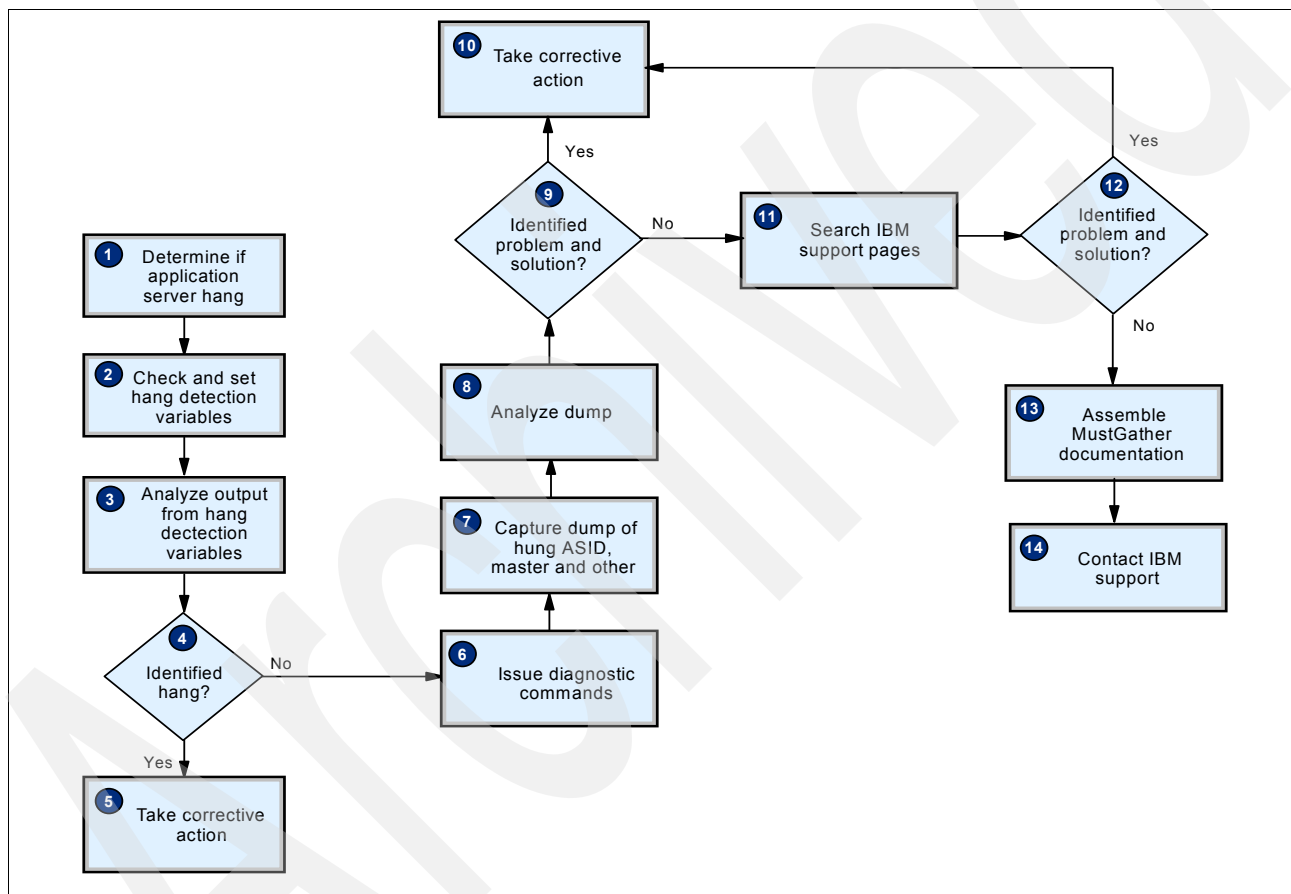


Figure 6-1 Flow chart for symptom: Hang in the application server

6.3 Diagnosing a hang

Follow the these steps to analyze and solve the hang problem:

1. Determine whether it is an application server hang.

Typically, a user who is not getting a browser response reports a hang. This could be a hang in the application, the HTTP server, or the WebSphere Application Server.

You can perform either of these checks to determine whether it is a WebSphere for z/OS hang:

- Run a simple test by entering a display command for the WebSphere for z/OS server that you suspect has a hang and wait for a response, for example:

MODIFY <server Name>,DISPLAY

If you do not get a response, it is very likely that the WebSphere for z/OS server is hung.

- Check the latest time stamps in the WebSphere for z/OS server logs against the current system time. How much time has passed between system time and last recorded activity?

If you get a response from the server or the last recorded activity is close to the system time, then the problem might be with the HTTP server or the application itself.

Check whether the request from the browser has arrived at the HTTP server by reviewing the HTTP server access logs and error logs. For more information, see 19.5, “IBM HTTP Server logs and trace” on page 232. If the HTTP request has arrived at the server and the HTTP server is responding to requests, then it might be the application that is hung.

2. Check and set hang detection variables.

WebSphere for z/OS V6 has a thread hang detection option and it is enabled by default. To adjust the hang detection policy values or to disable them, go to the Administrative Console and select **Servers** → **Application Servers** → *server_name*. Under **Infrastructure**, select **Administration** → **Custom Properties**. Then, select **New**. The properties are:

- Name: com.ibm.websphere.threadmonitor.interval
- Name: com.ibm.websphere.threadmonitor.threshold
- Name: com.ibm.websphere.threadmonitor.false.alarm.threshold

For full explanation of the thread detection properties, search for the WSVR0605W message at the WebSphere for z/OS V6 Information Center at:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

If the WebSphere for z/OS server address space is hung, you cannot set or adjust these properties at this time. You must wait until the hang situation is cleared.

3. Analyze output from hang detection variables.

If you have the hang detection variables set, then you can see WSVR0605W messages in your servant region job log output (Example 6-1).

Example 6-1 WSVR0605W message example

```
Trace: 2005/07/20 15:45:45.013 01 t=6C1AC8 c=UNK key=P2 (13007002)
ThreadId: 00000016
FunctionName: com.ibm.ws.runtime.component.ThreadMonitorImpl
SourceId: com.ibm.ws.runtime.component.ThreadMonitorImpl
Category: WARNING
ExtendedMessage: BB000221W: WSVR0605W: Thread "HManager.thread.pool: 0"
(00000030) has been active for 642415 milliseconds and may be hung. There is/are
1 thread(s) in total in the server that may be hung.
```

Note the thread name and ID. They might help you determine the problem when you are searching the IBM support pages or reporting the problem to IBM.

4. Have you identified the hang?

Using the hang detection properties and the output from the WSVR0605W message, have you been able to identify the hung thread? If yes, then take corrective action as described in step 5. If no, issue some diagnostic commands and prepare for a dump as described in step 6.

5. Take corrective action.

If you were able to identify the hung thread with the information from the variables, then fix the problem, or check with the application programmers or the IBM support pages for information about this specific thread.

If the thread problem continues to occur, you must further diagnose the cause by issuing specific commands and prepare for a dump. See step 6.

6. Issue diagnostic commands.

Many factors can cause an application server to hang. Usually a dump is required to diagnose the cause, but first you must determine the address space for the dump.

At a minimum, you should dump all the application server address spaces: the controller region, servant regions, and the daemon (the control region adjunct if appropriate).

Issue this command to determine which address spaces should be included in your dump:

D GRS,C

This MVS command displays enqueue contention on the system. Example 6-2 shows the output after the authors issued the command. The BACK1HFS job currently holds an OMVS file system latch. The WEBPR01 job, which is the application server that is currently hung, is waiting for the enqueue.

Example 6-2 Sample output from D GRS,C command

```
ISG343I 01.54.41 GRS STATUS 177
NO ENQ RESOURCE CONTENTION EXISTS
LATCH SET NAME:  SYS.BPX.A000.FSLIT.FILESYS.LSN
CREATOR JOBNAME: OMVS      CREATOR ASID: 000D
LATCH NUMBER:  221
  REQUESTOR  ASID  EXC/SHR   OWN/WAIT
  BACK1HFS   0087  EXCLUSIVE  OWN
  WEBPR01   011A  SHARED    WAIT
```

This means that you should dump OMVS, BACK1HFS, and WEBPR01 control and servant regions in the example to diagnose the hang problem.

Note: The DISPLAY GRS contention command might have to be routed to all systems if a sysplex is involved. For more information about Global Resource serialization (GRS) and other GRS commands that are available to analyze connection, refer to *z/OS V1R6.0 MVS Planning: Global Resource Serialization*, SA22-7600-03

7. Capture dump of hung ASID and others.

In a hang situation, it is always advisable to dump the OMVS address space and data space. Capture a dump of the relevant application server address spaces using the MVS DUMP command (Example 6-3).

Example 6-3 MVS DUMP command

```
DUMP COMM=(Descriptive name for this Webserver dump)
R rn,SDATA=(CSA,SQA,RGN,TRT,GRSQ,LPA,LSQA,SUM,NUC,PSA),CONT
R rn,JOBNAME=(OMVS,controlregionname,servantregionname),CONT
R rn DSPNAME ('OMVS'.*),END
```

Note: The more address spaces that you include in your dump, the larger the dump will be. Be sure that the dump completes successfully because you might encounter a space limitation problem, MAXSPACE. Any problems with the dump are recorded in the syslog. Use the MVS CHNGDUMP command to increase MAXSPACE.

8. Analyze dump.

Use IPCS to analyze the dump. For detailed information about IPCS and working with SVCDUMP, refer to 20.3, “SVC dumps” on page 247.

Analyzing a dump can be done in many ways and the same information can be found by invoking different commands and options:

– IP ST REGS WORKSHEET

You can use this IPCS command to verify dump title, time, and date.

– IP ANALYZE RESOURCE

This command results in contention analysis. It shows resources, such as a latch or file system, that are causing contention. Note the type of resource and the TCBs that are holding the resources.

– IP SUMM FORMAT

If more than one address space is dumped, you must supply the ASID or job name (servant region name). This command shows all of the TCBs for that address space. Use these TCB addresses in the LEDATA commands.

– IP VERBX LEDATA 'TCB(tcb_addr) NTHREADS(*)'

This command lists the TCB traceback information. This is the module flow for the task. From this, you can determine what modules the ASID could be waiting in and whether it is a valid wait step for that module or method. If it is in application code, then you might need to consult with the application owner.

Run this command against the TCBs that appeared in the analyze resource output as holding a resource that other TCBs are waiting for. You could also use any TCB that is listed in the summary format output. Example 6-4 shows sample output from the LEDATA command.

Example 6-4 output from ip verbx ledata 'tcb(00AC6B58) nthreads() asid(00a8)'*

TCB(00AC6B58) NTHREADS(*) ASID(00A8)

Language Environment Product 04 V01 R6.00

To Display Additional Information:

IP VERBX LEDATA 'CAA(6A5CF520)DSA(6BB181C0) ALL'

Information for enclave main

Information for thread 1CD56F600000003F

PCB Address: 1C50D080

TCB Address: 00AC6B58

Registers and PSW:

GPR0..... 00000086	GPR1..... 6BB18A00	GPR2..... 243C4680	GPR3..... 6BB18AEC
GPR4..... 6BB181C0	GPR5..... 6BB18A8C	GPR6..... 6BB18A90	GPR7..... 00000000
GPR8..... 9C7A1412	GPR9..... 6B59C048	GPR10.... 6BB18AE8	GPR11.... 6BB18AE0
GPR12.... 6A5CF520	GPR13.... 6BB18A44	GPR14.... 9C7A1500	GPR15.... 00001300

Traceback:

- IP VERBX GRSTRACE

With WebSphere dumps, analysis using only IPCS might not give the detail that is required to show the Java method and class that are necessary for debugging. The `svcdump.jar` tool, which is run on a Windows platform against an SVCDump, can provide this detail. Example 6-5 shows the output generated by the `svcdump.jar` tool.

Example 6-5 output from `svcdump.jar` against `tcb(00AC6B58)`

Dsa	Entry	Offset	Function
---	-----	-----	-----
6bb181c0	1c7a1408	49e5a8ea	recv
6bb182a0	665fbb98	0000015a	NET_Recv
6bb18340	665f78f0	00000292	Java_java_net_SocketInputStream_socketRead0
6bb187c0	69d17380	0000011e	java/net/SocketInputStream.socketRead0(Ljav
6bb18880	69d1f6d0	00000240	java/net/SocketInputStream.read([BII)I
6bb18a00	728f8d98	0000006e	java/io/BufferedInputStream.fill()V
6bb18b20	6978d9f0	00000096	java/io/BufferedInputStream.read1([BII)I


```

6bb18c40 69792fb8 0000009e java/io/BufferedInputStream.read([BII)I
6bb18ee0 7154c6d0 00000104 com/ibm/net/ssl/www2/protocol/http/y.a(Lcom
6bb19000 72d17bd8 00000302 com/ibm/net/ssl/www2/protocol/http/bb.getIn
6bb19240 70069cd8 00000410 gov/zena/mss/appenv/MSSLoginModule.getURLOu
6bb19380 70ff1e40 0000132c gov/zena/mss/appenv/MSSLoginModule.login()Z
6bb19560 70732400 000000aa sun/reflect/GeneratedMethodAccessor24.invo

```

Java stack:

Method	Location
-----	-----
java/net/SocketInputStream.socketRead0	Native Method
java/net/SocketInputStream.read	SocketInputStream.java(C
java/io/BufferedInputStream.fill	BufferedInputStream.java
java/io/BufferedInputStream.read1	BufferedInputStream.java
java/io/BufferedInputStream.read	BufferedInputStream.java
com/ibm/net/ssl/www2/protocol/http/y.b	(Compiled Code)
com/ibm/net/ssl/www2/protocol/http/y.a	(Compiled Code)
com/ibm/net/ssl/www2/protocol/http/bb.getInputStream	(Compiled Code)
java/net/URL.openStream	URL.java(Inlined Compile
gov/zena/mss/appenv/MSSLoginModule.getURLOutput	MSSLoginModule.java(Comp
gov/zena/mss/appenv/MSSLoginModule.login	MSSLoginModule.java(Comp

Important: In a hang situation, it is also important to check the settings for the *protocol_http* type variables, especially:

protocol_http_timeout_output_recovery

A SESSION setting cleans up the socket, but no attempt is made to disrupt the running of a dispatched HTTP request in a servant (region). The thread cannot be terminated when a timer for the thread hits.

A SERVANT setting causes the whole address space to go down when the timer hits. This is seen as a timeout EC3abend.

9. Have you identified the problem?

After analyzing the dump, have you been able to find the reason for the hang? Have you identified the resource that was held causing other work to wait? If yes, then take the corrective action as described in step 10.

If no, search the IBM support pages as described in step 11.

10. Take corrective action.

If you were able to identify the reason for the hang, take appropriate steps to fix the problem so that you can run the application server as desired.

11. Search IBM support pages.

If you have been unable to identify the cause of the hang, you should search the IBM support pages. Having analyzed the output from the commands that you issued and having reviewed the data in the dump, you now have information that you can use as a basis for such a search. You can start your search at the WebSphere for z/OS support site:

http://www-306.ibm.com/software/webservers/appserv/zos_os390/support/

From this site, you can click several links to access other support sites that are related to WebSphere Application Server, its components, and z/OS. Refer to Chapter 3, “Information sources” on page 25, which provides many valuable links and resources for solving problems in WebSphere for z/OS.

12. Have you identified the problem and solution?

After searching the support pages, were you able to find the cause of the problem or find a solution? If no, prepare to contact IBM support as described in step 13.

If yes, go to step 10.

13. Assemble MustGather documentation.

For more information about MustGather, see “MustGather” on page 16. Read the document *MustGather: Read first for WebSphere Application Server for z/OS* for help with assembling the appropriate documentation.

For a hang, supply information about:

- The version of your WebSphere Application Server and build level
- The version of the operating system and service level (PUT)
- The description of the problem

Include background information such as when the problem started to occur, whether it occurs at certain times, and whether there have been any changes to the system such as new maintenance or new applications.

- Controller region and servant region job logs
- Syslog showing diagnostic commands issued and their output
- SVCDump

14. Contact IBM support.

Refer to Chapter 2, “Contacting IBM: Information” on page 13, for information about and procedures for contacting IBM support.



Timeout

This chapter explains what a timeout is. The flow chart and step-by-step descriptions in this chapter can help you analyze the problem and find its cause. We also mention the analysis tools and reference information sources that are related to this symptom.

7.1 What is a timeout?

A timeout happens when a specific work unit does not complete in the specified amount of time. An application might not respond because it is busy processing a request, it is hung, or it is waiting for a request to return. However, a timeout is most likely an indication that something does not work right and therefore cannot be done within a certain amount of time.

If a process takes longer than expected, the session ends with a timeout when the timeout variables have been set to a low limit. The WebSphere for z/OS environment has several timeout variables that can be extended. You can also set the variables to higher levels if you want the process to write more information to the log before it times out to analyze the problem behind the timeout. However, increasing the timeout variable values does not solve the root cause of a problem.

7.2 Symptom flow chart: Timeout

Figure 7-1 shows the flow chart for a timeout. Each box has a number referring to the analysis in more detail.

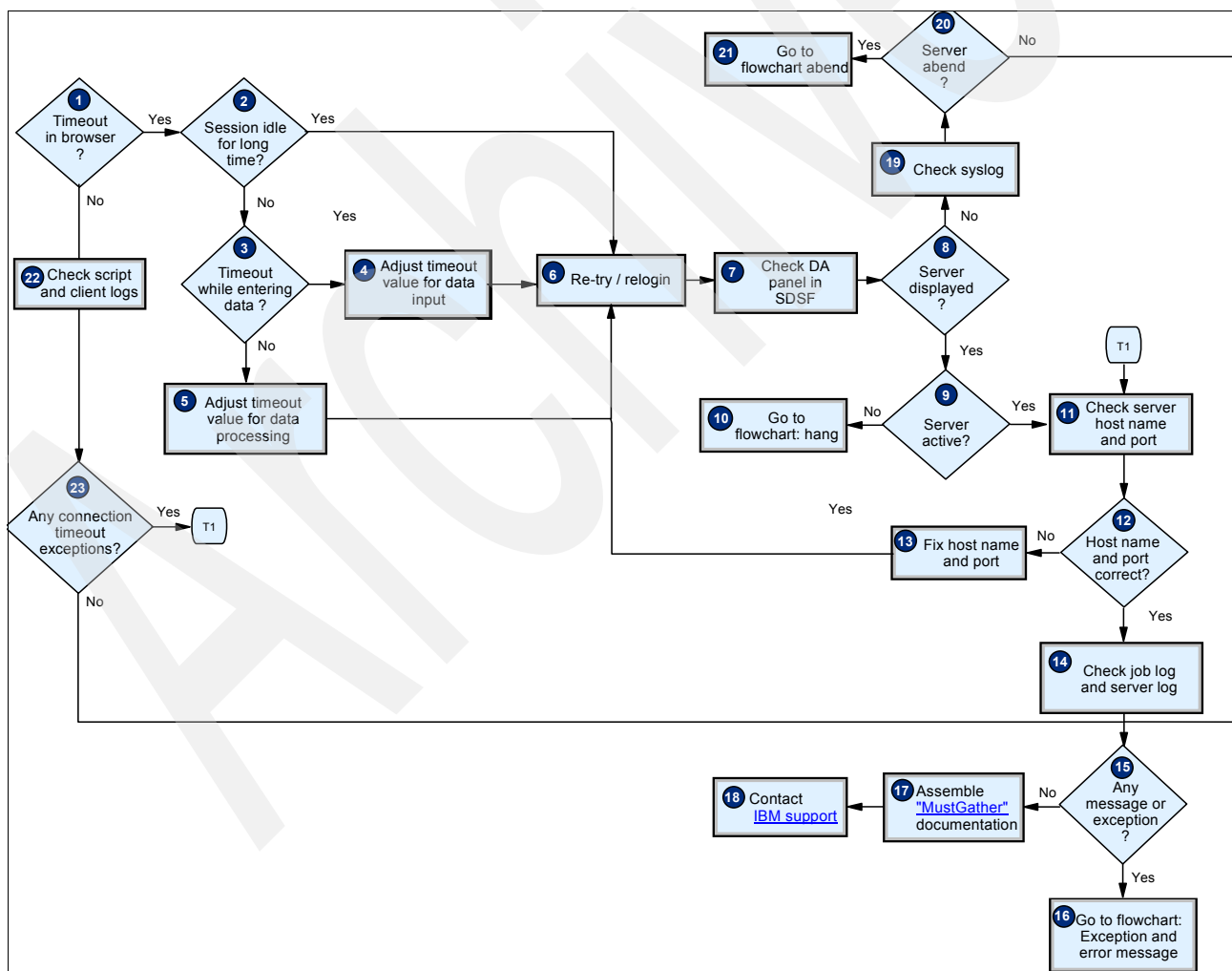


Figure 7-1 Flow chart for symptom: Timeout

7.3 Diagnosing a timeout

Follow these steps to analyze and solve the problem:

1. Is the timeout in a browser?

Do you experience the timeout in the WebSphere Administrative Console or any other Web client browser as a message with the keyword “timeout” or does your browser appear to be idle although you expect it to respond to a request? If yes, then proceed with step 2.

If no, then we assume that another component timed out and you should check the appropriate logs; see steps 16 through 21.

2. Is a session idle for a long time?

Web applications often time out when they are idle for more than a certain time because of specific session settings. Depending on how well the application is written, a message informing you of the session timeout might appear in the WebSphere Administrative Console (Figure 7-2). Otherwise, the session might just end. Was your session idle for a longer duration? If no, proceed with step 3.

If yes, access the application again, and see step 6.

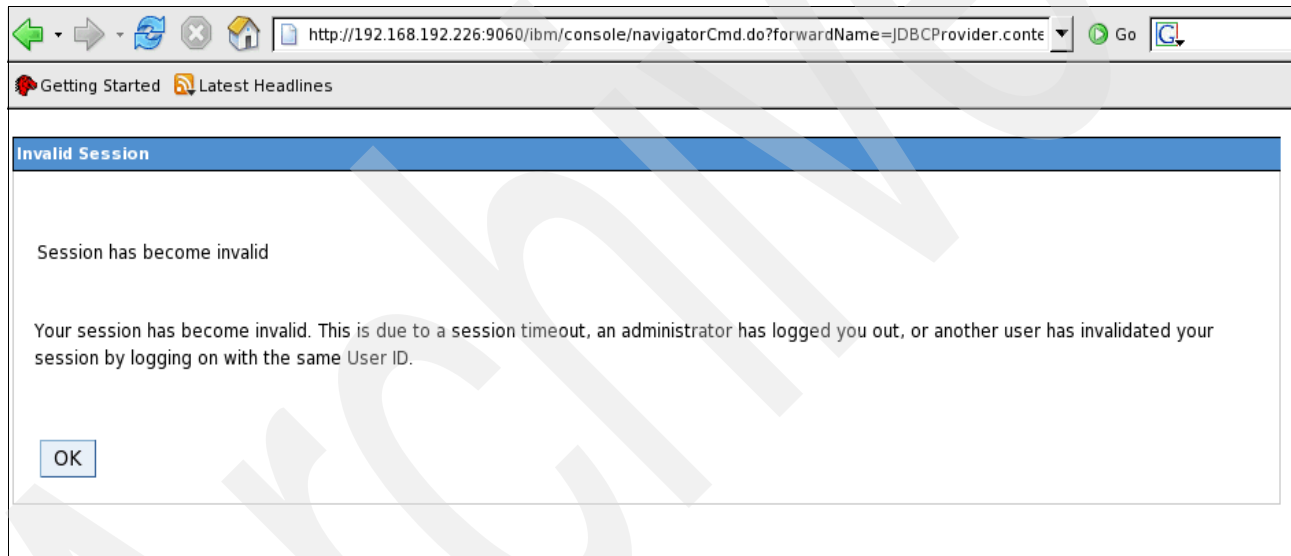


Figure 7-2 Session timeout

3. Timeout while entering data?

Do you experience a timeout while attempting to enter data in a Web client? If yes, adjust the timeout values as shown in step 4. If no, proceed with step 5.

4. Adjust timeout value for data input.

Because a timeout can occur when the application server is not tuned with timeout values, consult the application developers and testers to determine how much time was scheduled for data entry. If this information is not available, change the value of the `protocol_http_timeout_input` variable to zero. Do extensive testing to determine how long it takes to enter data, and adjust the `protocol_http_timeout_input` value accordingly.

To learn how to change the `protocol_http_timeout_input` value, search for “HTTP Transport timeout variables” or “controlling behavior through timeout values” at the WebSphere for z/OS Information Center:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

There are several articles that provide more information about timeout and how to set the timeout values that WebSphere for z/OS uses. Also, search for “understanding how timers work” for more links and references about timeout.

Note: For information about setting the Administrative Console session timeout value, search for “setting the session timeout for the administrative console” at the WebSphere for z/OS Information Center.

Proceed with step 6.

5. Adjust timeout value for data processing.

Because a timeout can occur when the application server is not tuned with timeout values, consult the application developers and testers to determine how much time was scheduled for data processing. If you do not have this information, change the value of the *protocol_http_timeout_output* variable to 0. Do extensive testing to determine how long it takes to process data and adjust the *protocol_http_timeout_output* value accordingly.

To learn how to change the *protocol_http_timeout_output* value, search for “HTTP Transport timeout variables” or “controlling behavior through timeout values” at the WebSphere for z/OS Information Center:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

There are several articles that provide more information about timeout and how to set the timeout values that WebSphere for z/OS uses. Also search for “understanding how timers work” for more links and references about timeout.

6. Retry and log in again.

Either your session was idle for more time than expected or you had to tune the timeout values for WebSphere for z/OS. Now try to access the Web client again, call the appropriate URL, and log in if necessary.

If you cannot get to the Web site you intended, you cannot log in, or the Web client appears to be frozen, proceed with step 7.

7. Check DA panel in SDSF.

Check the server activity by going to the DA (Active users) panel under SDSF and analyze the list that is produced (Figure 7-3).

SDSF DA P1A P1A PAG 0 SIO 11 CPU 12/**/ LINE 1-6 (6)																	
COMMAND INPUT ==> SCROLL ==> CSR																	
NP	JOBNAME	StepName	ProcStep	JobID	Owner	C	Pos	DP	PGN	Real	Paging	SIO	CPU%	ASID	ASIDX	EXCP-Cnt	C
	W60ADMC	W60ADMC	BBODAEMN	STC53228	WSDMNCR1	NS	FE		0	7465	0.00	0.00	0.00	143	000F	5,225	
	W60ASA2S	W60ASA2S	BBOSR	STC54099	ASSR1	IN	FB		0	115T	0.00	0.00	0.53	183	0007	738,183	
	W60AM01S	W60AM01S	BBOSR	STC53231	DMSR1	IN	FB		0	207T	0.00	0.00	0.53	187	000B	5,385,902	
	W60ANP1	W60ANP1	BBOCTL	STC53230	ASCR1	NS	FE		0	62T	0.00	0.67	10.70	171	000B	1,966,135	
	W60ASA2	W60ASA2	BBOCTL	STC53233	ASCR1	NS	FE		0	82T	0.00	0.67	10.70	180	0004	766,198	
	W60AM01	W60AM01	BBOCTL	STC53227	DMCR1	NS	FE		0	82T	0.00	0.67	10.70	189	000D	2,787,880	

Figure 7-3 Active Jobs in DA panel

8. Is the server up?

Check whether the server name (application servers and Deployment Manager for the Administrative Console) appears in the list (see JOBNAME in Figure 7-3). Are both the controller region and servant region in the list and are they up? If yes, proceed with step 9.

If no, go to step 19.

9. Is the server active?

Check whether the server that your Web browser is trying to access is active. You can check the server activity in the DA panel under SDSF. See whether the CPU% number (see Figure 7-3 on page 70) for the specific server is changing, an indication that your server is active. If it is not active, proceed with step 10.

If it is active, check the server host and port as shown in step 11.

10. Go to the flow chart for hang.

If the request was sent from the client side, and the server is up but did not respond, then the server could be hung. At this point you explore the hang symptom. See Chapter 6, "Hang" on page 59, for more information.

11. Check server host and port.

Verify the accuracy of the server host name and port number in the script or client that you are using to access the application server.

In the command line of the wsadmin script, the host name is followed by -host, and the port number is followed by -port. If the host name is not specified, the program uses the host name specified in the TCP/IP profile.

Also check whether the server listens on the right port by issuing **netstat -a** in the system with the specified application server. Check the output, verifying that the port that you are trying to access is open and that it is the right type. For admin scripting and clients, it must be an ILOP port, listed as *protocol_iiop_port* in the application server joblog.

12. Is the server host and port correct?

Are the host name and port correctly defined, active, and accessed by the client or script?
If no, proceed with step 13.

If yes, check the logs as described in step 14.

13. Fix host name and port.

According to your findings in previous steps, change the host name and port in the client or script, in the application server definition, or in your TCP/IP configuration to ensure that you can access the right application server. Go to step 6 when you are finished.

14. Check job and server logs.

If your server is active, review the recent server log or job log entries and look for any abnormal activity in the server and any token or keyword pointing to an error or problem. Also check the syslog for messages that indicate a problem in the system environment. Chapter 4, "Exceptions and error messages" on page 41 and Appendix A, "Messages and codes" on page 311 have more details about how to identify messages and exceptions.

If First Failure Data Capture (FFDC) is enabled, messages are written to the specified FFDC files (see Figure 7-4 on page 72). Check them for any messages or exceptions.

```
Trace: 2005/10/03 10:46:25.304 01 t=6C61C8 c=UNK key=P8 (13007002)
ThreadId: 0000002f
FunctionName: initialize
SourceId: com.ibm.ws.ffdc.IncidentStreamImpl.ServiceLogger
Category: INFO
ExtendedMessage: FFDC0009I: FFDC opened incident stream file
/P13/WebSphere/V6R1M2A/AppServerB/profiles/default/logs/ffdc/PLXMCLA1_P1BNLA1_s
erverB2_STC54100_W60ASB2S_05.10.03_14.46.25.0.txt
```

Figure 7-4 FFDC file information in trace

Note: For more information about FFDC, see 19.3, “First Failure Data Capture” on page 219, and the WebSphere Information Center at:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

15. Is there any message or exception?

Have you found a message or exception in the log that might be worth exploring? If yes, then proceed with step 16.

If no, then go to step 17.

16. Go to the flow chart for exceptions and error messages.

If you found a message or exception that is not self-explanatory but might lead you to the root cause of the problem, refer to Chapter 4, “Exceptions and error messages” on page 41.

17. Assemble MustGather documentation.

MustGather documents help with problem determination and save time resolving PMRs. For more information about MustGather, see “MustGather” on page 16. You can find MustGather documents by searching for the word *mustgather* at:

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

Read the document: *MustGather: Read first for WebSphere Application Server for z/OS*, for help assembling the appropriate documentation, available at:

<http://www.ibm.com/support/docview.wss?uid=swg21176043>

The minimum information that is necessary is:

- Problem description
Include information related to when the problem first started to occur.
- Software version and maintenance (build) level
You find the information in the job log of your application server. Search for “build level,” and you obtain a line similar to this:
BBOM0007I CURRENT CB SERVICE LEVEL IS build level cf10515.05 release
WAS601.ZNATV date 04/15/05 12:55:41.
- Operating system version and maintenance (PUT) level
- The job log of the application server in question (include both controller and servant region job logs)
- Any dumps or traces triggered by the problem

See also 2.3, “Before you contact IBM support” on page 15. Then proceed with step 18 to contact IBM Support.

18. Contact IBM support.

If you need to contact IBM support, refer to Chapter 2, “Contacting IBM: Information” on page 13, for instructions.

Provide the information that is outlined in the MustGather documentation step.

19. Check syslog.

If you cannot see the server in the DA panel, you might have to start the server first (if this was not done before) or analyze why the server stopped by checking the syslog for specific messages that indicate a problem.

Search for the last log entry related to the server in question and trace it back to where a problem occurred. There are several reasons for a server failing.

20. Is there a server abend?

A server abend can be a reason for a request timing out. Did you find an “abend” message for your server in the syslog? If yes, then proceed with step 21.

If not, then check for other error messages or exceptions and proceed with step 15.

21. Go to the flow chart for abend.

The abend EC3 code followed by a 0413xxxx reason code indicates an abend caused by a timeout. To identify the cause of the EC3-0413xxxx abend timeout (by debugging the EC3 dump using the IPCS debugging tool), search for “task with the EC3 abend” at the WebSphere Information Center:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

22. Check logs.

Check all other logs for a message or exception that might be related to the timeout:

- If you were running a **script** and experienced a timeout, these scripts (using Java Command Language or Jython) can run into the same type of problems as the Administrative Console.

Scripts generally put the log information into a file. Logs are not generated by default, so scripts have to be modified so that the errors can be written to logs. Search the Information Center for script information if you are not sure how to modify scripts for logging. Rerun the script to generate a log.

- If you were running other types of clients, such as an Object Request Broker (ORB) client, Java Message Server (JMS) client, Remote Method Invocation/Internet Inter-ORB protocol (RMI/IIOP) client or any fat client, and the client timed out while waiting for a response from the server or from an external source, check whether the clients generated any logs. If there are no logs, modify the clients to generate logs. Contact your application developers for the modification.
- Check all possible logs of **other** resources that could cause the timeout. Other resources include networks, databases, and any resources that are not visible in the browser or in the client logs. Other resource timeouts sometimes show up in the system log or job log.

Check the syslog (issue LOG in SDSF panel). For the job log, go to the server address spaces and scan through the logs for any timeouts. Instead of looking through all job logs, you can use log streams. Define LOGSTREAMs for all the address spaces, then run the BBORBLOG tool from the Time Sharing Option (TSO) command prompt to receive the job logs. See 19.2, “WebSphere error log (BBORBLOG)” on page 216, for more information about WebSphere logs and tools.

23. Are there any connection timeout exceptions?

If your script or client is waiting for a connection to establish that fails, or for a request sent through this connection but not returned, you might receive a connection timeout or reset exception in your log. Example 7-1 shows the connection error with the message number WASX7023E.

Example 7-1 Connection exception

WASX7023E: Error creating "SOAP" connection to host "WEBXXX.POK.IBM.COM";
exception information:
com.ibm.websphere.management.exception.ConnectorNotAvailableException:
[SOAPException: faultCode=SOAP-ENV:Client; msg=Connection reset;
targetException=java.net.SocketException: Connection reset]

You will most likely find another message that informs you of the reason and gives you the location for the specific log to review. In our case, we received message code WASX7213I, which means that this scripting client is not connected to a server process. It also pointed to the log file /WebSphere/V6R0M0A/AppServer1/profiles/default/logs/wsadmin.traceout for additional information.

Do you see a connection timeout in your script/client or system log? If yes, go to step 11.

If no, then go to step 15.

Does not stop

This chapter explains what does not stop is, and provides you with a flow chart and step-by-step descriptions that can help you analyze the problem and find its cause. We also mention the analysis tools and reference information sources that are related to this symptom.

8.1 What is the does not stop symptom?

This symptom occurs when you want to stop one of your WebSphere for z/OS V6 application servers. Using the Administrative Console, you attempt to stop the application server, but it does not stop. When you enter a STOP command for a region in one of the z/OS operator consoles, the address space stays active (that is, the address space does not disappear from your display).

You could enter the cancel command in the system console to force the region to stop and free the address space, but that would not solve the underlying problem, and it would be difficult to analyze the problem without the logs from the address space that is malfunctioning.

8.2 Symptom flow chart: Does not stop

Figure 8-1 shows the flow chart for does not stop. Each box has a number referring to the analysis in more detail.

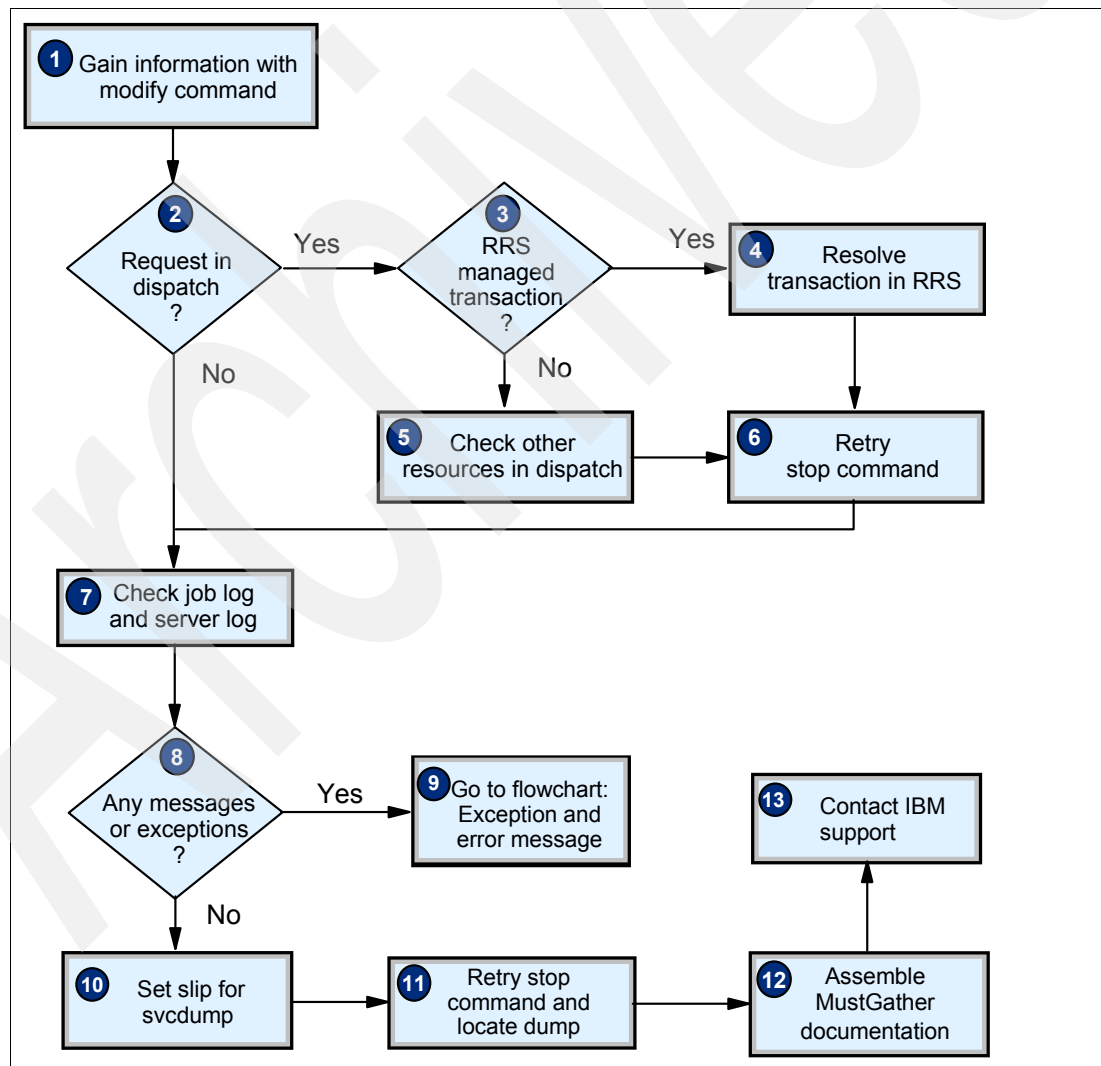


Figure 8-1 Flow chart for symptom: Does not stop

8.3 Diagnosing the symptom

Follow these steps to analyze and solve the problem:

1. Obtain information with the MODIFY command.

If the address space (controller region or daemon) did not stop after you waited for some time, obtain more information about this address space by using the MODIFY (F) command.

If you are not familiar with the MODIFY command, you can explore it by issuing this command in the z/OS operator console:

```
F WS491,HELP
```

WS491 is the job name of your controller region. The output shows you all of the options for the MODIFY command. See 18.2, “z/OS MODIFY commands” on page 196.

You can also search for “displaying WebSphere Application Server work” at the WebSphere Information Center:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

For explanations and demonstrations of the MODIFY command, see the Troubleshooting and support section at the WebSphere Information Center.

2. Is the request in dispatch?

When you display work repeatedly and see requests *in dispatch*, such as servlets, EJBs, or message driven beans (MDBs), this is an indication of a potential reason for the region not stopping. Did you see requests in dispatch? If yes, proceed with step 3.

If no, check the logs; see step 7.

3. Is it an RRS managed transaction?

Any direct corrective action is not immediately available. You should explore the specific request and learn why it is still in dispatch.

Is the request an RRS managed transaction? If yes, proceed with step 4.

If no, check on the resource in dispatch; see step 5.

4. Resolve transaction in RRS.

In the case of a transaction that is managed by RRS, you can get more information from the RRS panels in your TSO session. Look at the RRS Unit of Work recovery list. Depending on the information in the list, you might be able to release the transaction by committing or backing out. Talk to your application developers and system administrators about the consequences of these actions (data recovery and consistency issues).

For more information about RRS, refer to *Systems Programmer's Guide to Resource Recovery Services (RRS)*, SG24-6980.

5. Check other resources in dispatch.

Analyze the dispatch message in more detail and determine what resource (EJB, servlet, MDB) is still active. Consult your application developer or system administrator about what other resources are requested from the message and how much time it should take to release them. There might be a lock on the resources. Take the appropriate actions to release the resources.

6. Retry the stop command.

Try the stop command for the application server in question again. You might have solved the problem by resolving the resource in dispatch or the passing of additional time might have released resources in the background and given the server enough time to close down the address space properly. If the server is still running, proceed with step 7.

7. Check job log and server log.

If your server is active, review the recent server log or job log entries and look for any abnormal activities in the server and any token or keyword pointing to an error or problem. Also check the syslog for messages about a problem in the system environment.

Chapter 4, “Exceptions and error messages” on page 41 and Appendix A, “Messages and codes” on page 311 have more details about how to identify messages and exceptions.

If FFDC is enabled, messages are written to the specified FFDC files (Figure 8-2). Check them for any messages or exceptions.

```
Trace: 2005/10/03 10:46:25.304 01 t=6C61C8 c=UNK key=P8 (13007002)
ThreadId: 0000002f
FunctionName: initialize
SourceId: com.ibm.ws.ffdc.IncidentStreamImpl.ServiceLogger
Category: INFO
ExtendedMessage: FFDC0009I: FFDC opened incident stream file
/P13/WebSphere/V6R1M2A/AppServerB/profiles/default/logs/ffdc/PLXMCLA1_P1BNLA1_s
erverB2_STC54100_W60ASB2S_05.10.03_14.46.25.0.txt
```

Figure 8-2 FFDC file information in trace

Note: For more information about FFDC, see 19.3, “First Failure Data Capture” on page 219 and visit the WebSphere Information Center.

8. Is there any message or exception?

Have you found a message or exception in the log that might be worth exploring? If yes, then proceed with step 9.

If no, then go to step 10.

9. Go to the flow chart for exceptions and error messages.

If you found a message or exception that is not self-explanatory but might lead you to the root cause of the problem, refer to Chapter 4, “Exceptions and error messages” on page 41.

10. Set dump.

Dumps can be suppressed by the dump analysis and elimination process. When this is the case, you set a SLIP to capture a dump. This is done using the SLIP SET z/OS command.

Refer to *z/OS MVS System Commands*, SA22-7627-11, for a full description and syntax of the SLIP command.

11. Retry stop command and analyze dump.

Try the stop command for the application server in question again. You might have solved the problem by resolving the resource in dispatch or a dump might have been created.

Usually the name of the SVCDUMP data set is recorded in the syslog. If you are not sure whether a dump was taken or which data set the dump was written to, then search for the word “dump” in the syslog and locate any messages pertaining to the dump.

If there is a problem capturing the dump, an IEAxxx type message is issued, such as:

```
IEA911I PARTIAL DUMP ON MVS.01MP.DMP00056 678
IEA043I SVC DUMP REACHED MAXSPACE LIMIT - MAXSPACE=xxxx MEG
```

In that case, you should fix the dump problem first before you attempt to analyze the dump because crucial information might not be written to the dump. Also ensure that your WebSphere for z/OS servers have the authority to create and write to the dump data sets.

To analyze the SVCDUMP, invoke the z/OS MVS IPCS. There are several methods for analyzing SVCDUMP with IPCS and data from the SVCDUMP. We outline one approach here:

- a. Invoke IPCS and verify that you have the correct dump by checking the dump title, date, and time. To display this information, issue:

```
ip st validate worksheet
```

Figure 8-3 shows an example of output from this command.

```

MVS Diagnostic Worksheet

Dump Title: COMPON=WEBSPPHERE Z/OS,  COMPID=5655I3500,ISSUER=BBORLEXT,
                ABEND IN BB0ORB  /UNKNOWN

CPU Model 2084 Version 00 Serial no. 012345 Address 01
Date: 02/18/2005      Time: 12:38:22.102475 Local

Original dump dataset: SYSPRD1.PLXA.SVCD.D050218.H123818.C2.N00011

Information at time of entry to SVCDUMP:

HASID 04EC  PASID 04EC  SASID 04EC  PSW 070D1000 9C038948
```

Figure 8-3 Output from IPCS *ip st worksheet validate*

- b. In the same display, scroll down to see the diagnostic data report section and verify the abend code, reason code, and module name.
- c. Locate the PSW address of where the abend condition occurred and verify the module name in the summary format report, which is displayed using the following command:

```
ip summ format
```

Scroll to the bottom of the report and use **find previous** to locate the RTM2WA SUMMARY and control block data:

```
f 'rtm2wa summary' prev
```

- d. This takes you to RTM2WA SUMMARY, which shows RTM data. This is the time of error information as shown in Figure 8-4 on page 80. Note the PSW address.

RTM2WA SUMMARY					

+001C	Completion code		840C4000		
+008C	Abending program name/SVRB address	007C2070	00000000		
+0094	Abending program addr		00000000		
GPRs at time of error					
0-3	00000000	215D3F00	00000000	21D126A0	
4-7	34326EB0	273C2250	7CA9E300	A667FA00	
8-11	A667F7AA	00000000	267908E8	221BDAA0	
12-15	33FE6C50	00000000	267918E0	0405A00C	
+007C	EC PSW at time of error	072C2000	A667FA42	00040004	00000000
+00DC	SDWACOMP		00000000		
+00E8	Return code from recovery routine-00				
	Continue with termination-implies percolation				
+00E0	Retry Address returned from recovery exit		00000000		
+00E4	RB Address for retry		00000000		
+000C	CVT Address		00FCB018		
+0038	RTCT Address		00FB24E0		
+00C8	SCB Address		007C4AC0		

Figure 8-4 IPCS ip summ format output showing RTM2WA SUMMARY

To determine the TCB address, you must scroll up a little to find the RTM2WA control block data and note the TCBC value.

In our example, we have a PSW of 072C2000 A667FA42. The second word is a 31-bit address.

For information about the format of the PSW, refer to *z/Architecture Principles of Operation*, SA22-7832-03.

- e. Now locate the address in the dump storage. This is done from the IPCS main menu. In our example, we located address 2667FA42 (Figure 8-5).

ASID(X'04EC')	ADDRESS(2667FA42.)	STORAGE	-----
Command ==>			
2667FA42	A784	000A181B	9856E0D8 0D764700 xd...q.\Q....
2667FA50	FFA95800	D00018B0	A7F40005 18DB58B0 .z..}...x4.....
2667FA60	B00012BB	A774FF76	5820487C BF1F201C ...x.....@....
2667FA70	A784000D	A7AA0FF8	9856A0D8 0D764700 xd..x..8q..Q....

Figure 8-5 Browse dump storage using IPCS

- f. From the PSW address, scroll up and try to determine the module name using the eye catchers in the dump such as those in Figure 8-6.

2667E3B0	F2F0F0F5	F0F1F1F2	F2F0F3F1	F5F2F0F1	2005011220315201
2667E3C0	F0F2F0F0	0010E6F5	F1F0F2F0	F44D8282	0200..W610001(bb
2667E3D0	9696A299	815D0000	00C300C5	00C500F1	oosra)...C.E.E.1
2667E3E0	00005EF0	00000080	90684788	A74AFF80	..;0.....hx¢..
2667E3F0	0D805870	50485860	504C4100	00005810&..-&<.....

Figure 8-6 Search for eye catchers in dump storage near PSW address

The eye catchers are the ASCII characters found at the right of the storage. In our example, we found the **bboosra** module name.

- g. Obtaining a module name often is sufficient, but when WebSphere for z/OS is involved, it is sometimes necessary to go further and obtain the related method name.

Examine the traceback data. Using the TCB from the RTM2WA information, enter.

```
ip verbx ldata 'tcb(007c07f8) nthreads(*)'
```

When the output is displayed, locate the traceback information (Figure 8-7).

Traceback:									
PU Addr	PU Offset	Entry	E Addr	E Offset	Statement	Load Mod	Service	Stat	
2667F7A0	+000002A2	SRAggregator::refresh(JNIEnv*,_jobject*,_jobject*)							
			2667F7A0	+000002A2		SUBP00L2		Call	
266127B0	+00000072	Java_com_ibm_ws390_orb_ORBEJSBridge_refreshSRAggregator							
			266127B0	+00000072		SUBP00L2		Call	
36422E28	+0000013C	com/ibm/ws390/orb/ORBEJSBridge.refreshSRAggregator(ILjava/la							
			36422E28	+0000013C		SUBP00L0		Call	
36426388	+00000052	com/ibm/ws390/orb/SRAggregator.getSRObjectElementHT()Ljava/u							
			36426388	+00000052		SUBP00L0		Call	
36414658	+000000A4	com/ibm/ws390/management/ServantMBeanInvoker.invokeSpecified							
			36414658	+000000A4		SUBP00L0		Call	
7C200830	+000002BC	INVFRMMI	7C200830	+000002BC		*PATHNAM		Call	
7C5DA5B0	+00000030	c_invokerFromMMI							
			7C5DA5B0	+00000030		*PATHNAM		Call	
7CCF1788	+0000026E	mmipSelectInvokeJavaMethod							
			7CCF1788	+0000026E		*PATHNAM		Call	
7CCE4618	+00005C6A	mmipExecutedJava							
			7CCE4618	+00005C6A		*PATHNAM		Call	
7CCFD298	+000002DC	mmijExecutedJavaFromJIT							
			7CCFD298	+000002DC		*PATHNAM		Call	

Figure 8-7 Traceback using ipcs ldata

- h. The information found in the traceback in Figure 8-7 might be sufficient to find the module or method name. When the traceback provided by IPCS does not go far enough, a tool called svcdump.jar can be used. Refer to 21.2, “JVM dump and heap analysis tools” on page 254 and *WebSphere for z/OS V5 JVM Dump and Heap Analysis Tools*, REDP-3950, for more details about how to download and run the svcdump.jar tool.
- i. With the information that you obtained from the svcdump.jar tool, such as abend code, module, and method name, determine in which component the abend was taken. You can use this information to debug the module or search IBM support data for related information and a possible fix.
- j. After searching IBM support data, we found APAR PK06080 to address our problem.
- k. If you cannot find a solution, prepare MustGather documentation and contact IBM.

Further information about IPCS can be found in:

- *z/OS V1R6.0 MVS IPCS Commands*, SA22-7594-05
- *z/OS V1R2.0 MVS IPCS User's Guide*, SA22-7596-01.

Also refer to 20.3, “SVC dumps” on page 247, for more information about how to use SVCDUMP for problem analysis in WebSphere for z/OS.

12. Assemble MustGather documentation.

MustGather documents help with problem determination and save time resolving PMRs. For more information about MustGather, see “MustGather” on page 16. Read the document *MustGather: Read first for WebSphere Application Server for z/OS*, for help with assembling the appropriate documentation. It is available at:

<http://www.ibm.com/support/docview.wss?uid=swg21176043>

The minimum information necessary is:

- Problem description

Include information related to when the problem first started to occur.

- Software version and maintenance (build) level

You find the information in the job log of your application server. Search for “build level” and you obtain a line similar to this:

```
BBOM0007I CURRENT CB SERVICE LEVEL IS build level cf10515.05 release  
WAS601.ZNATV date 04/15/05 12:55:41.
```

- Operating system version and maintenance (PUT) level
- The job log of the application server in question (including both controller and servant region job logs)
- Any dumps or traces triggered by the problem.

See also 2.3, “Before you contact IBM support” on page 15. Then proceed with step 13 to contact IBM Support.

13. Contact IBM support.

If you need to contact IBM support, refer to Chapter 2, “Contacting IBM: Information” on page 13, for instructions.

Provide the information outlined in the MustGather documentation step.



Job failed

This chapter explains the job failed symptom. The flow chart and step-by-step descriptions in this chapter can help you analyze the problem and find its cause. We also mention the analysis tools and reference information sources that are related to this symptom.

9.1 What is job failed?

If a job that you were starting to run on your system stopped or did not run successfully, we say it is “job failed.” This is mostly relevant in the installation or migration phase when you run jobs to configure your environment or when you start an address space for your WebSphere Application Server.

9.2 Symptom flow chart: Job failed

Figure 9-1 shows the flow chart for this symptom. Each box has a number that refers to the steps that are described in 9.3, “Diagnosing the job failed symptom” on page 85.

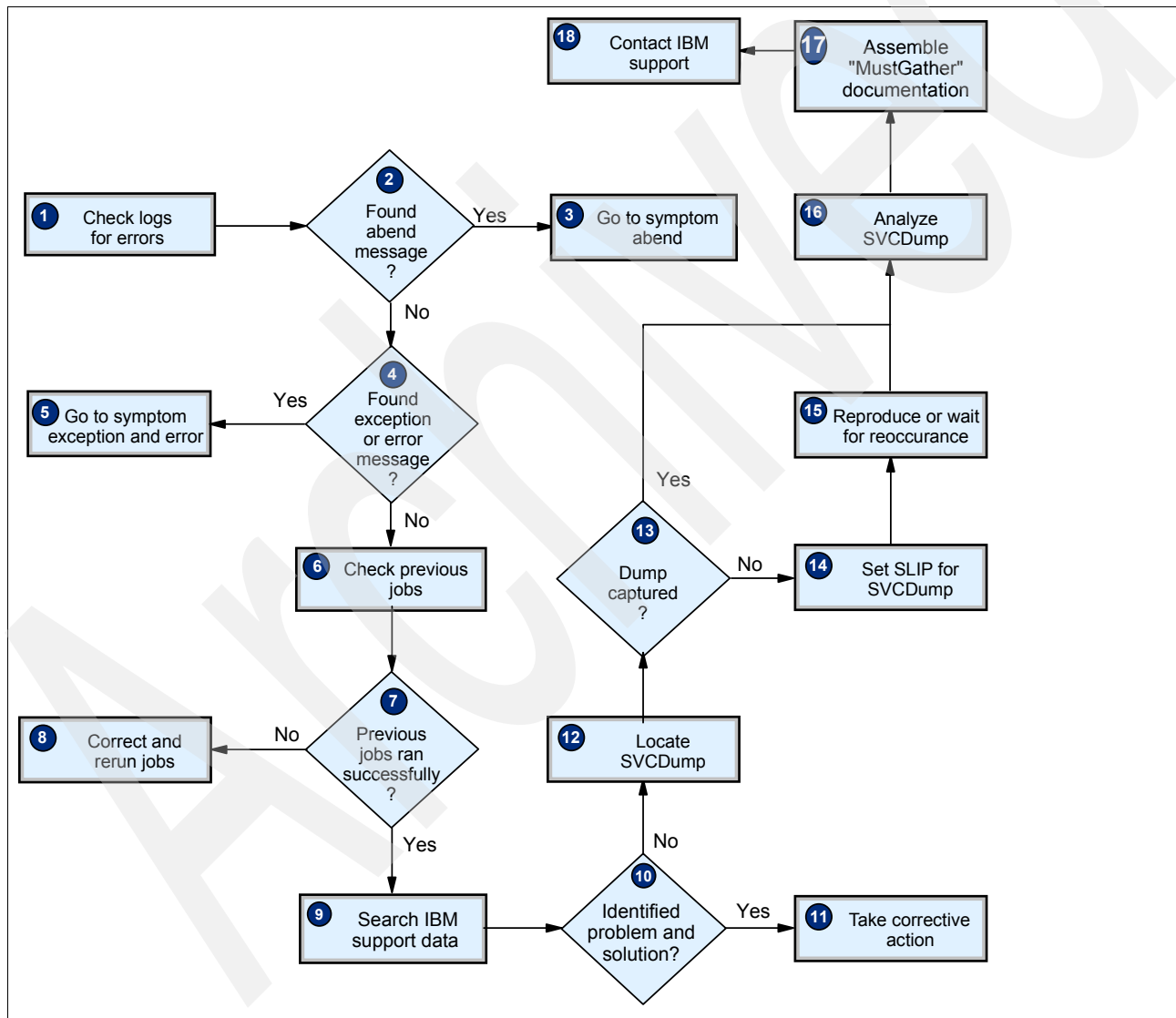


Figure 9-1 Flow chart for symptom: Job failed

9.3 Diagnosing the job failed symptom

Follow these steps to analyze and solve the problem:

1. Check logs for error messages.

Check error messages in the syslog (console log, SYSPRINT, SYSOUT) or job log. Look for keywords such as *error*, *abend*, *failed*, *problem*, and *stopped*. If you cannot find a message that points to a problem, check whether the job output is passed to any files such as <fileName>.out and <fileName>.err. If they do, check for error messages in those files.

If there is an error message, determine the kind of error or message.

Specific error messages generally have a message identifier that is followed by a message. An abend is usually associated with the keyword *abend* or a code, such as IEA. Java exceptions might not have a message identifier, but they are usually self explanatory or followed by a stack trace. See Chapter 4, “Exceptions and error messages” on page 41, for more information about identifying messages and exceptions.

2. Have you found an abend message?

Did you find an abend code or the keyword *abend* in the log? If yes, go to step 3. If no, then proceed with step 4.

3. Go to the flow chart for abend.

We explore the symptom abend in more detail in Chapter 5, “Abend” on page 49. Follow the steps that are described to analyze this symptom and find its cause.

4. Have you found exception or error message?

Did you find an exception or error message with a trace or message code to research? If yes, go to the appropriate symptom; see step 5. If no, check the previous jobs; see step 6.

5. Go to the flow chart for exception and error messages.

We explore this symptom in more detail in Chapter 4, “Exceptions and error messages” on page 41. Follow the steps that are described to analyze this symptom and find its cause.

6. Check previous jobs.

Sometimes specific jobs or started tasks must be running or must have been started previously before the current job or task can be run or can complete successfully. Analyze the jobs and started tasks to see if there are dependent jobs or started tasks. Check their logs to see whether they have succeeded. Compare the instructions with the actual results.

7. Have previous jobs ran successfully?

Have all dependent jobs completed successfully and without an error? Have all dependent started tasks started properly, without abending or issuing error messages? If no, then go to step 8. If yes, then proceed with step 9.

8. Correct and rerun jobs.

Analyze the errors and problems of the dependent jobs and started tasks, determine the problem causes, and solve them. Then rerun the jobs and check their output for success before you rerun the job in question (the one that failed previously).

9. Search IBM support data.

Depending on the life cycle stage that you are in with your WebSphere for z/OS application, refer to the corresponding chapter in Part 3, “Problem avoidance and best practices” for typical problems and how to fix or avoid them in that particular stage.

Search IBM support Web sites, such as the WebSphere for z/OS support site at:

http://www-306.ibm.com/software/webservers/appserv/zos_os390/support/

From this site, you can follow several links to other support sites that are related to WebSphere Application Server, its components, and z/OS. See Chapter 3, “Information sources” on page 25, which provides many valuable links and resources for solving problems in WebSphere for z/OS.

When you are searching problem databases for information or fixes that might be related to the job failed symptom, consider that such problems are reported in many formats. You might have to alter your search keyword to find a match.

10. Have you identified the problem and solution?

After you searched the support pages, were you able to find the cause of the problem and a solution? If yes, then take the corrective action by proceeding with step 11. If no, go to step 12.

11. Take corrective action.

The information that you have found using the IBM support data might have provided the following solutions:

- An existing APAR and PTF fix for your problem that is available for you to apply.
- Other reports of your symptoms that include a procedure for fixing the problem.

In such cases, follow the instructions that are provided, or apply the information to your specific problem. If you were able to solve it, document the problem and the fixes you have applied in your system change documentation for your specific WebSphere for z/OS environment for later reference. Otherwise, proceed with step 12.

12. Locate SVCDump.

You might find more information or hints regarding the failing job in the dump. Usually the name of the SVCDUMP data set is recorded in the syslog. If you are not sure whether a dump was captured or which data set the dump was written to, then search for the word *dump* in the syslog and locate any messages pertaining to the dump.

If there was a problem capturing the dump, an IEAxxx type message is issued, such as:

```
IEA911I PARTIAL DUMP ON MVS.01MP.DMP00056 678
IEA043I SVC DUMP REACHED MAXSPACE LIMIT - MAXSPACE=xxxx MEG
```

In that case, you should fix the dump problem before you attempt to analyze the dump because crucial information might not be written to the dump. Also, ensure that your WebSphere for z/OS servers have the authority to create and write to the dump data sets.

13. Was a dump captured?

Was a dump captured? Were you able to locate the dump? If no, then you must set a SLIP. Go to step 14. If you located the dump, then prepare to analyze the dump as described in step 16.

14. Set SLIP for SVCDump.

Dumps can be suppressed by the dump analysis and elimination process. When this is the case, you should use the z/OS command SLIP SET to set a SLIP to capture a dump when the symptom occurs. Example 9-1 shows a SLIP used to capture a dump for an abend EC3 (a started task failure). It uses a wild card for the reason code so that any of the 0413000* abend reason codes that occur are allowed. The ASIDLST is for current, home, primary, and secondary address spaces and can include other address spaces in the dump if you are in cross memory with them at the time.

Example 9-1 Example for setting a SLIP

```
SLIP SET,A=SVCD,COMP=EC3,REASON=0413000x,ID=WEC3,MATCHLIM=20,
SDATA=(ALLNUC,CSA,GRSQ,LPA,LSQA,PSA,RGN,SQA,SUM,SWA,TRT),
ASIDLST=(0,H,I,P,S)
```

Refer to the *z/OS MVS System Commands*, SA22-7627-11, for a full description and syntax of the SLIP command.

If you are unsure about the most appropriate SLIP, contact IBM support for assistance.

15. Reproduce or wait for reoccurrence.

With the SLIP set, try to reproduce the error. If you cannot reproduce the error, then wait for the problem to reoccur with the SLIP in place.

16. Analyze the SVCDUMP.

To analyze the SVCDUMP, invoke IPCS. Several methods can be used to analyze this symptom using IPCS and data from the SVCDUMP. We outlined one approach in “Analyze the SVCDUMP” on page 54 in Chapter 5, “Abend” on page 49. Further information about IPCS can be found in:

- *z/OS V1R6.0 MVS IPCS Commands*, SA22-7594-05
- *z/OS V1R2.0 MVS IPCS User's Guide*, SA22-7596-01

See 20.3, “SVC dumps” on page 247, for more information about how to use SVCDUMP for problem analysis in WebSphere for z/OS.

If you cannot find a solution, prepare MustGather documentation and contact IBM as described in step 17.

17. Assemble MustGather documentation.

Prepare the MustGather documentation for IBM support. For more information about MustGather, see “MustGather” on page 16.

For a failing job or started task, you should provide the following material:

- Problem description

Include information related to when the problem first started to occur.

- Software version and maintenance (build) level

Find this information in the job log of your application server. Search for “build level” to obtain a line similar to this:

```
BBOM0007I CURRENT CB SERVICE LEVEL IS build level cf10515.05 release  
WAS601.ZNATV date 04/15/05 12:55:41.
```

- Operating system version and maintenance (PUT) level
- The job log of the abending address space (including both controller and servant region job logs)
- The SVCDUMP triggered by the failing job or the SLIP

18. Contact IBM support.

If you need to contact IBM support, refer to Chapter 2, “Contacting IBM: Information” on page 13, for instructions. Provide the information outlined in the MustGather documentation step.

Archived

No response

This chapter explains what the no response symptom means. The flow chart and step-by-step descriptions in this chapter can help you analyze the problem and find its cause. We also mention the analysis tools and reference information sources related to this symptom.

10.1 What does “no response” mean?

This symptom represents a wide range of problems and can have many causes. If an application or a WebSphere for z/OS address space that has operated without problems previously does not respond to a request or a command (from the Administrative Console or the System Console), we refer to the symptom as “no response.”

10.2 Symptom flow chart: No response

Figure 10-1 shows the flow chart for no response. Each box has a number that refers to the steps that are described in more detail in 10.3, “Diagnosing the no response symptom” on page 91.

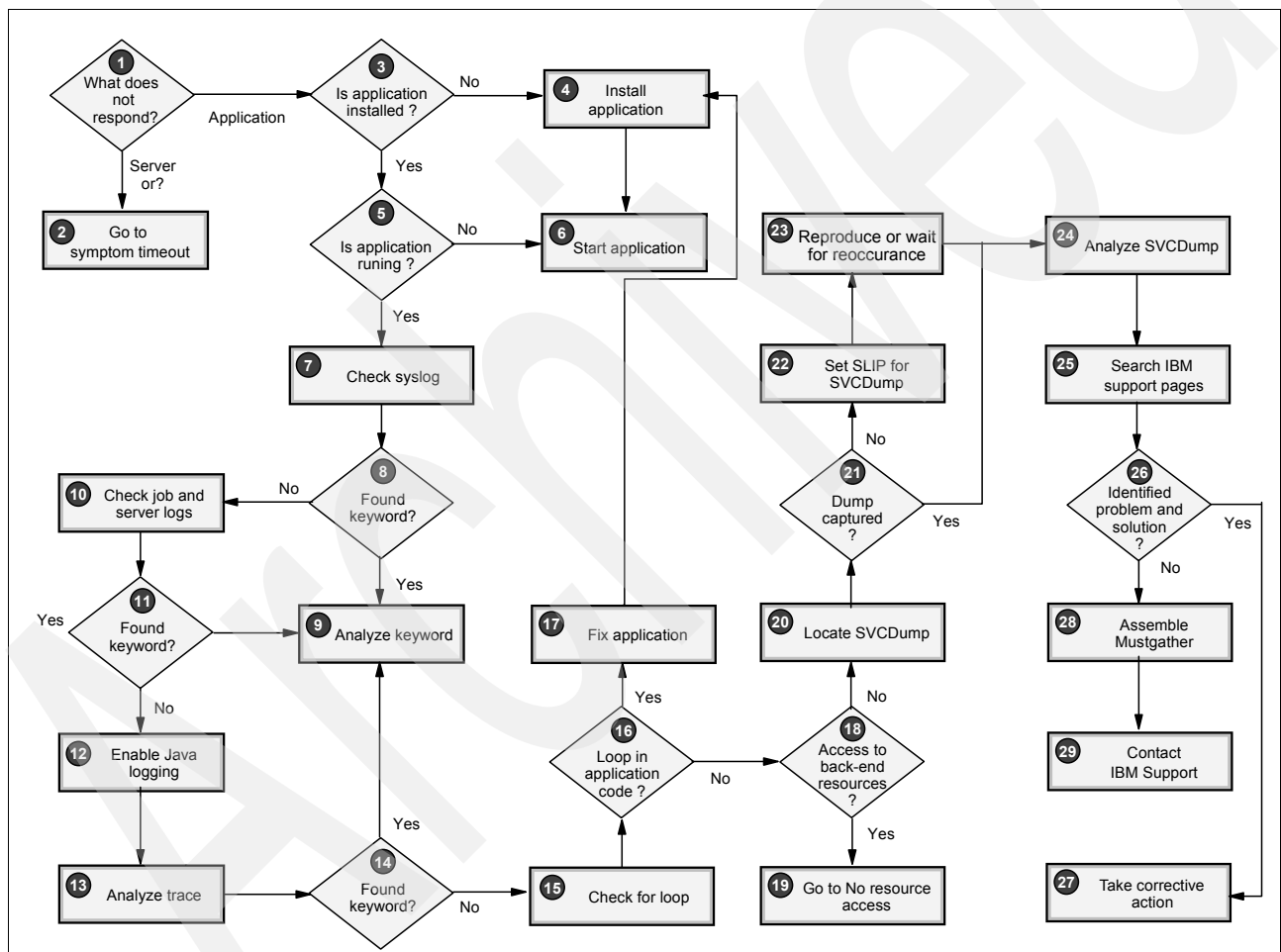


Figure 10-1 Flow chart for symptom: No response

10.3 Diagnosing the no response symptom

Symptoms such as hang, timeout, or does not stop are often the cause of no response. If you suspect that one of these symptoms is causing your no response or you have found a message that indicates one, see:

- ▶ Chapter 6, “Hang” on page 59
- ▶ Chapter 7, “Timeout” on page 67
- ▶ Chapter 8, “Does not stop” on page 75

Otherwise, follow these steps to analyze and solve this problem:

1. What does not respond?

To determine the cause of the problem, you must eliminate the areas that have not caused it. What does not respond? If an application request does not respond, go to step 3.

In any other case, proceed with step 2.

2. Analyze logs.

Even if you do not think your no response symptom is a timeout problem, see “Check DA panel in SDSF” on page 70. The steps described in Chapter 7, “Timeout” on page 67 might help to analyze the problem leading to the root cause.

Alternatively, go to step 7.

3. Is the application installed correctly?

To check the Administrative Console to see whether your application is installed correctly, go to **Application** → **Enterprise Applications**. Can you see your application in the list of installed applications? If no, then go to step 4.

If yes, then go to step 5.

4. Install application.

If the application does not appear in the list, you might not have installed it correctly. WebSphere Application Server provides several ways to install applications:

- The Administrative Console
- wsadmin scripts
- Java administrative programs that use JMX™ APIs
- Java programs that define a J2EE DeploymentManager object in accordance with J2EE Deployment API Specification (JSR-88)

To install your application using the Administrative Console, refer to “Installing application files” at the WebSphere for z/OS Information Center:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

After you install the application, start it. See step 6.

5. Is the application running?

If your application is correctly installed, you must check whether the application is started. In the Administrative Console, go to **Applications** → **Enterprise Applications**. All applications with a green arrow in the Status column have been started and are currently running (see Figure 10-2 on page 92).

Enterprise Applications		
Lists installed applications. A single application can be deployed onto multiple servers.		
⊞ Preferences		
<div> <div>Start</div> <div>Stop</div> <div>Install</div> <div>Uninstall</div> <div>Update</div> <div>Rollout Update</div> <div>Remove File</div> <div>Exp</div> </div>		
<div> <div>☑</div> <div>📄</div> <div>📶</div> <div>📶</div> </div>		
Select	Name ↕	Status ↻
<input type="checkbox"/>	DefaultApplication	➡
<input type="checkbox"/>	PlantsByWebSphere	➡
<input type="checkbox"/>	SamplesGallery	➡
<input checked="" type="checkbox"/>	TimeoutEAR	✖
<input type="checkbox"/>	TraderCICSEAR	➡
<input type="checkbox"/>	ivtApp	➡
<input type="checkbox"/>	query	➡
Total 7		

Figure 10-2 Status of installed applications in Administrative Console

Does your application have a green arrow? If no, proceed with step 6. If yes, go to step 7.

6. Start application.

To start the application, go to the Administrative Console and select **Applications** → **Enterprise Applications**. Select your application and click the **Start** button (see Figure 10-2.)

For more information about the status of applications and starting and stopping them, search for “Start and Stop applications” at the WebSphere for z/OS Information Center:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

7. Check syslog.

Check the syslog; issue **LOG** in the SDSF panel. Search for the last log entry related to the specific application server and trace it back to where a problem might have occurred. Look for any abnormal activity in the system or server and any token or keyword pointing to an error or problem.

8. Have you found a keyword?

Have you found any abnormal activity in the system or server or any token or keyword pointing to an error or problem in the log that might be worth exploring? If no, then go to step 10.

If yes, then proceed with step 9.

9. Analyze or search keyword.

If you found a symptom message or exception that is not self-explanatory but might lead you to the root cause of the problem, refer to the appropriate symptom chapters in this book:

- Chapter 5, “Abend” on page 49
- Chapter 6, “Hang” on page 59
- Chapter 7, “Timeout” on page 67

- Chapter 8, “Does not stop” on page 75
- Chapter 9, “Job failed” on page 83
- Chapter 11, “No resource access” on page 99

Chapter 4, “Exceptions and error messages” on page 41 and Appendix A, “Messages and codes” on page 311 have more details about how to identify messages and exceptions. Then, search for information about the specific message as described in step 23.

10. Check job and server logs.

Review the recent server log or job log entries and look for any abnormal activity in the server and any token or keyword pointing to an error or problem. Instead of looking through all job logs, you can use log streams. Define LOGSTREAMs for all the address spaces, then run the BBORBLOG tool from the TSO command prompt to receive the job logs. See 19.1, “Job logs and system log” on page 214 and 19.2, “WebSphere error log (BBORBLOG)” on page 216 for more information about WebSphere logs and tools.

If FFDC is enabled, messages are written to the specified FFDC files. Check them for any messages or exceptions. See 19.3, “First Failure Data Capture” on page 219, and the WebSphere Information Center at:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

If you use the IBM HTTP Server, you should analyze the appropriate logs that are described in 19.5, “IBM HTTP Server logs and trace” on page 232.

11. Have you found a keyword?

Have you found any abnormal activity in the system or server or any token or keyword pointing to an error or problem in the log that might be worth exploring? If no, then go to step 12.

If yes, then proceed with step 9.

12. Enable Java Logging.

To gather more detailed information relating to the execution path of a running application so that you can determine the root cause of the problem, use the Java Logging API. To enable this API:

- a. In the navigation pane, select **Servers** → **Application Servers** → <server_name>
- b. Click Diagnostic **Trace Service**.
- c. Select **Enable Log**.
- d. Select either **Memory buffer** or **file** and click Apply.
- e. To specify your configurations, go to **Troubleshooting** and select **Logging and tracing**.
- f. Click **Change Log Detail levels**.
- g. To make a static change to the configuration, click the Configuration tab. To change the configuration dynamically, click the Runtime tab. A list of well-known components, packages, and groups is displayed.
- h. Select a component, package, or group to set a logging level. Figure 10-3 on page 94 shows the Administrative Console panel for changing log level details. The list of components, packages, and groups shows all the components that are currently registered to the running server.

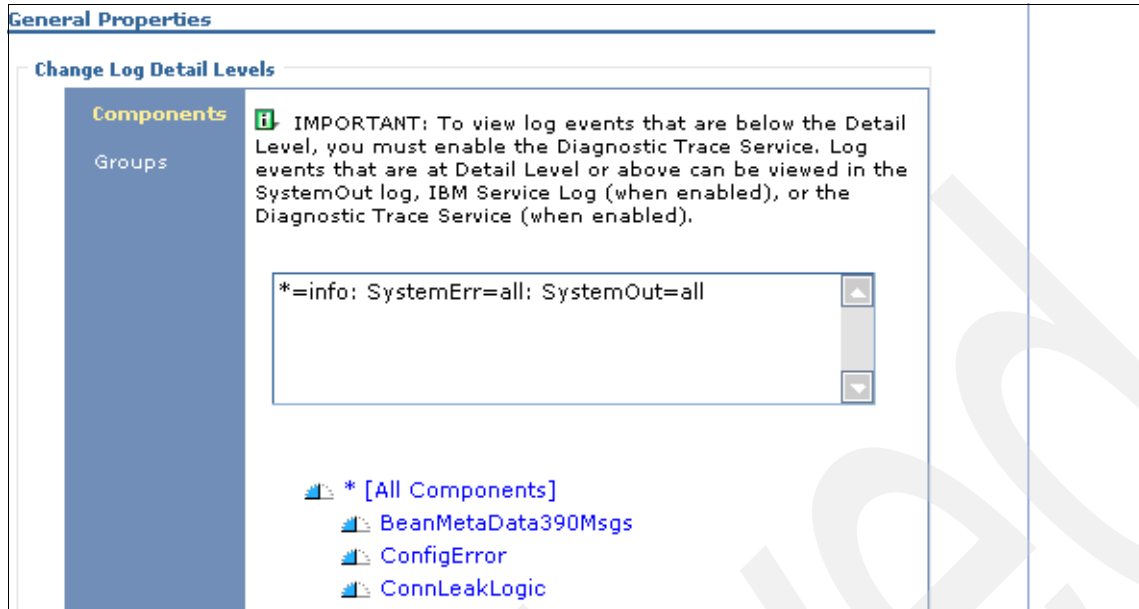


Figure 10-3 WebSphere for z/OS change log detail levels

- i. Click **Apply**, then **OK**.

See 19.4, “The Java Logging API” on page 227, for more detailed information.

After setting the trace, try to recreate the problem, stop the trace (or reduce the level), and analyze the trace that was produced for hints relating to the problem.

13. Analyze traces.

Go to the end of the file to see the last event that was recorded in the trace. The last entry always has the last method that was in memory (when an error occurred). The method name, its class, and its package name tell you the owner or provider of the package and are usually descriptive enough to hint at the root cause of the problem.

Also look for messages that are correlated to the HTTP header or client request/response problems of your application. Refer to “Trace Analyzer for WebSphere Application Server” on page 261, or use various other tools that are described in Chapter 21, “Diagnostic tools for WebSphere for z/OS” on page 253, for the analysis.

14. Have you found a keyword?

Have you found any abnormal activity in the trace, or any token or keyword pointing to an error or problem in the log that might be worth exploring? If no, go to step 15.

If yes, proceed with step 9.

15. Check for loop.

No response from an application can be caused by a loop in the application code.

Program loops can also result in a component hanging, usually followed by a timeout error. In some cases, things seem to work fine, but some tasks in the address space keep consuming system resources without producing a result. You can see this when CPU usage is high but nothing seems to justify it.

A dump or trace might be necessary because the log information is not sufficient for determining the cause of the loop or the unusual high resource consumption.

The indications of a loop are:

- You receive a repetitive message from a module waiting for work although nothing is being done, such as:
ExtendedMessage: <component> waiting for next server work
- You receive a repetitive message that a module is active and you are able to follow the executed address ranges, but the only thing changing is the time stamp.
- You receive a repetitive message from a module processing work or requests but the thread ID stays the same. Notice the ThreadID and FunctionName in Example 10-1. They might stay the same, but the trace header line with the time stamp changes if a loop is occurring. There might be several other messages between the repetitions. The shorter the loop cycle, the more likely you will be to recognize the loop.

Example 10-1 Looping thread

Trace: 2005/08/19 21:23:41.232 01 t=7D19C0 c=UNK key=P8 (13007002)
ThreadID: 0000006d
FunctionName: com.ibm.etools.validation.validationbuilder
SourceId: com.ibm.etools.validation.validationbuilder.UserStateRegistry
ExtendedMessage: closeUser - found UserPrefs: UserPreferences: nodeName:nd6552,
serverName: ws6552, userId: waspd2, refreshRate

If you suspect an application loop or hang:

- You can use IPCS to format a trace and analyze for recurring psw addresses.
Using the system trace, begin at the bottom of the file with the most recent entries. Look for any recurring or repetitive patterns in the system trace entries with the same psw addresses.
Using these addresses, browse them in dump storage using IPCS option 1. Scroll up from the address in storage and locate any eye catchers that identify module names. It might be difficult to determine the module names and relate them to Java methods. Use the TCB to run a traceback that might help identify what code the TCB is running. Use the command:

```
ip verbx ldata 'tcb(009C31C8) nthreads(*) asid(00fb)'
```


See 20.1.2, “Viewing CTRACE and JRas data through IPCS” on page 242.
- You can use the com.ibm.jvm.svc.dump.Dump utility to identify:
 - The thread under which a loop is occurring
 - The threads contending for resources or involved in a lockout
 - A thread waiting for some operation that is external to the server

A common mistake that causes a loop is forgetting to increase the counter in a **while** structure. Example 10-2 is a sample of Java code fragment for an infinite loop.

Example 10-2 Java code of an infinite loop

```
int ind = 0;  
int [] temp = {1, 3, 5, 7 };  
while (ind < temp.length) {  
    // other statements with no assignment to the variable "ind"  
}
```

If the *ind* variable is never greater or equal to *temp.length*, the loop will never terminate. The correct way to code such a condition is shown in Example 10-3 on page 96.

Example 10-3 Java code of an infinite loop fixed

```
int ind = 0;
int[] temp = {1, 3, 5, 7};
while (ind < temp.length) {
    // Others statements that do not assign anything to the variable "ind"
    ind ++;      // increment the counter
}
```

16. Is there a loop in the application code?

Have you found a loop in your application code? If yes, go to step 17.

If no, proceed with step 18.

17. Fix application.

If you have found an application loop, take information about the thread and component that is involved in the loop to the application developer to fix it. Redeploy fixed code and restart the application server as described in steps 4 and 5.

18. Do you have access to back-end resources?

If your application accesses other resources, the connection to those resources, or the resources themselves, might experience some problems. Do you access other resources? If yes, go to step 19.

If no, then proceed with step 20.

19. Go to No resource access.

If your application is accessing back-end resources, like a database, Java messaging service, or transaction service, check these resources. See Chapter 11, "No resource access" on page 99, for information about how to analyze this problem.

20. Locate SVCDump.

You might find more information or hints in the dump. Usually the name of the SVCDUMP data set is recorded in the syslog. If you are not sure whether a dump was taken or which data set the dump was written to, then search for the word *dump* in the syslog and locate any messages pertaining to the dump.

If there was a problem taking the dump, an IEAxxx type message is issued such as:

```
IEA911I PARTIAL DUMP ON MVS.01MP.DMP00056 678
IEA043I SVC DUMP REACHED MAXSPACE LIMIT - MAXSPACE=xxxx MEG
```

In that case, you should fix the dump problem before you attempt to analyze the dump because crucial information might not be written to the dump. Also ensure that your WebSphere for z/OS servers have the authority to create and write to the dump data sets.

21. Was a dump captured?

Was a dump captured? Were you able to locate the dump? If no, then you should set a SLIP. Go to step 22.

If you located the dump, then prepare to analyze the dump as described in step 24.

22. Set SLIP for SVCDump

Dumps can be suppressed by the dump analysis and elimination process. When this is the case, you should use the SLIP SET command to set a SLIP to capture a dump when the symptom occurs. Example 10-4 on page 97 shows a SLIP used to capture a dump for an abend EC3 (a started task failure). It uses a wild card for the reason code so that any of the 0413000* abend reason codes that occur are allowed. The ASIDLIST is for current,

home, primary, and secondary addresses to include other address spaces in the dump if you are in cross memory with them at the time.

Example 10-4 Example for setting a SLIP

```
SLIP SET,A=SVCD,COMP=EC3,REASON=0413000x,ID=WEC3,MATCHLIM=20,  
SDATA=(ALLNUC,CSA,GRSQ,LPA,LSQA,PSA,RGN,SQA,SUM,SWA,TRT),  
ASIDLST=(0,H,I,P,S)
```

Refer to the *z/OS MVS System Commands*, SA22-7627-11 for a full description and syntax of the SLIP command.

If you are unsure about the most appropriate SLIP, contact IBM support for assistance.

23. Reproduce or wait for reoccurrence.

With the SLIP set, try to reproduce the error. If you cannot reproduce the error, then wait for the problem to reoccur with the SLIP in place.

24. Analyze the SVCDUMP.

To analyze the SVCDUMP, invoke IPCS. Several methods can be used to analyze this symptom using IPCS and data from the SVCDUMP. We outlined one approach in “Analyze the SVCDUMP” on page 54 in Chapter 5, “Abend” on page 49. Further information about IPCS can be found in:

- *z/OS V1R6.0 MVS IPCS Commands*, SA22-7594-05
- *z/OS V1R2.0 MVS IPCS User's Guide*, SA22-7596-01

Refer to 20.3, “SVC dumps” on page 247 for more information about how to use SVCDUMP for problem analysis in WebSphere for z/OS.

25. Search IBM support data.

If you found a keyword, a key phrase, or error message in the dump that indicates the cause of the problem, search the IBM support Web sites and databases for a solution, especially the WebSphere for z/OS support site at:

http://www-306.ibm.com/software/webservers/appserv/zos_os390/support/

From this site, you can click several links to access other support sites that are related to WebSphere Application Server, its components, and z/OS. See Chapter 3, “Information sources” on page 25, which provides many valuable links and resources for solving problems in WebSphere for z/OS.

Tip: When searching problem databases for information or fixes, you may might have to alter your search keywords to find a match.

26. Identified problem and solution?

Were you able to find the cause of the problem and a solution? If yes, then take the corrective action in step 25.

If no, go to step 26.

27. Take corrective action.

If the problem is related to application code, or you suspect that the problem is related to the application and its access to resources, take the problem description, application logs, and traces produced to the application owner or developer.

Clarify which resources should be accessed and whether application access complies with J2EE and J2EE Connector Architecture (JCA) standards. Check whether connection properties are defined as intended, and walk through the trace step-by-step to find an

indication of what went wrong. Also check the application logs for messages and potential indicators, and analyze them together with the application developer to determine the exact point in the application where things went wrong.

For more information about resource access, JCA, pitfalls, hints and tips, and best practices, see the developerWorks Web site at:

<http://www.ibm.com/developerworks>

The information that you found using the IBM support data might have provided the following solutions:

- An existing APAR and PTF fix for your problem that is available for you to apply
- Other reports of your symptoms that include a procedure for fixing the problem

In such cases, follow the instructions that are provided or apply the information to your specific problem to solve it. Document the problem and the fixes that you have applied in your system change documentation for your specific WebSphere for z/OS environment for later reference.

28. Assemble MustGather documentation.

For more information about MustGather, see “MustGather” on page 16. You can find MustGather documents by searching on the word “mustgather” on the support Web site:

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

Read the document: *MustGather: Read first for WebSphere Application Server for z/OS*, for help assembling the appropriate documentation.

The minimum information necessary is:

- Problem description
Include information related to when the problem first started to occur.
- Software version and maintenance (build) level
You find the information in the job log of your application server. Search for “build level,” to obtain a line similar to this:

```
BBOM0007I CURRENT CB SERVICE LEVEL IS build level cf10515.05 release  
WAS601.ZNATV date 04/15/05 12:55:41.
```
- Operating system version and maintenance (PUT) level
- The job log of the abending address space (include both controller and servant region job logs)
- Any dumps or traces that are triggered by the problem or produced in the analysis

Tip: If you can, use the diagnostic tools that are mentioned in this chapter to identify the particular component or subcomponent that is responsible for the problem. In most cases, the components are in the application program code rather than product code from IBM. Present the component name together with the class and method name from the trace to your application development team or IBM (in the case of IBM components). This allows them to fix the code quicker.

29. Contact IBM support

If you need to contact IBM support, refer to Chapter 2, “Contacting IBM: Information” on page 13, for instructions.

Provide the information outlined in MustGather documentation step.



No resource access

This chapter explains the no resource access symptom. The flow chart and step-by-step descriptions in this chapter can help you analyze the problem and find its cause. We also mention the analysis tools and refer to information sources that are related to this symptom.

11.1 What is no resource access?

This symptom represents a wide range of causes. For the purpose of this chapter, we assume that “No resource access” means that an application had access to back-end resources such as database servers, transaction servers, or messaging services, but shows behavior that indicates that there is no longer access to these resources. For example, a request for data that is supposed to be displayed in a browser comes back with an error message or times out, or you see a “resource not found” message.

11.2 Symptom flow chart: No resource access

Figure 11-1 shows the flow chart for this symptom. Each box has a number that refers to the steps described in 11.3, “Diagnosing no resource access” on page 101.

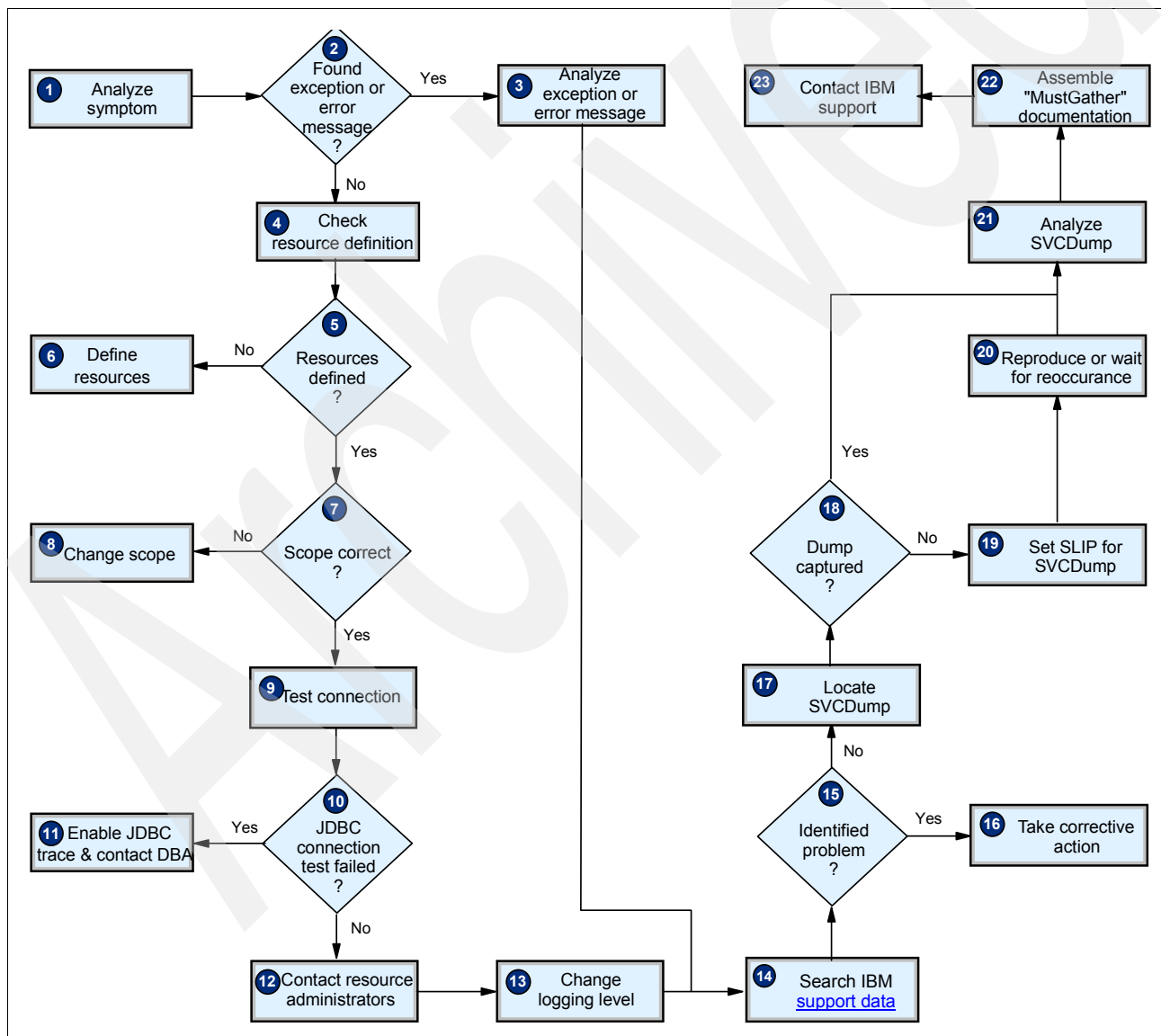


Figure 11-1 Flow chart for symptom: No resource access

11.3 Diagnosing no resource access

Because this symptom covers such a wide range of problems, the approaches to finding the root cause can vary widely. We suggest following the steps in Figure 11-1 on page 100 to eliminate all the possibilities of errors while attempting to access resources and to solve the problem.

1. Analyze the symptom.

There are three types of connectors to back-end data:

- A JDBC™ driver that is embedded with WebSphere if you are using DB2
- JMS that is based on IBM WebSphere MQ
- EIS JCA connectors that are provided by IBM if you are using IMS™ or CICS®

All of these connectors have traces and logs that can be collected to help diagnose errors.

When you experience the resource access problem, check whether you can see a specific error message in the browser, logs, or console. An incorrect configuration usually results in an exception logged in the syslog when the resource is being accessed. A timeout exception occurs when the back-end system server is unavailable.

2. Have you found an exception or error message?

Have you found a message or exception that might be related to the resource access problem and is worth exploring, such as the “insufficient authority” message in a job log (Example 11-1)? If no, then go to step 4.

If yes, then proceed with step 3.

Example 11-1 Sample failure message

```
ICH408I USER(WASPD2 ) GROUP(SYS1 ) NAME(USER1) /u/waspd2/.sh_history CL(FSOBJ
) FID(00000000000000004C03000000190000) INSUFFICIENT AUTHORITY TO OPEN
ACCESS INTENT(RW-) ACCESS ALLOWED(GROUP ---) EFFECTIVE UID(0000003174)
EFFECTIVE GID(0000000000)
```

3. Analyze exception or error message.

If you found a specific message that indicates the problem cause, pursue it and try to solve it. The authorization failure message in Example 11-1 can be resolved by requesting appropriate authorization from the security administrator so that the application can run with sufficient privileges. If you are not sure how to solve the problem, search for potential solutions on the WebSphere support page; see step 14.

If you found a message or exception that is not self-explanatory, refer to Chapter 4, “Exceptions and error messages” on page 41.

4. Check resource definition.

Check in the Administrative Console if your resource is defined. Go to the Resources panel and compare whether the list of resources represents the resource access requirements for the application.

5. Are the resources defined?

Did you find all the resources in the list? If no, proceed with step 6.

If yes, then go to step 7.

6. Define resources.

You must define the application resources in the Administrative Console.

For database access, follow these steps:

- a. Go to **Login** → **Resources** → **JDBC Providers** → **New**. Define the database type, provider type, and implementation type properties as shown in Figure 11-2, which is a sample configuration of a new JDBC provider for DB2. Click **Next**.

Figure 11-2 New JDBC provider for DB2

- b. Specify and confirm the location of the class path and native library path and click **OK**.
- c. In the next panel, select your new DB2 JDBC Provider and click **Data Sources**, then **New**. Figure 11-3 shows a sample of the Data Source properties configuration.

Figure 11-3 Properties for Data Sources

For more detailed information about all available parameters and their use, refer to the “Creating and configuring a JDBC provider and data source” topic at the WebSphere for z/OS Information Center:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

Also check data source names in the resources.xml configuration file.

For messaging services access, if you want to use JMS, you can find more detailed information about all available parameters and their use when you searching “configuring JMS resources for the WebSphere” at the WebSphere for z/OS Information Center

For transaction server access, if you want to access the resources of a transaction server or use any other resource adapter, you can find more information about all available parameters and their use by searching for “installing J2EE Connector resource adapters” at the WebSphere for z/OS Information Center.

7. Is the scope correct?

Go to the Administrative Console and click **Resources** to check whether your resource scope is configured for Cell, Cluster, Node, or Server. Is it correct? If no, then change it; see step 8.

If yes, go to step 9.

8. Change scope.

Change the scope for your resource access. Figure 11-4 shows how to change the scope of JDBC resources. For detailed information about all resources and their use, search for “administrative console scope” at the WebSphere for z/OS Information Center:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

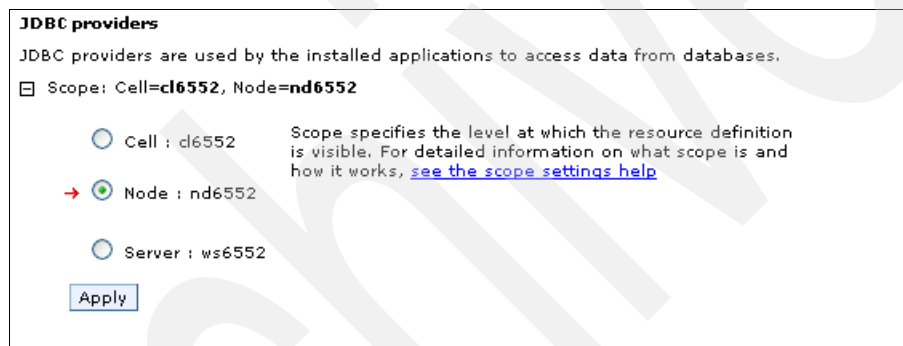


Figure 11-4 Changing database access scope

9. Test connection.

Test the connection to verify that the application can access the data source.

If you are using a JDBC provider, you can test the connection in the Administrative Console (that is, if the connection is available, if the application code can connect to the database, and if the resource is currently available).

To test the connection:

- Go to the Administrative Console and select **Resources** → **JDBC Providers** → **Additional Properties** → **Data Sources**.
- Select a data source.
- Click **Test Connection**. A message is displayed that reports the success or failure of the test.

Figure 11-5 on page 104 shows the panel for testing the connection for JDBC.

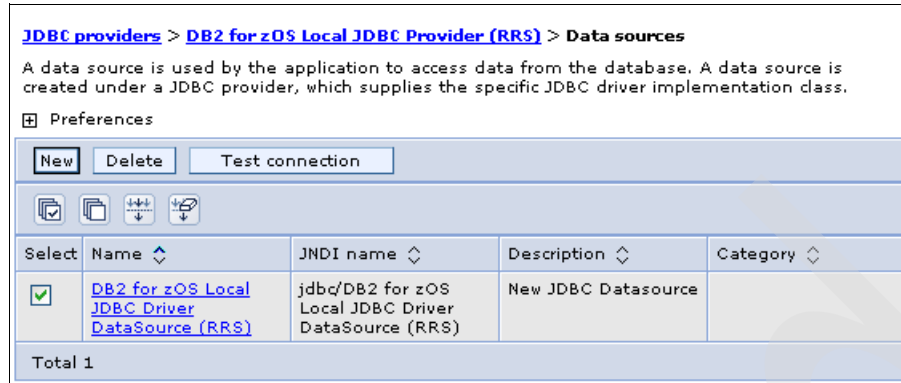


Figure 11-5 Test JDBC connection in Administrative Console

If you are using JMS or another JCA connector, you must check the connection in a different way because the Administrative Console does not provide a utility to test them.

From the symptom you experience and the application logic, you might be able to conclude which specific resource should be accessed. Then check the syslog, job log, and error log for messages that indicate requests for resources and their responses, or regarding resource access failures, such as “permission denied.”

10. JDBC connection test failed?

Did the JDBC connection test fail? If yes, proceed with step 11.

If no, go to step 12.

11. Enable JDBC trace and contact the DB2 administrator.

If the JDBC connection test failed, enable the JDBC trace as described in 20.2, “JDBC trace” on page 244 and at the WebSphere for z/OS Information Center. The JDBC trace output goes to an HFS file that is specified in the JDBC properties file. JDBC trace information shows Java methods, database names, plan names, user names, or connection pools.

Contact the DB2 administrator to discuss what resources you intended to access, what permission is needed, and whether the DB2 logs show any messages indicating resource access problems. Try to solve the problem with the help of the administrator.

12. Contact resource administrators.

If your connection timed out, or you assume that there is a network problem, contact the network administrator. Check the TCP/IP setup, firewalls, and ports. Try to solve the problem with the help of the administrator. Refer to 22.1, “TCP/IP related tools” on page 276, for various tools that might help, such as TCP/IP network packet tracing with Ethereal®.

If your application uses EIS resource adaptors:

- Contact the EIS administrator to verify that the subsystem is up and available because there is no direct way to test the connection to an EIS resource from the WebSphere Administrative Console.
- Go to the Administrative Console and select **Login** → **Resources** → **Resource Adapters**.
- Drill down to the resource name link as you would with the JDBC providers. Verify the configuration properties (given by the administrator or developer of the application) such as the spelling of resource names, the class path information for libraries, and security information.

- d. IMS and CICS also produce their own traces and logs. These subsystems very likely run in their own LPARs. Contact the administrator to get the traces and logs that are required for further analysis or ask them for help.
- e. Check with the security administrator to verify that the user who is configured to access the resource has the required permissions. Specific settings might prevent data access. See *WebSphere Application Server for z/OS V5 and J2EE 1.3 Security Handbook*, SG24-6086.

13. Change logging level.

Enable WebSphere for z/OS trace. If it is enabled, make sure that you have set it to a level that gives you enough information about connections and resource access attempts to research problems. To set the trace or change the level, follow these steps:

- a. In the navigation pane, select **Servers** → **Application Servers**.
- b. Click the name of the server that you want to set the trace for.
- c. Under Troubleshooting, click **Logging and tracing**.
- d. Click **Change Log Detail levels**.
- e. Select a component, package, or group to set a logging level.
- f. Click **Apply**, then **OK**.

Figure 11-6 shows the panel for setting the component, package, and group to a logging level.

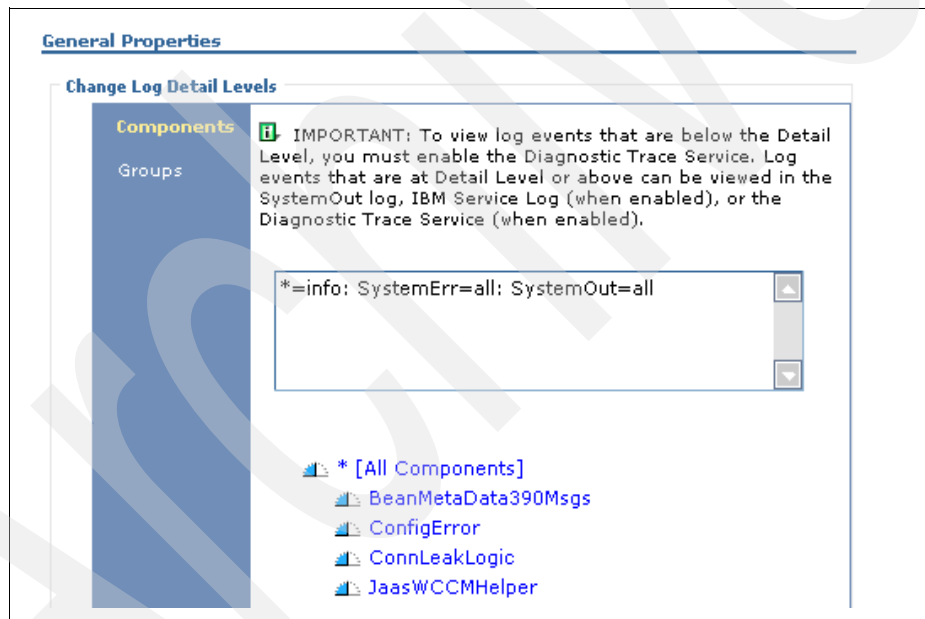


Figure 11-6 WebSphere for z/OS logging detail levels

Note: A logging level set to *all* will produce large traces. Make sure that you have enough space allocated for a large trace and consider the overhead.

See 19.4, “The Java Logging API” on page 227 and the WebSphere for z/OS Information Center for more information about the Java Logging API.

14. Search IBM support data.

After setting the trace, try to recreate the problem, stop the trace (or reduce level), and analyze the trace that is produced for hints relating to the problem. If you found an exception or error message that points to a resource access problem, search IBM support

Web sites and databases for a solution, especially the WebSphere for z/OS support site at:

http://www-306.ibm.com/software/webservers/appserv/zos_os390/support/

From this site, you can access other support sites that are related to WebSphere Application Server, its components, and z/OS. Also, Chapter 3, "Information sources" on page 25 provides many valuable links and resources for solving problems with WebSphere for z/OS.

Tip: When you search problem databases for information or fixes, you might have to alter your search keywords to find a match.

15. Have you identified the problem?

When you searched the trace and support pages, were you able to find the cause of the problem and a solution? If yes, then take the corrective action in step 16.

If no, go to step 17.

16. Take corrective action.

If the problem is related to application code, or you suspect a problem that is related to the application and its access to resources, take the problem description, the application logs, and your traces to the application owner or developer. Clarify which resources should be accessed and whether the application access complies with J2EE and JCA standards. Check that connection properties are defined as intended, and walk through the trace step-by-step to find an indication of what went wrong. Also check the application logs for messages and potential indicators and analyze them with the application developer to determine the exact point in the application where things went wrong.

For more information about resource access, JCA, pitfalls, hints and tips, and best practices, see the developerWorks Web site at:

<http://www.ibm.com/developerworks>

The information that you found using the IBM support data might have provided the following solutions:

- An existing APAR and PTF fix for your problem that is available for you to apply
- Other reports of your symptoms that include a procedure for fixing the problem

In such cases, follow the instructions that are provided or apply the information to your specific problem to solve it. Document the problem and the fixes that you have applied in your system change documentation for your specific WebSphere for z/OS environment for later reference.

17. Locate SVCDump.

You might find more information or hints about the failing resource access in the dump. Usually the name of the SVCDUMP data set is recorded in the syslog. If you are not sure whether a dump was taken or which data set the dump was written to, then search for the word *dump* in the syslog and locate any messages pertaining to the dump.

If there was a problem taking the dump, an IEAxxx type message is issued. For example:

```
IEA911I PARTIAL DUMP ON MVS.01MP.DMP00056 678
IEA043I SVC DUMP REACHED MAXSPACE LIMIT - MAXSPACE=xxxx MEG
```

In that case, you should fix the dump problem first before you attempt to analyze the dump because crucial information might not be written to the dump. Also, ensure that your WebSphere for z/OS servers have the authority to create and write to the dump data sets.

18. Was a dump captured?

Was a dump captured? Were you able to locate the dump? If no, set a SLIP, as described in step 19.

If you located the dump, then prepare to analyze the dump as described in step 21.

19. Set SLIP for SVCDump.

Dumps can be suppressed by the dump analysis and elimination process. In this case, set a SLIP using the z/OS SLIP SET command to capture a dump when the symptom occurs.

Example 11-2 shows a SLIP that was used to capture a dump for an abend EC3 (a started task failure). It uses a wild card for the reason code so that any of the 0413000* abend reason codes that occur are allowed. The ASIDLST is for current, home, primary, and secondary address spaces, and includes other address spaces in the dump in case you are in cross memory with them at the time.

Example 11-2 Example for setting a SLIP

```
SLIP SET,A=SVCD,COMP=EC3,REASON=0413000x,ID=WEC3,MATCHLIM=20,
SDATA=(ALLNUC,CSA,GRSQ,LPA,LSQA,PSA,RGN,SQA,SUM,SWA,TRT),
ASIDLST=(0,H,I,P,S)
```

Refer to *z/OS MVS System Commands*, SA22-7627-11, for a full description and syntax of the SLIP command.

If you are unsure about the most appropriate SLIP, contact IBM support for assistance.

20. Reproduce or wait for reoccurrence.

With the SLIP set, try to reproduce the error. If you cannot reproduce the error, wait for the problem to reoccur with the SLIP in place.

21. Analyze the SVCDUMP.

To analyze the SVCDUMP, invoke IPCS. Several methods can be used to analyze a dump using IPCS. We outlined one approach in “Analyze the SVCDUMP” on page 54 in Chapter 5, “Abend” on page 49. More information about IPCS is in:

- *z/OS V1R6.0 MVS IPCS Commands*, SA22-7594-05
- *z/OS V1R2.0 MVS IPCS User's Guide*, SA22-7596-01

Refer to 20.3, “SVC dumps” on page 247, for more information about how to use SVCDUMP for problem analysis in WebSphere for z/OS.

If you are unable to find a solution, prepare MustGather documentation and contact IBM as described in step 22.

22. Assemble MustGather documentation.

For more information about MustGather, see “MustGather” on page 16. One of the MustGather documents is *MustGather: Read first for WebSphere Application Server for z/OS*. Read it for help with assembling the appropriate documentation.

The minimum information that is needed is:

- Problem description

Include information that is related to when the problem first started to occur, whether it occurs only at certain times, and whether there have been any changes to the system such as maintenance or a new application.

- Version of WebSphere application server and build level

Find this information in the job log of your application server. Search for “build level” to obtain a line that is similar to this:

BBOM0007I CURRENT CB SERVICE LEVEL IS build level cf10515.05 release
WAS601.ZNATV date 04/15/05 12:55:41.

- Operating system version and maintenance (PUT) level
- The job log of the application server in question (including both controller and servant region job logs)
- Any dumps or traces triggered by the problem
- The SVCDUMP triggered by the SLIP

See 2.3, “Before you contact IBM support” on page 15. Then proceed with step 23 to contact IBM Support.

23. Contact IBM support

If you need to contact IBM support, refer to Chapter 2, “Contacting IBM: Information” on page 13, for instructions. Provide the information outlined in the MustGather documentation step.



High CPU utilization

This chapter explains the high CPU utilization symptom. The flow chart and step-by-step descriptions in this chapter can help you analyze the problem and find its cause. We also mention the analysis tools and reference information sources that are related to this symptom.

12.1 What is high CPU utilization?

When you are monitoring the performance of your server environment, you might notice that a certain address space has an increase in CPU usage, which we refer to as “high CPU utilization.” There are many causes for this symptom in a WebSphere for z/OS address space. It might be related to WebSphere for z/OS not performing well, but in most cases it is a sign that this address space is using CPU cycles because it is processing application requests in the server.

It is important to be familiar with CPU utilization information in various stages of an application life cycle. During the development and test phases or after the launch of an application, only a small amount of CPU might be used. However, application requests are likely to increase, as are the number of active applications and their various peak times. You must remember this when you are analyzing CPU utilization. High CPU utilization as such is not a bad thing because you want to utilize as many available resources as you can to justify the costs that are associated with them.

For this chapter, we assumed that you have noticed an unusual increase of CPU utilization and that you need to analyze what caused this increase and potentially remove the problem or explain the background.

Many CPU monitoring tools are available to provide information about how much CPU an address space is using. We used the SDSF Active Users panel (DA) and analyzed the CPU% figures.

In Example 12-1, an ADSF DA display shows that the job WS6422S is consuming approximately 9% of the CPU. We are sure that this is higher than what is normally observed for this server address space.

Example 12-1 SDSF DA display

Display Filter View Print Options Help												

SDSF DA WC42			PAG	0	SIO	1347	CPU	32/ 32	LINE 1-20 (485)			
COMMAND INPUT ==>									SCROLL ==> CSR			
PREFIX=* DEST=(ALL) OWNER=* SORT=CPU%/D SYSNAME=												
NP	JOBNAME	StepName	ProcStep	JobID	Owner	C	Pos	DP	Real	Paging	SIO	CPU%
	WS6422S	WS6422S	BBOSR	STC17374	WSCHTSK	IN	FD	4082	0.00	2925.2		9.20
	MASTER			STC25439	+MASTER+	NS	FF	9097	0.00	0.54		9.70
	SYSVIEW	SYSVIEW	AOFAPPL	STC25503	NETVTASK	NS	FE	8868	0.00	1.63		2.96
	WLM	WLM	IEFPROC			NS	FF	2298	0.00	0.00		2.61
	RMFGAT	RMFGAT	IEFPROC	STC25484	RMFTASK	NS	FE	2278	0.00	0.00		2.61
	WDSCH1BS	WDSCH1BS	BBOSR	STC17374	WDSCHTSK	IN	FD	4049	0.00	1334.6		2.61
	GRS	GRS				NS	FF	6290	0.00	0.00		2.26
	OM2CMS	OM2CMS	CMS	STC25619	OMITASK	NS	FE	2370	0.00	0.00		2.26
	U283614	IKJCEF	QATCCBLP	TSU13860	U283614	IN	FB	1314	0.00	0.00		2.09
	JES2	AAIB	IEFPROC			NS	FE	5141	0.00	0.00		1.74
	NET	NET	NET	STC25479	VTAMTASK	NS	FE	2051	0.00	0.00		3.22
	WDDEMNI	WDDEMNI	BBODAEMNI	STC17278	WDTASK	NS	FE	274	0.00	0.00		0.00

12.2 Symptom flow chart: High CPU utilization

Figure 12-1 on page 111 shows the flow chart for this symptom. Each box has a number that refers to the steps that are described in 12.3, “Diagnosing high CPU utilization” on page 111.

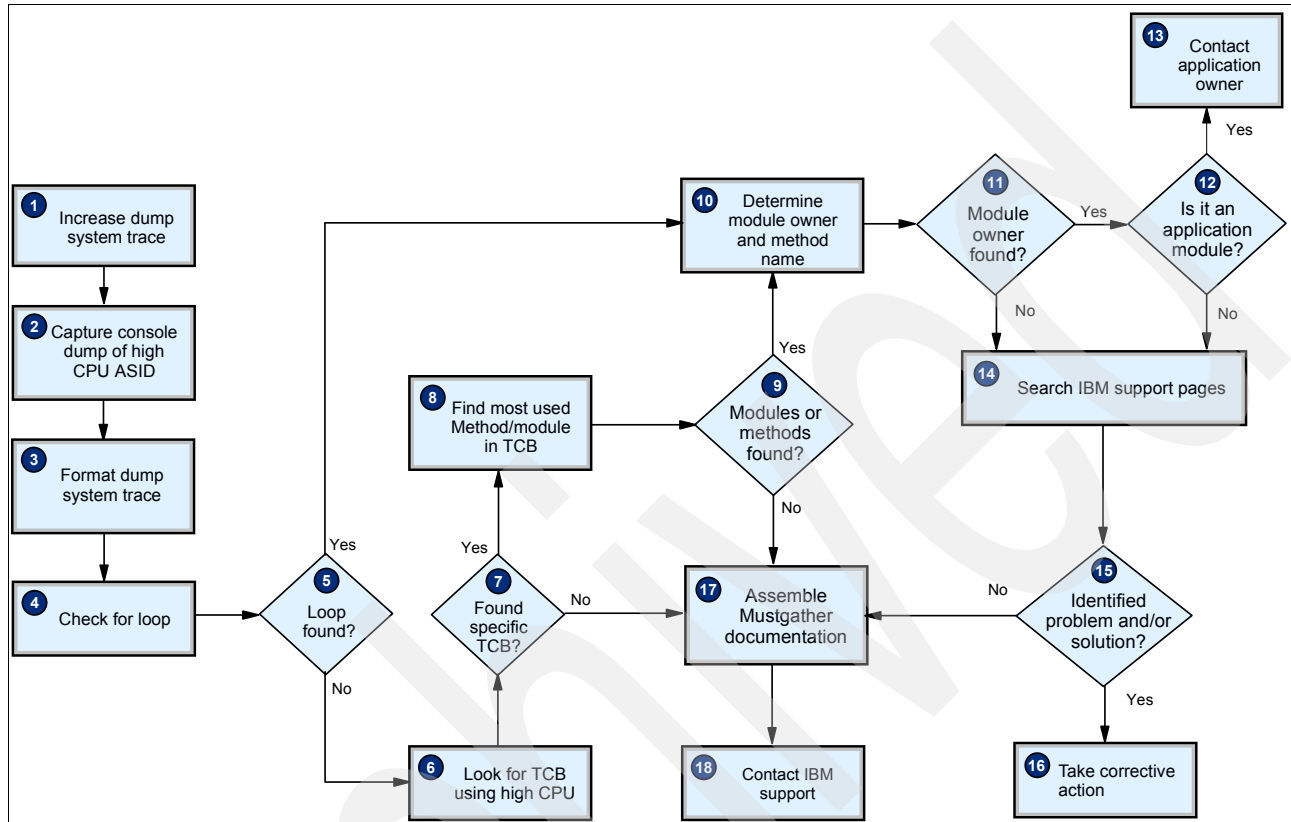


Figure 12-1 Flow chart for symptom: High CPU utilization

12.3 Diagnosing high CPU utilization

Follow these steps to analyze and solve the problem:

1. Increase dump system trace.

To determine whether an address space is using high CPU, some historical data about CPU usage of this address space must be available.

Increase the size of the system trace so that you can collect more trace data before you capture a console dump of the address space. The system trace has more entries and uses a larger time span to analyze the problem.

Issue the z/OS trace command:

```
TRACE ST,999K
```

2. Capture console dump of high CPU ASID.

Using the z/OS dump command, capture a console dump of the WebSphere Application Server address space that is using more CPU than usual. Include the OMVS address space and data spaces because they contain information about USS processes and threads.

Example 12-2 on page 112 shows an example of the DUMP command.

Example 12-2 DUMP command

```
DUMP COMM=(Descriptive name for this Webserver dump)
R rn,SDATA=(CSA,SQA,RGN,TRT,GRSQ,LPA,LSQA,SUM,NUC,PSA),CONT
R rn,JOBNAME=(OMVS,controlregionname,servantregionname),CONT
R rn DSPNAME ('OMVS'.*),END
```

3. Format the dump system trace.

Using the IPCS utility, format the system trace table in the dump with this command:

```
ip systrace time(local)
```

The system trace table shows the activity of the address space at the TCB level. The most recent events are at the bottom.

Example 12-3 shows a sample system trace table. The output has been truncated on the right to fit the page.

Example 12-3 System Trace Table

```
----- SYSTEM TRACE TABLE -----
--
--
PR ASID WU-ADDR- IDENT CD/D PSW----- ADDRESS- UNIQUE-1 UNIQUE-2 UNIQUE-3
UNIQUE-4 UNIQUE-5 UNIQUE-6
03 00FB 009E69C0 DSP 470C0400 80FF6EE8 00000000 40000001 CE6C4B58
03 00FB 009E69C0 PR ... 0 31B0AB86 00FF6EEE
03 00FB 009E69C0 PR ... 0 0663EAAB 31CBBAB4
03 00FB 009E69C0 PC ... 8 0663EAAB 01300
03 00FB 009E69C0 PC ... 0 31B0AB86 0030D
03 00FB 009E69C0 SSRV 128 00000000 CE6C4B58 40000001 00000000
00000000
03 00FB 009C3030 DSP 470C0400 80FF6EE8 00000000 40000001 CE6C3888
03 00FB 009C3030 PR ... 0 31B0AB86 00FF6EEE
03 00FB 009C3030 PR ... 0 0663EAAB 31CBBAB4
03 00FB 009C3030 PC ... 8 0663EAAB 01300
03 00FB 009C3030 PC ... 0 31B0AB86 0030D
03 00FB 009C3030 SSRV 128 00000000 CE6C3888 40000001 00000000
00000000
03 00FB 0099F828 DSP 470C0400 80FF6EE8 00000000 40000001 CE6BC268
03 00FB 0099F828 PR ... 0 31B0AB86 00FF6EEE
03 00FB 0099F828 PR ... 0 0663EAAB 31CBBAB4
03 00FB 0099F828 PC ... 8 0663EAAB 01300
03 00FB 0099F828 PC ... 0 31B0AB86 0030D
03 00FB 0099F828 SSRV 128 00000000 CE6BC268 40000001 00000000
00000000
```

4. Check for loop.

Using the system trace, begin at the bottom of the file with the most recent entries. Look for any recurring or repetitive patterns in the system trace entries with the same PSW addresses. Using these addresses, review them in dump storage with IPCS option 1. Scroll up from the address in storage and locate any eye catchers that identify module names. It might be difficult to determine the module names and relate them to Java methods. With the TCB, you can run a traceback that might help identify what code the TCB is running. Use this command:

```
ip verbx ldata 'tcb(009C31C8) nthreads(*) asid(00fb)'
```


Example 12-4 shows a truncated version of the TCB traceback information and the methods used.

Example 12-4 TCB traceback and methods

TCB(009C31C8) NTHREADS(*) ASID(00FB)

Language Environment Product 04 V01 R6.00

To Display Additional Information:

IP VERBX LEDATA 'CAA(5ACA4300)DSA(5ACA65B8) ALL'

Information for enclave main

Information for thread 322D057000000056

PCB Address: 31B0D080

TCB Address: 009C31C8

Registers and PSW:

```
GPR0..... 00000001  GPR1..... 7F3E3F90  GPR2..... 36D3B164  GPR3..... 5ACA34A0
GPR4..... 00008000  GPR5..... 31B0D4F8  GPR6..... 36D3AA50  GPR7..... 36D3B004
GPR8..... 7F3E3F90  GPR9..... 00000000  GPR10.... 36D3B000  GPR11.... 085DEFF0
GPR12.... 5ACA4300  GPR13.... 5ACA65B8  GPR14.... 5AF44600  GPR15.... 809CE608
PSW..... 478D0400 80000000 00000000 085DC5B8
```

Traceback:

DSA Addr	Program Unit	PU Addr	PU Offset	Entry	E Addr	E Offset
5ACA65B8	CEEOPCW	085DAC10	+000019A8	CEEOPCW	085DAC10	+000019A8
5ACA6500		08325060	+00000080	pthread_cond_wait	08325060	+00000080
5ACA6430		7C902068	+00000192	condWait	7C902068	+00000192
5ACA6370		7C914700	+0000027E	sysMonitorEnter	7C914700	+0000027E
5ACA62C0		7CD82DC0	+00000108	xmIsThreadInterrupted	7CD82DC0	+00000108
5ACA6208		7CAE7760	+000000F0	JVM_IsInterrupted	7CAE7760	+000000F0
5ACA6130		34386444	+000000DE	java/lang/Thread.isInterrupted(Z)Z	34386444	+000000DE
5ACA6048		5F250754	+00000086	EDU/oswego/cs/dl/util/concurrent/W	5F250754	+00000086
5ACA5F48		624687BC	+000000CE	org/grnds/foundation/cache/GrndsCa	624687BC	+000000CE
5ACA5D08		6118BB94	+000001DA	org/grnds/foundation/cache/GrndsCa	6118BB94	+000001DA
5ACA5C30		7CD3A610	+00000534	mmipSelectInvokeJavaMethod	7CD3A610	+00000534
5ACA5B60		7CD3A090	+00000AB4	INVOKDMY	7CD3A090	+00000AB4
5ACA5A90		7CD39550	+00000096	EXECJAVA	7CD39550	+00000096
5ACA59E0		7CD5A608	+000000EC	mmipExecuteJava	7CD5A608	+000000EC
5ACA58B0		7CD609C0	+000004D0	xeRunDynamicMethod	7CD609C0	+000004D0
5ACA5800		7CADBC08	+000000E0	threadRTO	7CADBC08	+000000E0
5ACA5608		7CD81D48	+00000362	xmExecuteThread		

5ACA5560	7CD6CB28	+0000006C	threadStart	7CD81D48	+00000362
5ACA53F8	7C931638	+00000A38	ThreadUtils_Shell	7CD6CB28	+0000006C
				7C931638	+00000A38
5ACA5340	081C90D8	+0000001A	@GETFN	081C9030	+000000C2
7F1A4EE0	CEEOPCMM	00C68DA8	+0000090E	CEEOPCMM	00C68DA8
					+0000090E

5. Have you found a loop?

Have you found a loop in your output? If no, proceed with step 6.

If yes, go to step 10.

6. Look for TCB with high CPU utilization.

Use the system trace entries, starting from the bottom of the trace with the most recent entries, to look for the TCB using high CPU. Identify which TCBs have the most entries. Determine which TCBs are using more CPU time.

7. Have you found a specific TCB?

Did you find specific TCBs causing high CPU usage? If yes, then proceed with step 8.

If no, IBM support has specific utilities and tools that they can run against the system trace to extract TCB and time statistics. Go to step 17 (to prepare for contacting IBM).

8. Find most used method or module in TCB.

In the system trace for the TCB, look at the column IDENT and focus on the DSP and SRB trace entries. A DSP trace entry represents dispatch of a task. An SRB trace entry represents the initial dispatch of a service request. Note the PSW addresses.

Browse the PSW addresses of the DSP and SRB entries in dump storage using IPCS option 1. Locate the module eye catchers. Using the data that has been collected, determine which modules show the most activity.

9. Were the modules or a method found?

Were you able to identify specific modules and methods with the most activity (causing higher CPU utilization)? If yes, determine the owner, and proceed with step 10.

If no, go to step 17 to prepare for contacting IBM support.

10. Determine module owner and method name.

From the name of the module or method, you should be able to identify the owner of the code. Specific prefixes indicate specific owners.

IBM modules have a prefix that identifies them to a component (for example, BBO modules indicate WebSphere for z/OS code that is owned by IBM). These are documented in Chapter 1 in *z/OS V1R4.0 MVS Diagnosis: Reference GA22-7588-03*.

See A.2, "System and component message table" on page 315, to identify which other IBM products (such as z/OS components or subsystems) might have created your particular error message.

11. Is a module identified?

Have you identified the module? If yes, proceed with step 12.

If no, go to step 14.

12. Is it an application module?

The module owner can help determine why this module uses more CPU than usual. Is the module code from an application or a non-IBM product? If yes, proceed with step 13.

If no, go to step 14.

13. Contact application owner.

Contact the owner of the application and identify whether the code has any known problems. Identify documentation that they might require to investigate the problem. For example:

- Traces
- Response and throughput of JSP™, servlets, and EJBs (use Tivoli Performance Viewer)
- Time stamps in application

Using the module and method names, consult the owner of the code.

Tip: Even if the module is not from IBM, sometimes you might find hints and tips related to other products and application in context with WebSphere for z/OS when you search the WebSphere Information Center or various other sources outlined in Chapter 3, “Information sources” on page 25.

14. Search IBM support pages.

If the module is an IBM module or method or you are not sure who owns the module, search the IBM support pages and determine the owner and check whether the code has known problems.

To search the IBM support Web sites and databases, specifically the WebSphere for z/OS support site, go to:

http://www-306.ibm.com/software/webservers/appserv/zos_os390/support/

From this site, you can access other support sites that are related to WebSphere Application Server, its components, and z/OS.

You can also consult the specific product manuals or search the IBM Software support Web site at:

<http://www-950.ibm.com/search/SupportSearchWeb/SupportSearch?pageCode=SPS>

See Chapter 3, “Information sources” on page 25, which provides many valuable links and resources for solving problems in WebSphere for z/OS.

Tip: When you are searching problem databases for information or fixes that are related to the module or method, consider the format of the search criteria. You might have to alter your search keyword to find a match.

If you have exhausted all resources and no apparent fix is found for your problem, proceed with step 15 to prepare to contact IBM.

15. Have you identified the problem and solution?

After searching the support pages, did you find the cause of the problem and a solution? If yes, then take the corrective action; see step 16.

If no, prepare to contact IBM support; see step 17.

16. Take corrective action.

The information that you found using the IBM support data might provide the following solutions:

- An existing APAR and PTF fix for your problem that is available for you to apply
- Other reports of your symptoms that include a procedure for fixing the problem

In such cases, follow the instructions that are provided or apply the information to your specific problem to solve it.

Document the problem and the fixes that you have applied in your system change documentation for your specific WebSphere for z/OS environment for later reference.

17. Assemble MustGather documentation.

Read *MustGather: Read first for WebSphere Application Server for z/OS*, for help with assembling the appropriate documentation. Although it is for WebSphere for z/OS V5, the document: *Mustgather: High CPU causing Hang or Loop running V5 for z/OS*, might also help you collect the right documentation.

For more information about MustGather, see “MustGather” on page 16.

The minimum information necessary is:

- Problem description

Include information that is related to when the problem first started to occur, whether it occurs only at certain times, or what changes have been applied, such as maintenance or a new application.

- Version of WebSphere Application Server and build level

Find this information in the job log of your application server. Search for “build level,” to obtain a line similar to this:

```
BBOM0007I CURRENT CB SERVICE LEVEL IS build level cf10515.05 release  
WAS601.ZNATV date 04/15/05 12:55:41.
```

- Operating system version and maintenance (PUT) level

- The job log of the application server in question (including both controller and servant region job logs)

- Any dumps or traces triggered by the problem

- Information obtained from performance reports

See 2.3, “Before you contact IBM support” on page 15. Then proceed with step 18 to contact IBM support.

18. Contact IBM support.

If you need to contact IBM support, refer to Chapter 2, “Contacting IBM: Information” on page 13, for instructions. Provide the information that is outlined in the previous step.



WebSphere for z/OS performance analysis

This chapter is a general introduction to key performance factors in a WebSphere for z/OS production environment. Along with our general recommendations, we describe a performance troubleshooting approach. We use examples to explain how to pinpoint the source of the problem and provide interpretation of data and rules of thumb.

13.1 Performance terminology

The performance of a server can be defined as a measure of how well it carries out a task. For a computer system or application, we usually take this to mean how fast it carries out the task, but it can also include a measure of how many tasks it can complete in a given time.

The following sections define major terms that are applied to analyzing performance.

13.1.1 Response time

According to the *IBM Dictionary of Computing*, which cites *International Organization for Standardization Information Technology Vocabulary* as the source, response time is: “The elapsed time between the end of an inquiry or demand on a computer system and the beginning of a response; for example, the length of the time between an indication of the end of an inquiry and the display of the first character of the response at a user terminal.”

A Web user’s view might differ. Consider the case where a servlet or JSP returns an HTML page that includes many GIFs and other, similar files. The user is likely to view response time as the time between clicking the mouse until the resulting page is completely rendered in the browser. The previous definition would stop the response time clock when the first byte of HTML is received by the browser, not when the page has been completely rendered.

If you use an HTTP server that is external to WebSphere to serve your static content, it might be impossible (or, at least, very difficult) to find a means of measuring response time in the same manner that a Web user perceives it. You would have to be able to measure the combined time for serving all the elements of the generated HTML page. If you use WebSphere to serve your static content, the servlet/JSP request and each GIF (or other similar files) that is referenced in the HTML would appear as separate items that have to be combined in some manner to form the complete response time. Luckily, this is rarely a problem because there are a number of points in the network and browser where static content is cached.

IBM Resource Measurement Facility (RMF™) reports for WebSphere for z/OS measure the time between work being queued by a controller region and that piece of work being completed in the servant region. We use this definition of response time in this chapter.

13.1.2 Throughput

Throughput is a measure of the amount of work that is going through a system in a given time. Typically, this can be measured as the number of transactions per second. As with response time, our measurement of throughput with WebSphere for z/OS is measured based on the work that is completed by the servant region.

13.1.3 Transaction

A strict definition of transaction is “logical unit of work.” When one transfers money from one account to another, it must be removed from the first account and then added to the second. The transaction includes both of these processes and they must both complete successfully for the transaction to be considered complete. A mechanism is also required to ensure that if one of the processes fails, the other is either not attempted or is also undone.

A Web browser user purchasing a book from a Web site might consider the complete process of selecting the book, entering payment and delivery details, and then finalizing the purchase as a single transaction.

WebSphere considers each incoming request as a transaction. Each of the requests to WebSphere generated by the customer as they go through the process of buying a book is treated as a separate transaction. Our discussion in this chapter uses the term transaction as viewed by z/OS Workload Manager and reported by RMF.

13.1.4 Hit rate

Hit rate is often used as a measure of activity on a Web site. A hit is the retrieval of any single item from a Web server. Therefore, a Web page with four graphic items actually counts as five hits: one for the HTML page and one for each of the graphic items. Hit rate is the number of hits in a given time. Although this measures all the interactions between the user and browser, it tends to hide the more valuable measure of the number of pages being accessed.

13.1.5 Page view rate

A more valuable measure than hit rate, page view rate counts complete pages that are retrieved in a given time rather than all the individual elements.

Important: The above definitions should be understood when interpreting WebSphere transaction rate from an RMF workload activity report. For WebSphere for z/OS, RMF views each request as a transaction, whether it is a call to a J2EE application or a request to a static page element.

If WebSphere for z/OS is serving static pages, the transaction rate reported by RMF will in fact be closer to a measure of the hit rate.

If static content is served from another source, for example, a WebSphere Edge server front end, and requests that are issued to the back-end application server are mostly for J2EE applications, then the value reported by RMF will be closer to the resulting page view rate.

13.1.6 Number of clients and think time

The number of clients is the number of users that are connected to the Web site. However, in contrast with existing applications, there is no direct relation between the number of clients that are connected and the load on the Web server because of the heterogeneous nature of Web applications. In a traditional CICS or IMS application, users are usually logged on and working almost continuously.

In a Web application such as one used to buy a book, there is more browsing while users evaluate the information that has been returned. This is *think time*. While the users are thinking, they are still effectively connected to the site but they are not driving work in WebSphere (although there might still be a session object from their previous interaction). Thus, in an application that has a long think time, there might be a large number of concurrent users, also called clients, but a low transaction rate in WebSphere Application Server.

13.1.7 Resource

A resource is any item that can be used in the transaction process. This can be a physical resource (for example, CPU or memory) or a logical resource (for example, JDBC connection or a queue in WLM). When a WebSphere transaction accesses data in DB2 or CICS, it might also be convenient to refer to DB2 or CICS as a resource.

For a transaction to complete, it must be able to access all the resources it requires. For a transaction to perform well, there must be enough of these resources available and they need to be available quickly enough. How much is enough? How quick is quick? There is no firm answer. It depends on your business requirements.

13.2 Managing performance of WebSphere transactions

WebSphere for z/OS uses the z/OS Workload Management (WLM) function to start and manage servers in response to workload activity. It is a WebSphere for z/OS requirement that WLM run in goal mode. If your z/OS system is running WLM in compatibility mode, follow the instructions in *z/OS Planning: Workload Management*, SA22-7602, to implement the goal mode before proceeding with WebSphere for z/OS customization.

Each J2EE application server in a WebSphere for z/OS cell uses WLM to start servants as WLM application environments. Thus, each application server must be associated with a WLM application environment name. The **cluster transition name** in the WebSphere for z/OS configuration is used as the WLM application environment name.

WebSphere for z/OS makes use of dynamic WLM application environments when they are available. (The WLM service that added dynamic application environments is a prerequisite for WebSphere for z/OS.) Therefore, you do not need to use the WLM ISPF panels to manually create application environments for WebSphere for z/OS.

The response time and throughput of WebSphere transactions are managed based on their assigned service class, associated performance objectives, and availability of system resources, which we explain in more detail in the following sections.

13.2.1 Managing the number of servant regions

Each WebSphere application server can have one or multiple servant regions per server instance based upon the settings that are defined in the WLM application environment.

How many servant regions are created depends on the WLM determination of:

- ▶ How the work is meeting the performance goals
- ▶ The importance of the work compared to other work in the system
- ▶ The availability of system resources that are needed to satisfy those objectives
- ▶ Whether starting more address spaces can help achieve the objectives (see Figure 13-1 on page 121)

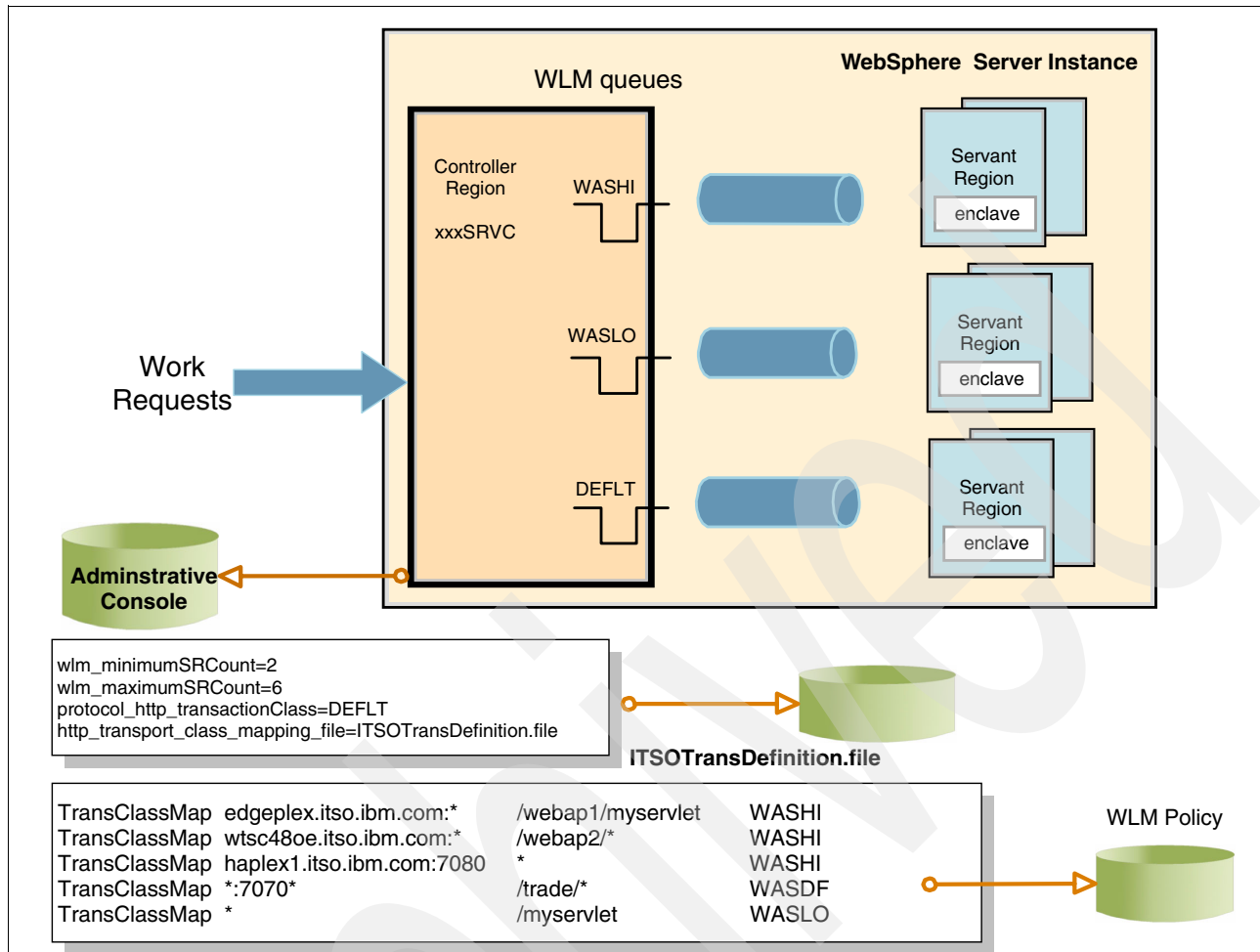


Figure 13-1 WebSphere workload definition with Workload Manager

By default, the minimum number of regions for J2EE servers is one; there is no default maximum. You can override the maximum and minimum number of servant regions that WLM will start with three parameters in the Administrative Console:

- ▶ Minimum number of instances: `wlm_minimumSRCount`

This parameter is used to start up a basic number of servant regions before the day's work arrives. This can reduce the time that is spent waiting for WLM to determine that more servant regions are needed. To keep work from coming in through the protocol handler before servant regions are ready, use:

`protocol_accept_http_work_after_min_srs=nn`

- ▶ Maximum number of instances: `wlm_maximumSRCount`

This parameter is used to cap the number of address spaces started by WLM if you determine that excessive servant regions might contribute to service degradation (for example, if real storage is limited).

- ▶ Multiple instances enabled

This parameter is used to limit application server to one servant region. Even if minimum and maximum numbers of servant regions are defined as > 1 , the ports will not be open. Ensure that you have selected the option to allow more servant regions.

Transactions that are received by the application server controller region are passed to servant regions through a set of WLM queues. The number of queues is determined by the number of service classes that are defined, and one servant region only serves one service class at a given time. To ensure that you do not limit the parallelism of execution under full load, wlm_maximumSRCount should be set so that, at minimum, it is as large as the number of service classes defined.

A wlm_maximumSRCount setting that is too low creates a situation where fewer servers are available than WLM queues. The result might be a queue bottleneck under full load conditions, because WLM can be restricted from starting enough servant regions to handle the workload. As a consequence, the system might experience queuing delays in the WLM queues resulting in transactions with elongated response time or timeout errors.

For more information about classifying z/OS workload, see the WebSphere Information Center Web site at:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

13.2.2 Managing the number of JVM threads

To manage the number of JVM threads:

1. In the Administrative Console, select **Servers** → **Application Servers** → **server_name** → **ORB Service** → **Advanced Settings** → **Workload Profile**.
2. Experiment with the following parameters and the number of servant regions to optimize your performance:
 - ISOLATE: 1 thread
 - NORMAL: 3 threads
 - CPUBOUND: Number of CPs, minimum of 3
 - IOBOUND: Number of CPs x 3, minimum=5, maximum=30
 - LONGWAIT: 40
3. Allow for increased concurrency. WebSphere for z/OS does not need threads as placeholders for work because it uses WLM queues. Plan for number of in and ready threads to be 2 to 3 times the number of CPs. Remember, too many threads in a JVM create interference and more frequent garbage collection.

13.2.3 Classifying servant region enclaves (WebSphere transactions)

Each WebSphere transaction is dispatched as a WLM enclave and is managed in the servant region according to the service class that is assigned based on the CB service classification rules.

This WLM classification is used for WebSphere applications that run in the servant region as part of the dispatched enclave. The classification can be based on the following criteria:

- ▶ Server name (CN): Recommended (cluster transition name=applenv name)
- ▶ Server instance name (SI): Not recommended (consider instances share work)
- ▶ User ID assigned to the transaction (UI): Not recommended (unusual attempt)
- ▶ Transaction class (TN): Recommended (many assignment methods)

Figure 13-2 on page 123 is an example of WLM classification by server name (CN) and transaction classes (TN).

```

Subsystem-Type  Xref  Notes  Options  Help
-----
                Modify Rules for the Subsystem Type                Row 1 to 10 of 10
Command ==> _____ SCROLL ==> PAGE

Subsystem Type . : CB                Fold qualifier names?  Y  (Y or N)
Description . . . WebSphere App Server

Action codes:  A=After      C=Copy      M=Move      I=Insert rule
               B=Before     D=Delete row  R=Repeat    IS=Insert Sub-rule
                                   More ==>

-----Qualifier-----
Action  Type      Name      Start

_____ 1  CN      FMISRV*  _____
_____ 1  CN      FMESRV*  _____
_____ 1  CN      OMESRV*  _____
_____ 1  CN      OMTSRV*  _____
_____ 1  CN      INTSRV*  _____
_____ 1  CN      INESRV*  _____
_____ 1  TN      WASLO    _____
_____ 1  TN      WASDF    _____
_____ 1  TN      WASHI    _____

-----Class-----
Service      Report
DEFAULTS: WASDF      OTHER
_____ WASDF
_____ WASDF
_____ WASLO      WASE
_____ WASLO      WASE
_____ WASLO      WASE
_____ WASLO      WASE
_____ WASLO      WASE
_____ WASLO      WASE
_____ WASDF      WASE
_____ WASDF      WASE
_____ WASHI      WASE

***** BOTTOM OF DATA *****
F1=Help  F2=Split  F3=Exit  F4=Return  F7=Up  F8=Down  F9=Swap
F10=Left  F11=Right  F12=Cancel

```

Figure 13-2 WLM definitions for servers and transaction classes in CB subsystem

You can assign a default transaction class for the server or server instance in the *protocol_http_transactionClass* or *protocol_https_transactionClass* environmental variables (see Figure 13-1 on page 121).

For more information, search for “workload classification file” at the WebSphere Information Center:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

You can use the virtual host name, port number, or URI template to map the HTTP request to a transaction class with a filtering file that is specified in the *http_transport_class_mapping_file* variable (see the mapping definitions at the bottom of Figure 13-1 on page 121).

The authors recommend that you define WebSphere transaction service classes using a percentage response time objective as illustrated in Figure 13-3 on page 124. The response time goal for the Service Class WASHI is: 90% of all transactions finish within 0.2 seconds.

Modify a Service Class		Row 1 to 2 of 2
Command ==> _____		
Service Class Name	: WASHI	
Description	LSA510 WAS 200MS RT	
Workload Name	WAS	(name or ?)
Base Resource Group	_____	(name or ?)
Cpu Critical	NO	(YES or NO)
Specify BASE GOAL information. Action Codes: I=Insert new period, E=Edit period, D=Delete period.		
<div> <div>---Period---</div> <div>-----Goal-----</div> </div>		
Action #	Duration	Imp. Description
1	1	90% complete within 00:00:00.200
***** Bottom of data *****		
F1=Help	F2=Split	F3=Exit
F4=Return	F7=Up	F8=Down
F9=Swap	F10=Menu Bar	F12=Cancel

Figure 13-3 WLM definition of Service Class WASHI

A response time objective is usually consistent with the business requirement of a Web application. The response time value can be adjusted depending on the type of application.

Tip: Avoid multi-period goals, because second and subsequent periods are not aggressively managed.

Response time goals are better than velocity goals in a true production environment because velocity goals must be recalibrated with environmental changes, such as CPU and workload.

This option automatically generates response time distribution information that is reported through an RMF report (see “Response time distribution” on page 292). This option is useful when you must troubleshoot response time issues.

For other methods for assigning transaction classes to incoming work requests, see *Performance Engineering & Tuning for WebSphere V5 and V6 on z/OS*, PRS804, at:

<http://www.ibm.com/support/techdocs>

13.2.4 Classifying servant regions

The authors recommend that you define a report class for the server address space activity so that you can monitor the activity in the servant region for service tasks such as garbage collection and timer management. This WLM classification is used for tasks that run in the servant region under the control of the step task and not as part of the enclave.

Classify the WebSphere servant regions with a service goal that is high enough to allow them to compete effectively with other workloads and so that they can be given control quickly when WLM determines they are needed. In other words, tasks with the qualifier INESRV* are classified with service class VEL85, which could be defined as Importance=1 and Velocity=85

(Figure 13-4). Again, the authors recommend defining a reporting class to isolate the activity into a specific workload report (for example, WAS2 for INESRV*).

```

Subsystem-Type  Xref  Notes  Options  Help
-----
                                Modify Rules for the Subsystem Type      Row 1 to 16 of 61
Command ==> _____ SCROLL ==> PAGE

Subsystem Type . : STC          Fold qualifier names?  Y  (Y or N)
Description . . . Use Modify to enter YOUR rules

Action codes:  A=After      C=Copy      M=Move      I=Insert rule
               B=Before     D=Delete row R=Repeat    IS=Insert Sub-rule
                                           More ==>

-----Qualifier-----
Action  Type      Name      Start      Service      Report
-----
          1  TN      HWS710*  _____  DEFAULTS:  SYSSTC  OTHER
          1  TN      FMESRVS*  _____  IMSCNTL    WAS1
          1  TN      FMISRV*  _____  VEL80      WASS
          1  TN      INESRV*  _____  VEL80      WASS
          1  TN      WSESRV*  _____  VEL80      WASS
          1  TN      OMESRV*  _____  VEL80      WASS
          1  TN      HAO*     _____  _____   WAS
          1  TN      FMESRV*  _____  VEL85      WAS1
          1  TN      INESRV*  _____  VEL85      WAS2
          1  TN      OMESRV*  _____  VEL85      WAS3
F1=Help  F2=Split  F3=Exit  F4=Return  F7=Up      F8=Down  F9=Swap
F10=Left F11=Right F12=Cancel

```

Figure 13-4 WLM definition of the servant regions, STC subsystem

13.2.5 Classifying controller regions

A certain amount of processing occurs in the WebSphere application controller regions to receive work into the system, manage the HTTP Transport Handler, classify the work, and so forth. Therefore, controller regions should also be classified in SYSSTC or a high velocity goal.

For more information, see *Workload management (WLM) tuning tips for z/OS* at the Information Center:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

13.2.6 Special considerations for HTTP requests over multiple servants

WebSphere uses a “hot server” strategy to route HTTP requests. This means that HTTP requests are routed to servant regions that have recently dispatched work with threads available. “Hot servers” have pages in memory, application methods, and caches full of data. HTTP requests with session affinity are routed to the servant regions with the associated session objects.

While this can improve performance considerably, it also bears some risks. “Hot” servant regions can get overloaded with work, while others are idle. Garbage collection or the loss of a servant region can impact many sessions.

To distribute HTTP requests evenly across servant regions:

1. For the desired servers in the Environment variables in the Administrative Console, specify:
`wlm_stateful_session_placement_on=1`
2. Optimize the minimum and maximum number of servant regions.
3. Consider eliminating transaction class mapping and minimize the number of different service classes for these servers.

To determine whether the classification scheme that was implemented is classifying work as expected, use the z/OS operator command to display WLM classification of work requests:

```
F <server>,DISPLAY,WORK,CLINFO
```

See 18.2, “z/OS MODIFY commands” on page 196 for more information about z/OS commands.

13.3 Introduction to performance analysis and management

Performance management typically includes:

- ▶ Performance monitoring: Seeing that everything is running smoothly
- ▶ Performance analysis: Getting to the root of problems
- ▶ System tuning: Ensuring the best usage of resources
- ▶ Capacity planning: Ensuring that you have enough resources

We only address the first two items for a WebSphere for z/OS environment in this book.

When you are managing application performance, you must be able to identify a performance problem. Therefore, you must also set your performance expectations so that you can determine if a problem is only perceived as a performance problem or if it really represents one before you start analyzing what caused the problem.

13.3.1 Setting your performance expectations

How many transactions per second should you expect from a given WebSphere for z/OS implementation? Like all applications, WebSphere for z/OS applications use system resources that you can rely on when setting expectations. These include:

- ▶ Monitoring and extrapolation

If you already have a running application, by taking appropriate measurements on a regular basis, you can understand how your application performs in normal operation. Any deviation from this baseline can represent a performance problem. Be careful when projecting forward from monitored data. The numbers might not scale linearly, especially if you are already approaching a limit. Carefully review logical resources and physical resources, such as CPU.

- ▶ Experience

Be careful when generalizing. Different applications behave differently. Benchmarks generally only tell you how well the benchmark environment was prepared; they do not guarantee how your application behaves in your production environment.

- ▶ Unrelated experience in other environments, systems, or subsystems

Do not compare the behavior of other subsystems, such as CICS, with WebSphere for z/OS. Other subsystems serve different types of users and requests. Although

WebSphere for z/OS is J2EE compliant, WebSphere for z/OS applications do not necessarily behave in the same way as those for WebSphere on distributed platforms because the underlying system functions are used differently.

- ▶ **Business requirements**

Unless they are backed by data that confirms that these requirements can be met in your environment, in reality this might be little more than a statement of intent.

- ▶ **Load testing**

Using workload simulation tools, such as WebSphere Studio Workload Simulator (see 22.5, “Stress test tools” on page 300), you can evaluate how an application will behave in your environment as long as you can recreate a testing environment that matches the projected production environment.

- ▶ **Capacity planning**

Although there is very little information published on this topic, your IBM representative or authorized Business Partner has access to Technical Support to do pre-sales sizing estimates for your WebSphere for z/OS applications.

Note: Keep in mind that, although it is important to set expectations, some of the methods might lead to unreliable and unrealistic expectations.

13.3.2 What is a performance problem and how do you manage it?

There are many views on what constitutes a performance problem. Most of them revolve around unacceptably slow response times or high resource usage, which we can collectively refer to as pain. The need for performance investigation and analysis is detected by system indicators or users complaining about slow response. The major indicators of performance problems are:

- ▶ Complaints from users
- ▶ Service level objectives not being met
- ▶ Alerts from monitoring tools
- ▶ Unexpected changes in reported usage
- ▶ System resource indicators (paging rates, direct access storage device (DASD) response)
- ▶ Expected system throughput not being attained

Most of these indications assume that some degree of monitoring is in place. Without monitoring, it is impossible to make objective judgements when you are comparing current performance to past performance or trying to determine what is normal for a given application. It is also impossible to verify a user's complaint objectively without knowing what is really happening in your system.

Ultimately, you must decide for yourself whether a given situation is a problem that is worth pursuing. This decision is based on your own experience, knowledge of your system, and sometimes politics. For the following discussions, we have assumed that you are trying to relieve some sort of numerically quantifiable pain in your system, that is, your expectations for an application in the service region are not being met for:

- ▶ Response time measured in milliseconds per transaction
- ▶ Throughput measured in transactions completed per second

This means that there is a potential performance problem that needs to be solved. Generally, a performance problem is the result of a workload that is not getting the resources it needs to complete in time. Or, less commonly, the resource is obtained but is not fast enough to provide the desired response time.

A common cause of performance problems is having several address spaces or threads (or tasks) compete for the same resource. This could be a hardware resource or a serially usable software resource.

Most problems revolve around unacceptably high response times or resource usage. However, the definition of “unacceptably high” varies from one installation to another. For z/OS, the peaks and troughs of other workloads in the same system image impact the WebSphere environment and vice versa.

The business might have to prioritize other workloads on the image at some point in time, such as year-end batch processing, even though this can be detrimental to the performance of WebSphere for z/OS applications.

Note: The tools and techniques in this chapter can help you identify where your resources are being consumed and why and where your application is experiencing delays. They cannot tell you whether such answers are applicable to your situation. Ultimately, this is a business decision.

13.3.3 What to do about a performance problem

How and whether you apply a solution ultimately depends on business priorities. If a proposed solution to a WebSphere performance problem requires taking resources from another application, you must determine whether this is a price worth paying. If the recommended solution involves extensive application recoding, the cost might not be justifiable if the application has a short life expectancy.

When you are considering cost effectiveness, do not forget the cost of users abandoning your site because it is too slow. They can take their purchases elsewhere and might never try your site again. You might never be aware of the business that you are losing.

You should consider the following factors when you face a performance problem:

- ▶ Make sure your performance expectations are realistic.
Try to understand what can actually be achieved with your application given your hardware and software configuration. As discussed in 13.3.1, “Setting your performance expectations” on page 126, this is actually a difficult question to answer. The result of the problem determination process might be to reset your expectations.
- ▶ Make sure that you have not caused the problem.
Make sure that you follow published performance configuration guidelines. We recommend that you run a G5 server or later and have at least 1 GB or more of real storage to avoid Java performance issues. For more complex applications, you might need more. The definition of “more complex” is rather vague, so it might be best to plan for 2 GB from the outset.

WebSphere for z/OS tuning recommendations can be found at the WebSphere Information Center by searching for “tuning performance”:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

For more tuning tips, see *Performance Engineering & Tuning for WebSphere V5 and V6 on z/OS*, PRS804, at:

<http://www.ibm.com/support/techdocs>

Check the IBM HTTP server recommendations. If you are performance conscious, you are probably using the HTTP Transport Handler that runs in the WebSphere controller region. However, if you are also using the IBM HTTP Server, you should also check installation

recommendations in the IBM HTTP Server manual, available for download under the “Features and Elements” list of your particular z/OS version, at:

<http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/>

On Demand Business is a fast-moving world. Consequently, there is a seemingly never-ending supply of maintenance. Keeping current with maintenance is more important than for more traditional workloads and often brings improved performance.

Check for WebSphere performance information in APARs. The latest information can be found at:

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg27006970>

Attention: Never attempt to tune an application that runs smoothly and without performance problems. Otherwise, you might limit resources, and that can lead to performance problems.

Always tune your applications in test systems first to analyze the impact and consequences. Only tune one variable at a time and document the exact environment, the tuned variable, and the impact that was experienced for later reference.

- Fix the problem, if you can.

Generally, a performance problem is the result of the workload not getting the physical or logical resources that are necessary for it to complete in a timely manner, so the solution is to make more resources available to the application. You can do this by:

- Buying more

If there is no other means for making your application performance meet your expectations, add more resources. At times, this might also be more cost effective than recoding a badly-written application. Beware of induced costs.

- Stealing it

Take it from a less important application. The price that you pay is lower service to the application from which the resources are stolen.

- Using less

For example, fixing badly written code, revising poorly performing SQL queries, or adding appropriate indexes to improve fetches from databases might involve labor costs, but these costs might still be lower than buying new hardware.

- Live with the problem.

If none of the above options is technically or financially possible, you must change your expectations. At least you know why the performance is not meeting your previous expectations. Your users might be disappointed by the answer, but at least you can give them facts and convince them that the situation is understood and under control. Changing the perception might be an important factor in user satisfaction.

13.4 Diagnosing performance problems

This section describes a general approach to troubleshooting a WebSphere application performance problem in a production environment. The intention is to identify the cause of the problem. The actual solution depends on the problem itself.

Performance analysis is an iterative process, and you should be prepared to go through the process a number of times. You should include all components of the WebSphere for z/OS

environment (Figure 13-5) when you are analyzing potential problem areas for your application performance. The correction that you apply as you try to remove the bottleneck might not produce the improvements that you are looking for, or might just move you from one problem to another further down the line.

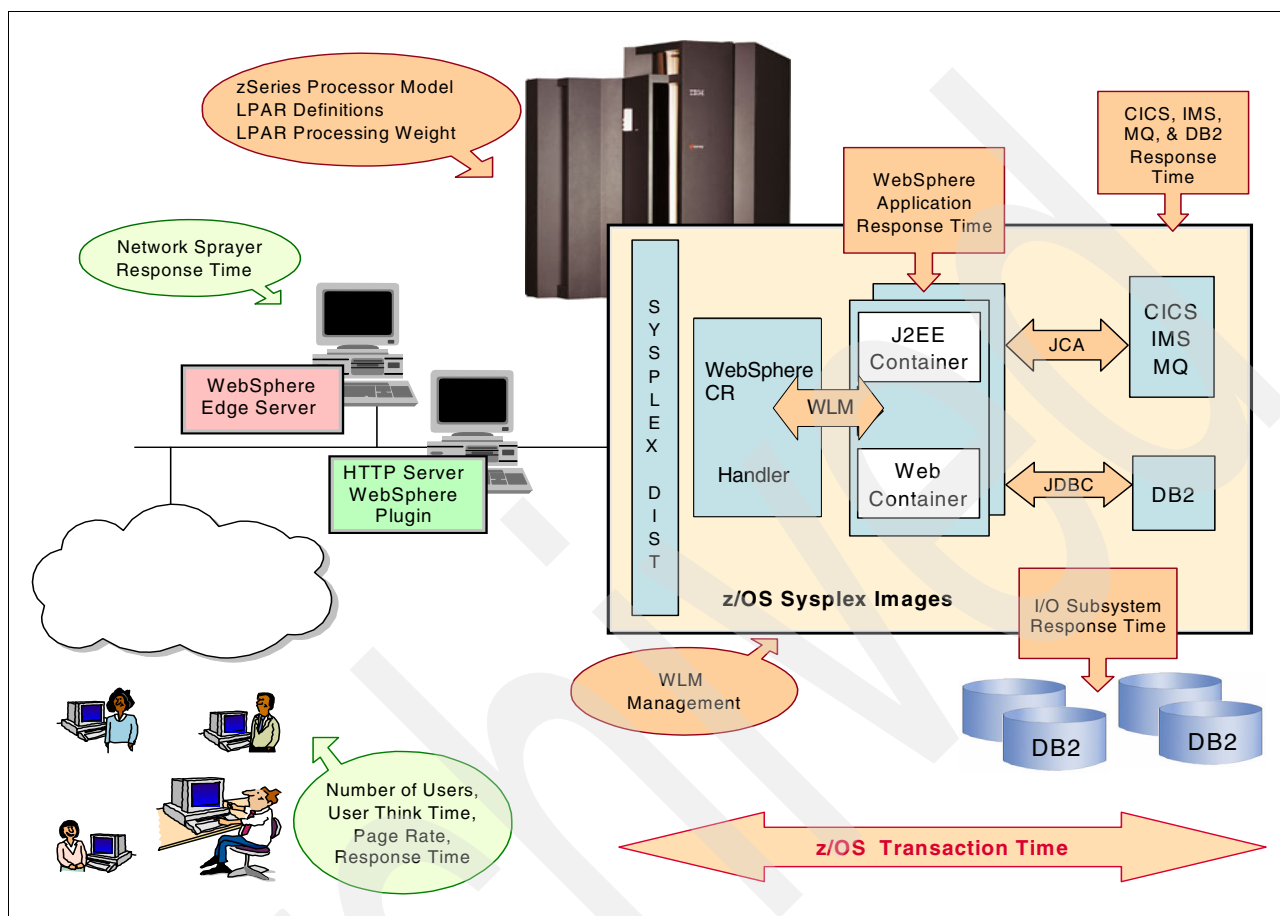


Figure 13-5 Performance monitoring: An overview

13.4.1 Understanding the expectations

Understand the performance expectations of your system resources and applications. In an ideal world, you would have a detailed, documented Service Level Agreement that has been derived from accurate capacity planning and confirmed by detailed historical monitoring data. Reality might be different. The means by which performance expectations are derived are discussed in 13.3.1, “Setting your performance expectations” on page 126. The answer might be “I don’t know.” Be aware that the less you understand your performance expectations, the murkier the problem seems and the harder it is to identify it. At various points in the process, you are required to make a judgement.

To decide whether a number in a report is good or bad for your application with your combination of hardware and software and your business priorities is virtually impossible unless you have a clear understanding of what the data means and what is acceptable in your environment.

Performance analysis has no simple point-and-click solution. The tool that can do this for you has not yet been invented. You must make judgements based on experience, rules of thumb, or guesswork to progress. These judgements might sometimes be wrong and lead you in a wrong direction.

13.4.2 Quantify: Take a quick snapshot view of the system

Performance issues, especially when they originate from user or management complaints, tend to generate strong feelings, even frustration. In such cases, the best solution is not to participate in the turmoil but to gather factual information to quantify the problem as follows:

- ▶ If something is extremely out of line with experience, investigate it immediately.
- ▶ If something is only moderately out of line, start analyzing the hardware and software environment at the system level, particularly the LPAR processing weights, the CPU queue, and the paging activity as follows:
 - LPAR processing weights

Check that the LPAR is receiving the expected level of hardware resources. This should be checked against documented norms for your installation. It is possible that a change has been implemented incorrectly, resulting in less service to the LPAR that you are interested in. There might have been a deliberate change to assign resources to another LPAR for business reasons.

Use the RMF Partition Data Report described in 22.3.2, “Analyzing RMF reports” on page 285, to check the LPAR CPU usage against the guaranteed and maximum share available to your partition. Figure 13-6 shows an example of LPAR CPU activity over 60 minutes, and the guaranteed and maximum share.

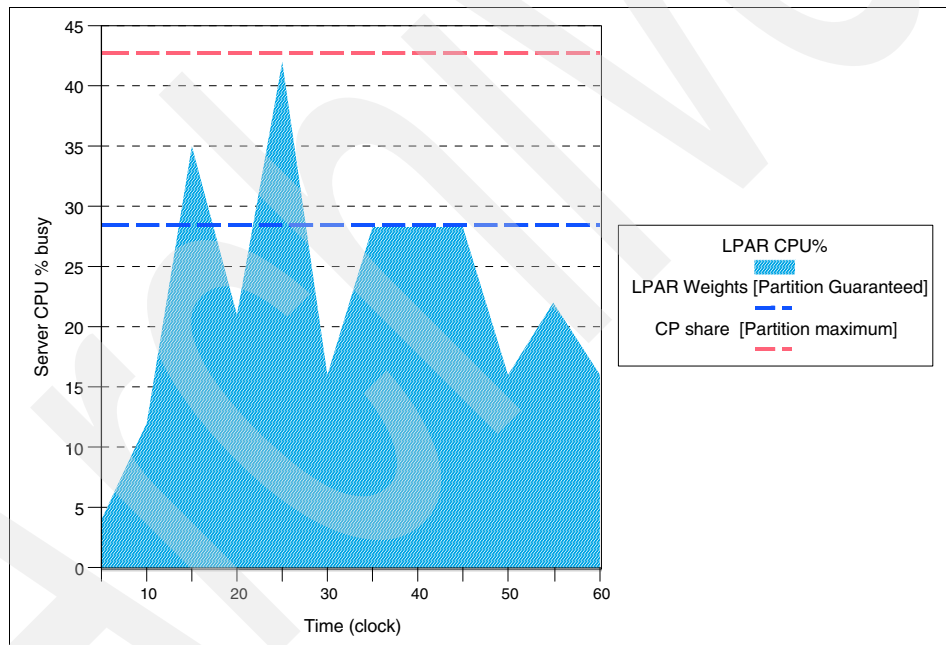


Figure 13-6 Partition view from Partition Data Report

For intervals 35, 40, and 45 it is highly probable that the partition is CPU-constrained to its guaranteed share because of activity in other logical partitions. Although not an anomaly, this is something that you should remember for the rest of the analysis.

Remember that your LPAR CPU share is relative to the sum of the weights of all partitions. As a consequence, your guaranteed share is reevaluated for every change in the logical configuration:

- Every time a logical partition is activated or deactivated
- Every time operations update the processing weights
- Dynamically if your logical partition participates in an LPAR cluster

The Partition Data Report alone cannot tell you whether this is good, bad, or normal, but you can use it to determine whether these factors meet your expectations.

- CPU queue

In the distributed world, running CPU above 50% is unusual. In zSeries, running CPU at 90% or more is not necessarily an indication of a problem and is, indeed, common. Even 100% is not necessarily a problem; in this case, you should investigate further to evaluate how much queuing it causes. The CPU Activity Report can help you.

Check the Queuing Report in the CPU Report. If the queue length substantially exceeds three times the number of CPs online in the configuration, one workload might have a CPU delay problem. Although it might not be a performance problem and might only affect a non-priority batch workload, you should remember it for a later step.

- Paging activity

Check the system paging level in the RMF Summary Report. The RMF Summary Report indicates demand paging rate for the whole system. As with CPU, high paging is not necessarily a problem, but high paging might lead to a CPU penalty and response time problems.

If system paging is indicated, then check if paging occurs at the servant region level. Check the STORAGE and PAGING sections in the Workload Report for the servant regions. Make sure that you check the address space, *not* the enclave, because the enclave shows zero values for PAGING and STORAGE.

13.4.3 Finding the cause

When you have an overview of the system behavior, you can check the WebSphere workload. Whenever possible, try to quantify the characteristics of the workload:

- ▶ Throughput, expressed in transactions per second obtained from the Workload Report
- ▶ CP usage, both total CPU% from the Summary or CPU report, and WebSphere applications from APPL% in the workload report
- ▶ Response time, either an average (AVG) from the Workload Report, or preferably 90th percentile evaluated from the Response Time Distribution Report

Figure 13-7 on page 133 shows two typical workload examples. In a z/OS production system, total CP usage is typically in the 90% to 100% range. This is not a problem. Note that all CPU percentages have been normalized to reflect the capacity of the whole sysplex as explained in 22.3.2, “Analyzing RMF reports” on page 285.

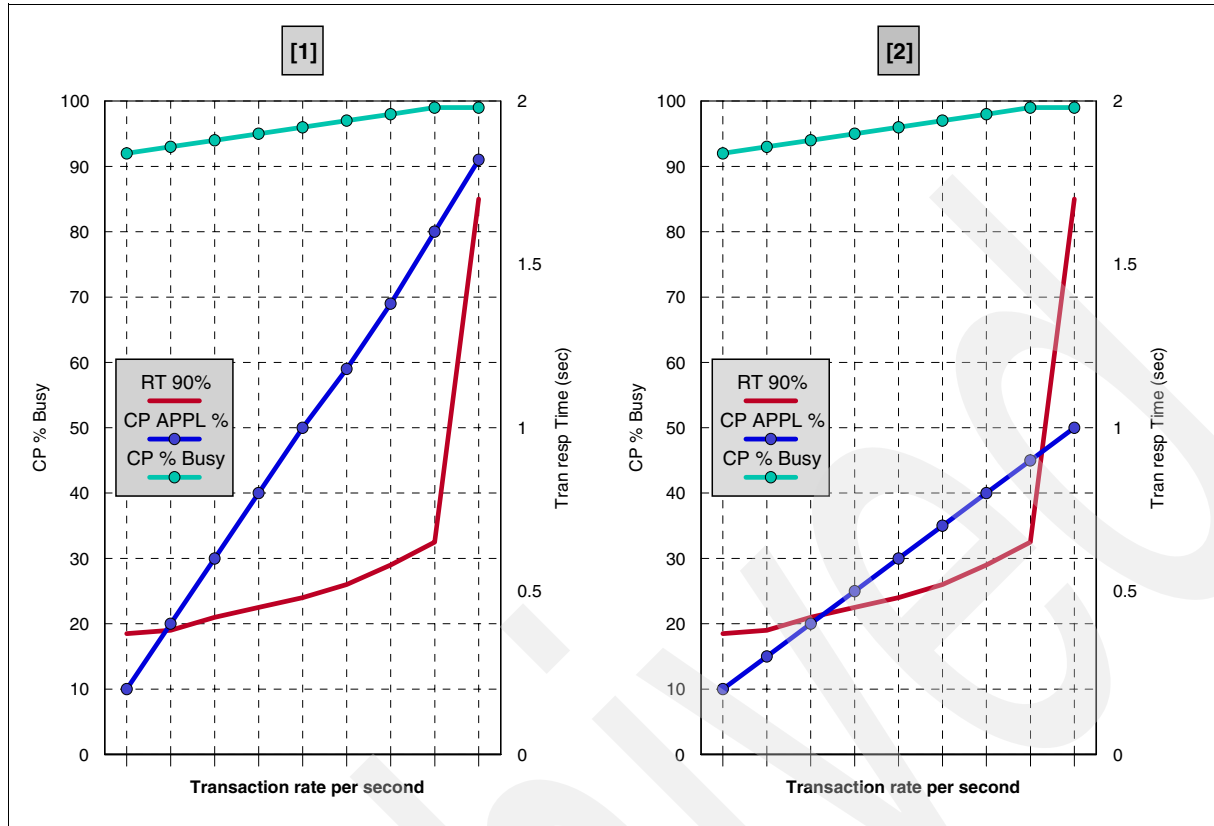


Figure 13-7 CP usage, response time, and throughput

In both examples, the workload CP APPL% grows almost linearly with the transaction rate. This is to be expected when no problem is present. Although visibility might vary with the length of the measurement interval, it is very likely that, in the case of a performance issue, CP usage and throughput do not correlate linearly.

Graph 1 in Figure 13-7 illustrates a situation with no problems. The system behaves as expected in the observed range, even though the amount of CPU resources used might not meet your expectations. The 90th percentile response time remains sub-second until the workload CP usage reaches 90%, where there is an important increase. This is normal.

Graph 2 illustrates a typical throughput problem. The bend of the response time curve appears long before the APPL% CP usage reaches 90%. More investigation is required to determine the cause of the problem:

- ▶ Check the WLM definitions. Other workloads that are running on the sysplex and competing for CP resources might take precedence over WebSphere applications. If this is by mistake, change the WLM settings. If this is desired, WLM is enforcing business priorities as defined and it is no longer an issue that has a technical solution.
- ▶ Check other resources for constraints. If the workload increases, it might be another z/OS-managed physical resource (I/O or storage) or a logical resource that is a constraint. If it is a logical resource, it might be in the WebSphere infrastructure, in another z/OS component, another subsystem (DB2, CICS, IMS), or in the application itself.

Figure 13-8 on page 134 illustrates another practical example of a WebSphere application workload that is running dedicated in a single z/OS image. It was run on partition SC48 with two online CPs (hence the 200% on the CP% busy axis) on a zSeries model 1C8.

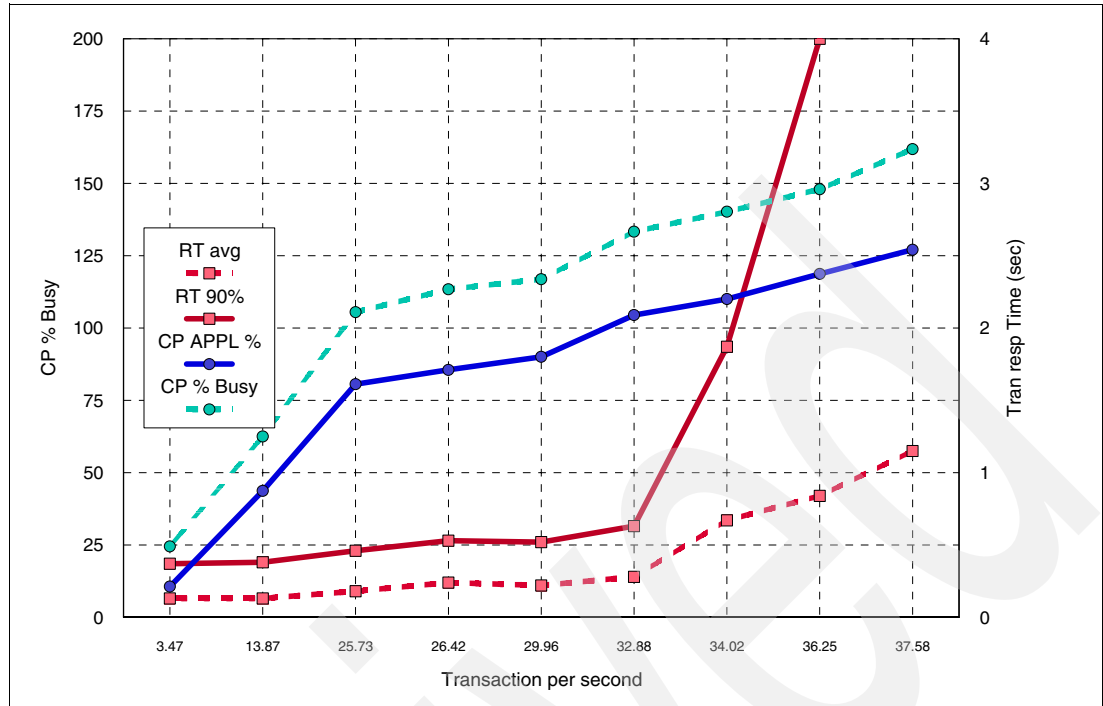


Figure 13-8 CPU% and response time versus throughput

The CPU usage from APPL% plots linearly with the throughput. We used a linear regression with a 0.99 R-square for the example. If the workload APPL% does not plot in a linear fashion, then it is usually an indication of a performance problem. The response time does *not* plot linearly. It slowly grows up to a point where it jumps significantly. The bend of the curve indicates the scalability limit of the workload given the current logical and physical configuration. The bend of the response time curve appears just above 32 transactions per second while the APPL% is approximately 110% and the total LPAR CPU% is 140%.

From the graph in Figure 13-8, you can deduce that:

- ▶ There is a response time problem above 32 transactions per second.
- ▶ It is *not* related to CPU usage of the WebSphere Application Server.
- ▶ It is *not* related to a CPU queuing problem at the server level because the server has not reached the LPAR guaranteed share.
- ▶ A memory problem is not likely.

Another indicator that can prove useful is the average CP usage per transaction. It can be expressed in various units. The authors used the number of milliseconds of CP per transaction.

In normal circumstances the average CP millisecond per transaction should be nearly constant across the throughput range, as shown in Figure 13-9 on page 135. A significant variation is usually an indication of a performance problem.

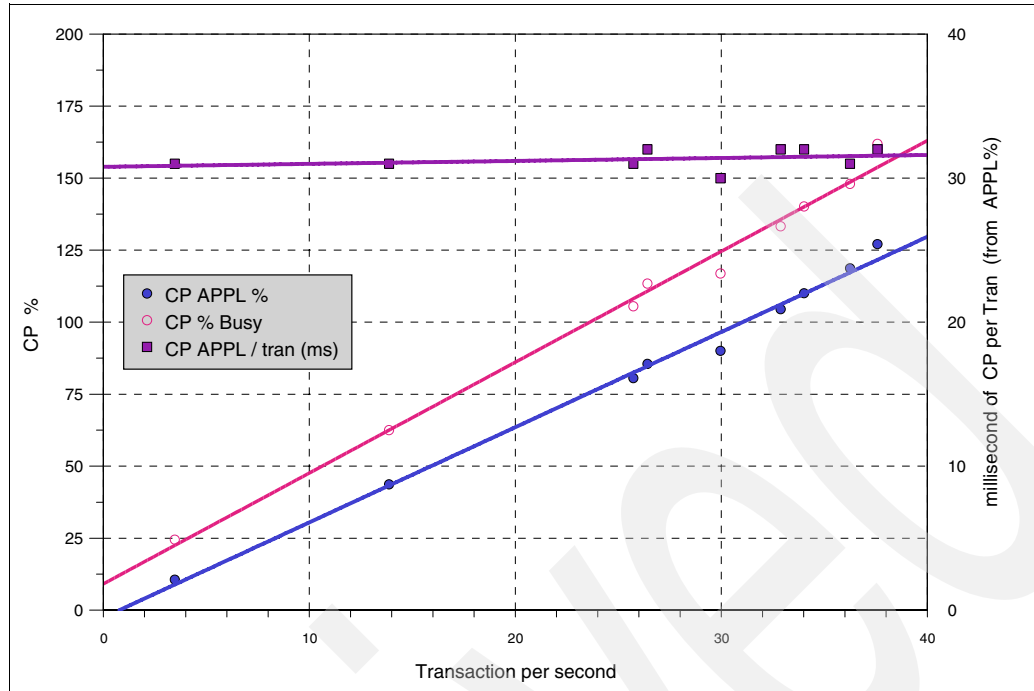


Figure 13-9 CP time per transaction

You can quantify the CP per transaction using three methods, the only important point being that the interpretation of the numbers should be kept consistent with the method chosen:

- ▶ The workload CP APPL%, that is, only the workload that is reported in the enclave. This workload includes application CP time in WebSphere and in any subsystem (DB2, MQ, IMS, or CICS) called on behalf of the transaction.
- ▶ The workload CP APPL% plus the WebSphere server address spaces. The time then reflects variations when additional servers are started/stopped because of the Application Environment or when servers are recycled. It also shows the time that is incurred because of the Garbage Collector.
- ▶ The total CP time, that is, the workload CP APPL% plus the WebSphere server address spaces plus the apportioned uncaptured CP time. Although this is the gross value preferred for cost calculations, it might not be the best one to use for performance analysis.

Depending on your conclusion about where the performance problem might be, you can use one of the three options to analyze and solve the problem:

- ▶ If CPU consumption in the server address space (not the application environment) is higher than expected, you are probably experiencing a memory leak or heap problem. Refer to 13.4.4, “Analyzing a heap or memory problem” on page 136.
 - ▶ If response time becomes significantly worse (getting to the bend of the curve) as you apply more load without using available CPU, then you are probably experiencing a delay problem. Refer to 13.4.5, “Analyzing a response time problem” on page 136.
4. If CPU utilization seems higher than expected for the current transaction rate, you are probably experiencing CPU problems. Refer to 13.4.6, “Analyzing a high CPU usage problem” on page 137.

13.4.4 Analyzing a heap or memory problem

Although it should never happen in a production environment, a memory leak is one of the first issues to investigate if you suspect you are having problems. There are two reasons why:

- ▶ A memory leak creates both CPU and delay problems. Unless you have a specialized tool to analyze WebSphere application performance, reporting values are distorted by the problem and you cannot follow a performance path.
- ▶ A memory leak affects the servant region address space, and, therefore, the availability of the WebSphere infrastructure.

If you suspect a memory leak, you should:

1. Run a verbose garbage collection trace as explained in 21.5, “Java Garbage Collection Formatter” on page 262. If the garbage collection analysis confirms a memory leak, there is very little that can be done in the production environment. Send the faulty application back to development for correction.
2. Check for the correct Java heap size. To do so, check the percentage of time spent processing garbage collection. A good rule is spending less than 5% of your time processing garbage collection.
3. Select a garbage collection cycle after your application has run long enough to reach a steady state. Repeat this test for a number of garbage cycles after that.
4. Locate the YYYY time since the last allocation failure for this garbage collection: “YYYY ms since last AF.” Then, note the XXX time that it took to complete the garbage collection processing: “completed in XXX ms.” Divide XXX by YYYY and see if it is smaller than 5%:
$$\text{"completed in XXX ms"} / \text{"YYYY ms since last AF"} < 5\%$$
5. If your garbage collection processing is using less than 5% of the time, then your heap size is fine. If you are spending 5% or more of your time in Java garbage collection, increase your heap size. Then check your Java GC activity again.

Note: If you are in a storage-constrained system, increasing the Java heap to reduce garbage collection overhead may result in more paging.

13.4.5 Analyzing a response time problem

Analyze the requests that do not meet response time expectations. If the logical user request response times in WebSphere servant regions are fine, check the components before servant regions. This is mostly network related and is outside the scope of this redbook. See 22.1, “TCP/IP related tools” on page 276, for information about specific network problem analysis tools.

For the logical user requests that are not meeting response time expectations in the WebSphere servant regions:

1. Map the logical user requests to WebSphere transactions. Sometimes this is difficult and you must guess or make assumptions.
2. Segregate the WebSphere transactions into sets of good and bad transactions based on response times.
3. Examine the resources used in each component of all bad transactions and identify common features and anomalies.
4. Form hypotheses that explain 80% of the observations of good and bad transaction sets. One hypothesis could be that the code is written badly. Do not attempt to use profilers in production. Send the code back to a test system for profiling.

5. Test hypotheses, one by one, by gathering additional information.
6. Be prepared to repeat this process until the identified problem has been resolved.

For more information about analyzing increased response times of WebSphere for z/OS applications or If you suspect the delay in the back-end subsystems, see *Monitoring WebSphere Application Performance on z/OS*, SG24-6825.

13.4.6 Analyzing a high CPU usage problem

If you experience high CPU consumption in the WebSphere Application Server, follow these steps:

1. Collect RMF Monitor I, including a Workload Activity Report.
2. Make sure that the WebSphere application is really the source of your CPU activity, or if you are consuming CPU somewhere else. See Chapter 12, “High CPU utilization” on page 109.
3. Locate the APPL% value for the application environment associated with your WebSphere application. Calculate the CPU cost per transaction.
4. Compare the WebSphere APPL% value with the APPL% for the whole system and with the APPL% value from other report classes in the system.
5. Check the system uncaptured time for any unusual value.
6. If the problem is really in the WebSphere application, collect System Management Facility (SMF) 120 interval records to help locate the beans and methods with significant CPU activity:
 - To enable SMF data collection, select “Custom properties” under the Administrative Console and select the SMF records requested.
 - Looking at a summary view of your SMF 120 data, you should be able to locate which beans and methods are experiencing poor performance. If you are not familiar with the application, you might need help from your application developer to understand the cause of the problem. Remember that these records report both elapsed time and CPU time. For CPU pain, CPU time values are more interesting.
 - In some cases, SMF 120 data does not provide information that is low-level enough to isolate the problem. For example, activity in servlets, JSPs, and regular Java classes that are called by these servlets and JSPs is accumulated under method, dispatch, bean, RemoteWebAppBean. For lower level analysis of where you have delays, some other tool might be necessary.
 - In some cases, it might be necessary to profile your application. This is something that should not be done in a production system; send the application back to development for further testing. Check WebSphere Studio Application Developer or other Java profiling tools, available from IBM or other vendors. Check the technical Sales Library Web site (Techdocs) for a white paper about WebSphere for z/OS application debugging and profiling:

<http://www-1.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100250>

13.5 Related information sources

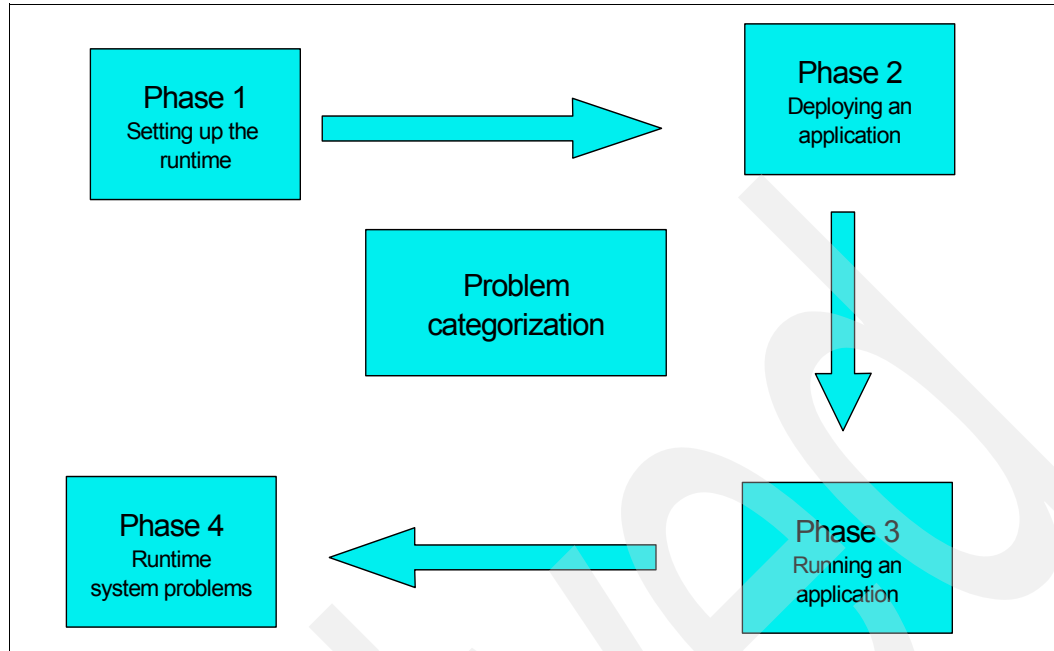
For related information, see:

- ▶ WebSphere Application Server: Performance Information
<http://www-306.ibm.com/software/webservers/appserv/was/performance.html>
- ▶ WebSphere Application Server Best Practices for Performance and Scalability
http://www-306.ibm.com/software/webservers/appserv/ws_bestpractices.pdf
- ▶ Performance Analysis for Java Web sites
<http://www.awprofessional.com/bookstore/product.asp?isbn=0201844540&redir=1&r1=1>
- ▶ The IBM resource for developers
<http://www-130.ibm.com/developerworks>
- ▶ Tuning Garbage Collection with the 1.4.2 Java Virtual Machine
<http://java.sun.com/docs/hotspot/gc1.4.2/>
- ▶ Operating System Environment Manager for z/OS
<http://www-306.ibm.com/software/awdtools/osem/features/>

Problem avoidance and best practices

This part is intended for system programmers and administrators who not only want to solve problems but also want to learn about typical problems in various stages of the WebSphere for z/OS life cycle and how to avoid them.

We identified the possible problem areas that can be encountered when you use WebSphere for z/OS and have arranged them into four specific problem phases that correspond to steps in the WebSphere for z/OS life cycle as shown in the following figure.



Problem categorization based on WebSphere life cycle

These phases are:

- ▶ Phase 1: Installation, configuration, and migration (see Chapter 14 on page 141)
- ▶ Phase 2: Application deployment (see Chapter 15 on page 153)
- ▶ Phase 3: Testing and running applications (see Chapter 16 on page 165)
- ▶ Phase 4: System environment (production) (see in Chapter 17 on page 177)

We give a general overview of the problem areas in the individual phases, explain how to analyze them, and provide valuable hints and tips about how to avoid problems.

Phase 1: Installation, configuration, and migration

To prevent errors and successfully install WebSphere for z/OS, you must implement the necessary features, subsystems, and resources that are required for the runtime environment. Figure 14-1 shows common problem areas.

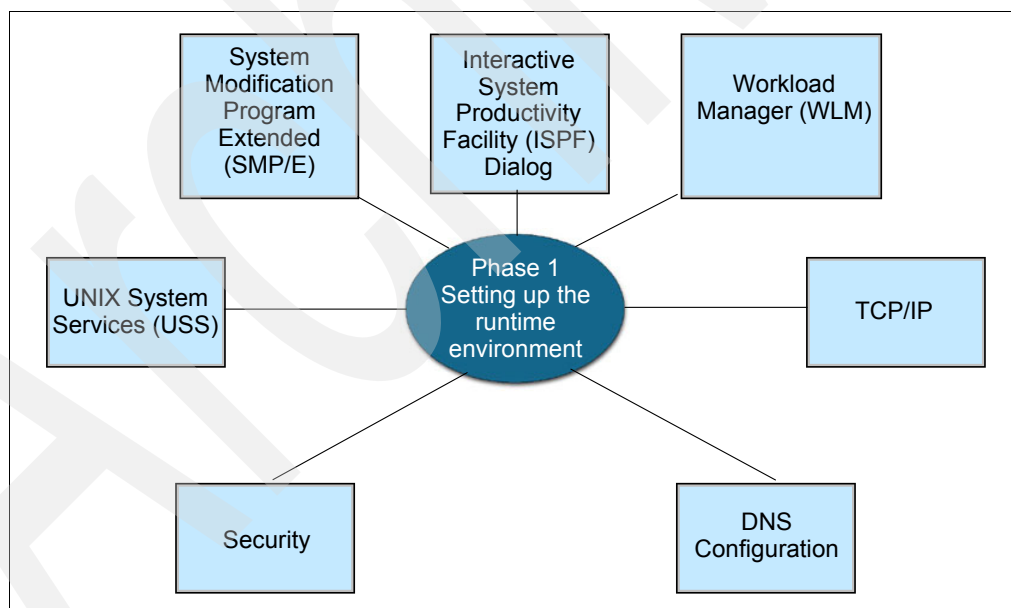


Figure 14-1 Problem areas in Phase 1

This chapter describes various methods for preventing common problems during the installation and migration processes of WebSphere for z/OS. We give hints and tips for the coexistence of WebSphere for z/OS V6 with previous releases and list common problems and their solutions. We also mention means and tools that can help you solve problems in this phase.

14.1 Preparing the Installation

This section is based on best practices and summarizes tasks that you might want to verify before installing your WebSphere for z/OS system:

1. Prepare all the necessary z/OS subsystems and complete the planning for customizing your system environment before attempting to install WebSphere for z/OS.
2. On the WebSphere for z/OS home page in the panel on the left, click **Library** or go directly to:

<http://www-306.ibm.com/software/webservers/appserv/was/library>

Under the tab for V6 and WebSphere for z/OS, you can access *WebSphere Application Server V6 for z/OS Program Directory*, GI11-2825, or go to the *WebSphere Application Server V6.0 Information Center* Web site:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

3. Search for “planning for installation” at the Information Center to understand the various tasks for preparing the installation of WebSphere Application Server V6 for z/OS. Refer to “Steps for creating your implementation plan” and follow all the steps (unless it says optional) to ensure a smooth installation process.

To help prevent errors and successfully prepare the environment for WebSphere for z/OS, use the following checklist:

- ☐ Fill out the worksheet provided by the Information Center after you search for “Checklist: Preparing the base operating system.”
- ☐ Contact your security administrator to set up a RACF user ID and authorize the ID so that it has read/write access to the WebSphere Application Server for z/OS files (BBO.* data sets and HFS files).
- ☐ Increase your paging by one 3390-3 volume if your storage is constrained, or two if your system does any paging of the WebSphere Application Server address spaces.
- ☐ Make sure that your address space is large enough. Some WebSphere application servers might require a 1 GB virtual region to run any workload.
- ☐ Set up the WLM dynamic application environment so that WebSphere for z/OS can use the capacity automatically.

The dynamic WLM application environment is a function of WLM that became available when APAR OW54622 was made available. It is incorporated into z/OS Version 1.5 and later. With the new WLM function, programs can dynamically create application environments.

WebSphere for z/OS V6 is designed to make use of the dynamic WLM application environment if it sees the function available. Generally, this is a function of WebSphere that you cannot turn off.

If the dynamic capability of WLM is not available, WebSphere relies on static WLM application environments. Static application environments that are still in existence when APAR OW54622 is applied simply stop being used in favor of dynamically created ones of the same name.

You also must collect and determine important information about your specific setup for components of WebSphere Application Server for z/OS.

For planning related to naming conventions, TCP port allocation, shared HFS, and clustering, refer to the white paper and spreadsheet available at:

<http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS1331>

The following checklist can help you with your specific setup:

- ☐ The recommended primary size allocation for the HFS is 250 cylinders (3390); the recommended secondary allocation is 100 cylinders (3390).
- ☐ If possible, set up your HFS so that the root HFS is shared by all processors and so that the deployment manager configuration is in an HFS configuration on a system-generic mount point.
- ☐ Understand the HFSs for the application servers, the nodes, the daemons, and the cells.
- ☐ It might be necessary to resize your system dump data sets because of the size of WebSphere address spaces, and where possible, evaluate the use of dynamic dump data sets.
- ☐ If you are running in a sysplex, set up your TCP/IP with Sysplex Distributor to make use of dynamic virtual IP addresses (DVIPAs).
- ☐ If ARM is enabled, you might want to disable it for the WebSphere Application Server address spaces during installation and customization to avoid unnecessary restarts of the address spaces. After installation and customization are complete, you should consider re-enabling ARM.

Search for “Ensuring problem avoidance” at the Information Center for additional information about USS/HFS configurations, System Modification Program Extended (SMP/E) tasks, ISPF dialogs, TCP/IP configurations, and security information.

14.2 Installation and configuration

After the planning is completed and the z/OS base systems are ready for installation, go to the section “Installing your application serving environment” at the Information Center for more detailed information about installing WebSphere for z/OS:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

You can also refer to *Installing your application serving environment*, GA22-7957-03, from the library Web site at:

<http://www-306.ibm.com/software/webervers/appserv/was/library>

To help prevent errors and successfully install and configure WebSphere for z/OS, use the following checklist:

- ☐ Complete the Customization Dialog worksheets.

These worksheets are provided for each task in the installation and configuration dialog boxes to help you determine what values you should enter in the Define Variables stage of customization. You can find the worksheets at the Information Center by searching for “Customization Dialog worksheet.”

Consult your system programmers and administrators and other WebSphere and z/OS experts to determine the values of all the variables.
- ☐ When selecting server, cell, or node names, always avoid special non-alphanumeric characters because they are used as HFS directory names and are parsed in XML files that might have problems with special characters, such as blanks, slashes, dashes, tildes, question marks, or underscores.
- ☐ Where possible, use the default names the first time that you install WebSphere Application Server to make the installation instructions easier to follow.

- ❑ Initial customization of WebSphere Application Server for z/OS V6.0.1 requires that an installation be at a minimum Service Level of 6.0.1.2 (PTF UQ04304). Check for the latest maintenance requirements at the following Web site:

http://www-306.ibm.com/software/webservers/appserv/zos_os390/support/

- ❑ Check the product PSP bucket WASAS601 subset H28W601 to verify that all the suggested maintenance has been applied.
- ❑ Make sure that the product code HFSs are mounted in the directories that you have chosen in the planning session.
- ❑ Your installation might limit (control) the specification of **REGION=**, usually through the JES2 EXIT06 exit or the JES3 IATUX03 exit. If so, relax this restriction for the WebSphere for z/OS JCL procedures.
- ❑ Navigating the configuration HFS with a UID of 0 can alter files or their ownership and permission attributes, making them inaccessible to the WebSphere for z/OS runtime servers and administrators. It is better to use the WebSphere for z/OS administrator user ID.
- ❑ Always run the installation jobs from the same system where WebSphere for z/OS is being installed. Use the JOBPARM card below the JOB card to avoid running the jobs from different systems. The syntax for the JOBPARM is /*JOBPARM SYSAFF=SXX where SXX is the system name.
- ❑ When using DB2 for z/OS, the messaging engine cannot dynamically create the data store tables. This means that you must manually create these tables using the DDL statements produced by the **sibDDLgenerator** command. You can find instructions at:

http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.pmc.zseries.doc/ref/rjm0630_.html

You can redirect the output of the command to a file so that you can submit it to DB2 for z/OS later. One way to do this is to use SPUFI, but before you do this, you must copy the file to an FB80 data set first. As the final step, you must clear the Create tables box (for every messaging engine that use SDB2 for z/OS) from the data store panel of the Administrative Console.

14.3 Migration

When you migrate WebSphere Application Server products, you change the existing environment and applications so that they are compatible with the current product version.

To understand the migration process, go to “Migrating, coexisting, and interoperating,” at the Information Center:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

Or, see *WebSphere Application Server V6 for z/OS, Migrating, coexisting, and interoperating*, SA23-2207, which is available from the WebSphere for z/OS library Web site:

<http://www-306.ibm.com/software/webservers/appserv/was/library>

14.3.1 Migrating from V5.x to V6.0.x

The major difference between V5.x and V6.0.x is that the Java Development Kit (JDK) is embedded in V6.0.x, while it was external to V5.x. Users do not have to worry about the JDK change because the migration process takes care of it during the transformation process.

The migration utilities in WebSphere for z/OS 6.0.x support migration from V5.x. Search for “Migrating product configurations” at the Information Center and use it as a starting point for planning information, customization dialogs, and V5.x to V6.0.x migration explanations for stand-alone application server nodes, deployment managers, and federated nodes.

Migration from V5.x to V6.0.1 is the same as that from V5.0 to V5.1 at the highest level. You copy the existing HFS configuration, transform it to V6.0.x, and write it to a new HFS.

Prior to migration, the old configuration is renamed so that if, for any reason, the migration fails, users should be able to go back to their previous configuration.

See *Migrating from WebSphere for z/OS V5.x to V6 - An Example Migration*, WP100559, which is available from:

<http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100559>

14.3.2 Migrating from V4.0.1 to V6.0.x

There is no migration path from V4.0.1 to V6.0.x. WebSphere for z/OS v6 must be installed from scratch. You must use the Rational® Application Development tool to repackage applications that were developed for V4.0.1 to make them compliant with the J2EE V1.4.2 specification and to be able to deploy them in WebSphere for z/OS V6.0.x.

14.3.3 Migrating from V3.5 Standard Edition to Version 6.0.x

Migration from WebSphere Application Server for z/OS V3.5 Standard Edition (SE) involves changes to application structures, development, and deployment because you are moving to a platform that is fully compliant with J2EE specifications. J2EE technology clearly separates development and the creation of applications from application administration, deployment, and management. Using the J2EE model, you can develop applications independently from their final deployment environment. This task separation simplifies the process of promoting an application from initial development through production, or moving an application from one server to another. The intent is to change only the application deployment parameters, and not the application code.

There is no migration path from Version 3.5 SE to Version 6.0.x. You must use the Rational Application Development tool to redevelop and package applications so that they can be deployed in WebSphere for z/OS V6.0.x.

14.3.4 Checklist for migration

To help prevent errors and successfully migrate WebSphere for z/OS, use the following checklist:

- ☐ Update the prerequisites to the levels required by Version 6.x. Prior levels of WebSphere for z/OS generally continue to run at the higher prerequisite levels.
- ☐ The authors strongly recommend that you be at Service Level 6.0.2.2 (PTF UQ07342) before initiating the migration process.
- ☐ Be aware of other versions of WebSphere for z/OS that you have running in your system when going through the dialog, because the customization dialog does not detect them for you.
- ☐ Before you run any migration jobs, ensure that every application server node in the WebSphere for z/OS V6 system has a WebSphere administrator user ID and password in its soap.client.props file.

- ❑ If you are running other versions of WebSphere or z/OS, watch out for potential location service daemon port collisions or LPA issues that are caused by different versions in one system.
- ❑ Review the ports that have been defined to ensure that the WebSphere for z/OS V6 installation does not conflict with existing port definitions for previous WebSphere for z/OS releases. In particular, when you are installing V6 to coexist with V4.01 or V5.x, note that the default daemon port definition for all versions (V6, V5, and V4) is the same.
- ❑ The authors recommend setting up V6.x installation with the STEPLIB if V6.x must coexist with any prior releases. Because of naming conflicts, V5 and V6 product code cannot be in LPA at the same time. To support coexistence:
 - Place the V6 SBBOLPA data set in the STEPLIB of the V6 daemon. If a prior level of SBBOLoad is in LPA, add a V6 STEPLIB for SBBOLoad.
BBORTSS5 must be in LPA to make CTRACE work. If V5.0x is already installed and running in the system, then BBORTSS5 must be in LPA already. To check whether BBORTSS5 is in LPA, use D PROG,LPA,MOD=BBORTSS5 from the SDSF, syslog, or console. This does not cause any coexistence issues, because the DLL name is different from V4.0.1 and the module is the same for V5 and V6.
 - Only one set of PPT entries can be active at one time for a given program. WebSphere for z/OS V4.0.1 and V5 both use the BBOCTL program as a controller region.
If you are running V4.0.1 and V5 on the same system, their BBOCTL programs share the same PPT entry. Prior to V4.0.1, Service Level W401610, the PPT attribute SYST was required for the BBOCTL program; it is not required after that service level. Therefore, including the SYST keyword in the PPT entry for the V5 BBOCTL program causes an informational message (IEF188I PROBLEM PROGRAM ATTRIBUTES ASSIGNED) to appear when you start a V5 server (V5 does not require the SYST attribute). This message does not affect the functionality of the WebSphere address spaces.
If you do not want this message to appear, and your V4.0.1 Service Level is at least W401610, you can delete SYST from the SCHEDxx member to stop the message from being generated.
If you are not at this service level, you must leave SYST in the PPT for BBOCTL to start the V4.0.1 server. IEF188I is issued when V5 server is started as long as BBOCTL is defined as a system task.
 - The authors recommend using the WLM dynamic application environment when you are configuring V6 so that a specific server name can be used by V6 and by a server on V5.
- ❑ A return code of 0 means nothing in a migration job. Be sure to review the .err and .out logs carefully for diagnostic information.
- ❑ A migration cannot be restarted after the process is started. If something fails during the process, you must start again from the beginning.
- ❑ Always run the jobs from the same system where the node being migrated is located.
Use the JOBPARM card below the JOB card to avoid running the jobs from different system. Syntax for the JOBPARM is /*JOBPARM SYSAFF=SXX where SXX is the system name.

14.4 Coexistence

WebSphere for z/OS V6 can coexist with any of the prior WebSphere releases on the same LPAR.

V6.0.x can coexist with V5.0.x or higher in the same cell with a few known limitations, such as:

- ▶ The Deployment Manager must be at the highest release level in the cell that has mixed releases.
- ▶ If you have multiple V5.0.x nodes on the same LPAR, they all should be migrated simultaneously because Version 5.0.x nodes cannot coexist with Version 6.0.x nodes on the same LPAR. This restriction does not apply for V5.1.x and V6.0.x nodes.
- ▶ The V6.0.x Deployment Manager can manage Version 5.x nodes. In a coexistence situation, the V5.x configuration tree cannot be modified until Service Level W6012XX or higher of the V6.0x driver. At this level, the restriction is lifted but modification can be done to existing nodes and servers.

Important: There must only be *one* daemon *per cell* in an LPAR.

The following restrictions apply when multiple releases or versions are in the same LPAR:

- ▶ Cells cannot have the same short names.
- ▶ Only one version of the code can exist in LPA/LNKLST on the same LPAR; the others must be included in the STEPLIB.

For successful coexistence, ensure that:

- ▶ The load modules are in LPA for one system, and the load modules are in STEPLIB for the other system.
- ▶ The ports are unique between the two systems.
- ▶ The daemon_group_name values are unique between the two systems. This is a known cause of the ABEND EC3 with reason code 02060018.

14.5 Most common problems

This section describes the most common problems that users encounter during WebSphere for z/OS V6 installation, configuration, migration, and in a coexistence setup. The following list includes typical problems, solutions, and tools to use:

- ▶ **Problem:** You receive GIM23901E messages during APPLY using CBPDO.
 - **Solution:** Apply the latest maintenance to solve the problem.
 - **Tools used:** SYSLOG and job log from the SMP APPLY step.
- ▶ **Problem:** You receive “Error CEE3250C. The system or user abend SCC3 R=00020001 was issued from compile unit BBODBDLD at entry point BBODBDLD.”
 - **Solution:** Put SBBOLPA in LPA or in STEPLIB when you are installing WebSphere Application Server for z/OS. The problem is a documentation error in the Customization Dialog. See APAR PQ80728.
 - **Tools used:** SYSLOG, job log, and editors.

- **Problem:** Message IGW01513T appears during the SMP/E receive of WebSphere for z/OS and OS/390®. This message is produced by the utility IEBCOPY while copying files to TLIB (target library).

- **Solution:** This error is the result of the output record format of the PDSE being forced to *Fixed Blocked* (FB) when it should be *Undefined*. If you have a DFSMS ACS routine that controls the allocation of the PDSE data sets and forces the PDSE record format to FB, this problem is the result. You can resolve it if you use a DFSMS ACS routine to ensure that PDSEs are created with:

RECFM=U

- **Tools used:** SYSLOG and job log from SMP receive.

- **Problem:** After building your WebSphere for z/OS V5 environment, you need to change the WebSphere configuration root directory.

- **Solution:** It is part of the setup process to specify the WebSphere configuration root directory. You then run a series of configuration jobs to build the WebSphere for z/OS V5 environment. After building your V5 environment, there is no simple way to change this root directory. The only option is to rebuild the V5 environment from scratch, specifying the new configuration root directory in the ISPF panels and rerunning all of the configuration batch jobs. If you have made significant changes to the V5 environment, such as defining several clusters, creating various definitions, and installing a number of applications, you must redo all of this work, which can be a lengthy process. The value of the configuration root directory file is that it is stored in several places. This value should be changed in several files and some definitions could be easily missed. A procedure that addresses this problem can be found on the WebSphere for z/OS support page. Search for “configuration root directory.”

- **Tools used:** None.

- **Problem:** You receive this error message in the console:

SECJ4046E: Duplicate login configuration name system.wssecurity.IDAssertion.
Will over write. Duplicate login configuration name
system.wssecurity.Signature. Will over write.

- **Solution:** Remove all duplicate entries in wsjaas.conf, and then restart WebSphere for z/OS. If the node is part of a WebSphere cell, you might need to remove duplicate entries in all four locations where wsjaas.conf is stored:

- `install_root /bin/wsinstance/propdefaults`
- `DeploymentManager/bin/wsinstance/propdefaults`
- `install_root /properties`
- `DeploymentManager/properties`

- **Tools used:** None.

- **Problem:** You do not know how to check the version and history information of your WebSphere Application Server for z/OS V5 environment.

- **Solution:** Look in the SystemOut.log file for the specific installation for the base `Install_Root/logs/nodeagent/SystemOut.log` and for the Deployment Manager `Install_Root/logs/dmgr/SystemOut.log`.

Alternatively, you can run the versionInfo command from the /bin directory of the specific installation, for example:

`Install_Root/bin>versionInfo.sh`

- **Tools used:** None.

- **Problem:** When WebSphere V5.0.2 for z/OS is running, starting up a new V5.1 application server causes the Control Region to abend with SEC3 in the BBOSSACE module.

- **Solution:** Further review of the configuration setting in the job logs for both V5.0.2 and V5.1.0 shows that the names of the daemon group (cell name) are identical:

daemon_group_name: PDCELL

This is a coexistence issue. Rename daemon_group_name for V5.1.0 to a different name.

- **Tools used:** Job log.

- **Problem:** When you are generating installation jobs, you receive the error message BBOMNINS stating that:

BBOMNINS: BBOIPCSP does not exist

Or:

BBOSCHED does not exist.

The instructions for the creation of a Deployment Manager node referred to two optional members, BBOSCHED and BBOIPCSCP. However, they are not being generated for use. Also, the START command for the node agent in the managed node generates instructions that reference the wrong node name.

- **Solution:** Update the Customization Dialog to generate the missing jobs, and correct the START command. APAR PK07293 is associated with Service Level (Fix Pack) 6.0.2.1 (Build Level cf10533.10) of WebSphere Application Server V6.0.1 for z/OS.

- **Tools used:** FTP tool.

- **Problem:** The release number no longer appears in the job output, and the following message appears in the log:

BBOM0007I CURRENT CB SERVICE LEVEL IS build level cf10515.05

- **Solution:** The cf1 stands for Cumulative Fix 1. You will see a cf2 in the future when a series of WebSphere for z/OS APARs is rolled into Cumulative Fix 2. Therefore, *cf10515.05* stands for: Cumulative Fix1, and the service date is the fifteenth week of 2005 (0515), and the .05 is the day of the week that the PTF was cut. The .05 is for internal use and it could be any value from .01 to .05. So the next set of PTFs could be cf10521.xx because we expect a PTF to be available about once every six to eight weeks. As to whether the cf1 will change with each PTF release, the answer is no. Only a major series of PTFs (a level set across WebSphere family of products across platforms) would result in the cf1 changing to cf2. So, you can anticipate cf10521.xx or cf10523.xx for the next series of fixes. Although it is more difficult, you still must keep track of WebSphere maintenance levels. See the APAR/PTF table for WebSphere Application Server V6.0.1 for z/OS at:

http://www-306.ibm.com/software/webservers/appserv/zos_os390/support/

- **Tools used:** FTP tool.

- **Problem:** After migrating the deployment manager cell from WebSphere Application Server for z/OS Version 5.0.2 at service level W502030 to Version 6.0.1.2, the Deployment Manager does not start, although the migration job ran successfully. The error message is:

BB000220E: WSVR0009E: Error occurred during startup

META-INF/ws-server-components.xml com.ibm.ws.runtime.WsServerImpl

com.ibm.ws.runtime.WsServerImpl

- **Solution:** If trace is not enabled, enable it first and analyze the trace. In this case, the messages in Example 14-1 were issued and used to analyze and solve the problem.

Example 14-1 Error message when start up deploy manager

```
BB000222I: SECJ0240I: Security service initialization completed successfully
BB000222I: PROX0000I: z/OS Web Router v6 deployment recognized with configuration
root=<null>.
BB000221W: SECJ0288E: Error during security initialization.
BB000222I: HMGR0206I: The Coordinator is an Active Coordinator for core group
DefaultCoreGroup.
BB000220E: HMGR0024W: An error was encountered while looking up the IP address for
the host name of a core group member. The host name is GENERIC and the server name
is SYSPROG\SYSPROG\dmgr. The member will be excluded from the core group.
com.ibm.ws.hamanager.coordinator.dcs.CoreStackMembershipManager
com.ibm.ws.hamanager.coordinator.dcs.CoreStackMembershipManager
BB000220E: HMGR0002E: HA Manager services on this process were not started. This
server is not a member of a core group.
com.ibm.ws.hamanager.coordinator.dcs.CoreStackMembershipManager
com.ibm.ws.hamanager.coordinator.dcs.CoreStackMembershipManager
BB000222I: CWRCB0103I: The core group bridge service has stopped.
BB000220E: WSVR0009E: Error occurred during startup.
META-INF/ws-server-components.xml com.ibm.ws.runtime.WsServerImpl
com.ibm.ws.runtime.WsServerImpl
...
Caused by: com.ibm.wsspi.hamanager.HAException: Local Member SYSPROG\SYSPROG\dmgr
is not a member of the core group .at
com.ibm.ws.hamanager.coordinator.dcs.CoreStackMembershipManager.<init>(CoreStackMe
mbershipManager.java:129) .at
com.ibm.ws.hamanager.coordinator.impl.DCSPluginImpl.<init>(DCSPluginImpl
.java:207)
.... 13 more
com.ibm.ws.runtime.WsServerImpl com.ibm.ws.runtime.WsServerImpl
BossLog: { 0023} 2005/06/19 23:28:22.241 01 SYSTEM=XCSF SERVER=PPT5MGR
PID=0X05010055 TID=0X18AA0410 00000000 c=UNK ./bbolsys.cpp+839 ... BB000157E JVM
EXIT API DRIVEN. JVM EXITING WITH CODE=-1
```

An error was encountered during the IP address lookup for the host name of a core group member. The host name is GENERIC and the server name is SYSPROG\SYSPROG\dmgr. The member is excluded from the core group. One of the environment variables in the BBOWMG3D migration job is dcsHost=GENERIC.

After PTFs UK04303 and UK04304, the High Availability Manager Host has no field for entering the name in the second panel of “3. Server customization” (in the migration process). Therefore, the BBOWDMG3 job is generated with the dcsHost=GENERIC value.

Edit the dcsHost variable in the job to state the correct host name and rerun that job. Start the Deployment Manager again.

– **Tools used:** Trace Analyzer for WebSphere Application Server.

- **Problem:** During the configuration of WebSphere Application Server V6 for z/OS and an attempt to access the Network Deployment cell, an error message appears (in BBORBLOG):

SRVE0017W: A WebGroup/Virtual Host to handle Not found. has not been defined.

- **Solution:** This symptom is common when the BBOWWPFJ job was not executed correctly. Check if the return code was zero (RC=0). If there is a TIME option in the JCL for the BBOWWPFJ job your task can exceed the CPU runtime limit. Remove this option and submit this job again.

- **Problem:** After migrating an application from a WebSphere for z/OS V5 cluster to version 6, the application started successfully. You were prompted for a password and authorized successfully but the browser returned an HTTP 404 error.
 - **Solution:** Several attempts to migrate the application failed. After analyzing the problem systematically using the flow charts from Part 2, “Problem symptoms and their resolutions” on page 39, we came to the conclusion that it is not a migration or an application server problem. The error was related to using security functionality with IBM Tivoli Access Manager for z/OS. We changed the Tivoli Access Manager configuration and applied the latest maintenance for WebSphere for z/OS to include the latest Java for z/OS updates to fix the problem.
 - **Tools used:** Flow charts, Tivoli Access Manager, latest WebSphere for z/OS maintenance release.
- **Problem:** When logging out of the Administrative Console, you often receive an HTTP Error 404:

Error 404 An error occurred while processing request:
 /ibm/console/ibm/console/logon.jsp
 Message: SRVE0200E: Servlet [_ibmjsp.ibm.console._logon]: Could not find required servlet class - _ibmjsp.ibm.console._logon

 - **Solution:** This is a known problem and relates to versions before 6.0.2. It is addressed in APAR PK07829, which is shipped with version 6.0.2. Apply the maintenance to fix this problem.
 - **Tools used:** APAR PK07829, which is shipped with version 6.0.2.

14.6 Related references

Learn about using WebSphere Application Server with this reading list, compiled for customers, consultants, and other technical specialists by IBM Software Services for WebSphere:

http://www-128.ibm.com/developerworks/websphere/library/techarticles/0305_issw/recommendedreading.html

Consider using WebSphere for z/OS IBM services. For example, you could have “WebSphere Proof of Concept for z/OS.” IBM consultants design and implement a working solution to your business problem that is identified during the Architecture and Design Workshop by using preconfigured IBM hardware and software. This solution allows you to implement a WebSphere production environment without interfering with your day-to-day business functions. For more information, see:

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

Archived



Phase 2: Application deployment

This chapter provides you with information about how to avoid the most common problems that occur while you are assembling and deploying an application. We mention some typical problems and their solutions, and list some helpful tools and commands for this phase.

Note: This chapter assumes that the WebSphere for z/OS run time has been installed correctly with the latest maintenance, and all the installation and configuration jobs have run successfully.

15.1 Tools for the deployment phase

Before installing an enterprise application or other installable module in an application server, you must develop and assemble the module and configure the target server. Before choosing a server as a target for the module, ensure that the node version for the server is compatible with your module.

Develop and assemble J2EE modules using one of the three tools supported by WebSphere Application Server for z/OS for assembly:

- ▶ Application Server Toolkit
- ▶ Rational Web Developer
- ▶ Rational Application Developer

For more information about these tools, search for “Assembling applications” in the Information Center at:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

To install and deploy the application files, you can use the wsadmin tool for production environments and unattended operations or command-line tools to start and stop application servers, check server status, add or remove nodes, and complete similar tasks. WebSphere for z/OS also supports a Java programming interface for developing administrative programs. All of the administrative tools that are supplied with the product are written according to the API, which is based on the industry standard Java Management Extensions (JMX) specification.

15.1.1 Installing and deploying application files

These are the tools supported by WebSphere Application Server V6 for z/OS for installation and deployment of application files:

- ▶ Administrative Console

See the topics “Installing application files with the console” and “Starting and stopping applications” in the Information Center for information about using the Administrative Console to install and deploy applications. We focus on problems related to this tool.

- ▶ wsadmin scripts with **startApplication**

The wsadmin tool only supports Java Command Language (JACL) and Jython scripting languages. JACL is the language specified by default. If you want to use the Jython scripting language, use the -lang option or specify it in the wsadmin.properties file. Start the wsadmin scripting client interactively, as an individual command, in a script, or in a profile. Then, refer to “Deploying applications using scripting” and “Starting applications with scripting” at the Information Center for more information.

Because the wsadmin tool is mainly intended for production environments and unattended operations and because it is described in detail at the Information Center, we focus on problems that occur when the Administrative Console is used in the deployment process.

- ▶ Java administrative programs that use JMX APIs and ApplicationManager or AppManagement MBeans

WebSphere for z/OS supports access to the administrative functions through a set of Java classes and methods. You can write a Java program that performs any of the administrative features of the WebSphere for z/OS administrative tools. You can also extend the basic WebSphere for z/OS administrative system to include your own managed resources.

Investigate these tools with the Java APIs to determine the best ways to administer WebSphere for z/OS and your applications. For information about the Java APIs, see “Java Management Extensions (JMX) API documentation” at the Information Center, which outlines the following procedure for taking advantage of these tools:

- a. Create a custom Java administrative client program using the Java administrative APIs.

This topic describes how to develop a Java program that uses the WebSphere Application Server administrative APIs to access the administrative system of WebSphere for z/OS.

- b. Extend the WebSphere for z/OS administrative system with custom MBeans.

This topic describes how to extend the WebSphere for z/OS administration system by supplying and registering new JMX MBeans in one of the application server processes. In this case, you can use the administrative classes and methods to add newly managed objects to the administrative system.

- c. Deploy and manage a custom Java administrative client program for use with multiple J2EE application servers.

This topic describes how to connect to a J2EE server and how to manage multiple vendor servers.

- d. Manage applications through programming.

This topic describes how to use Java MBean programming to install, update, and delete a J2EE application in WebSphere for z/OS.

- Java programs that define a J2EE DeploymentManager object in accordance with J2EE Deployment API Specification (JSR-88)

JSR-88 defines a contract between a tool provider and a platform that allows tools from multiple vendors to configure, deploy, and manage applications on any J2EE product platform. The tool provider typically supplies software tools and an integrated development environment (IDE) for developing and assembling J2EE application modules. The J2EE platform provides application management functions that deploy, undeploy, start, stop, and otherwise manage J2EE applications.

WebSphere for z/OS is a J2EE 1.4 specification-compliant platform that implements the JSR-88 APIs. See the Information Center topic “Installing J2EE modules with JSR-88” for more information.

15.1.2 Logging and tracing

The tasks for developing, deploying, and maintaining applications are complex. When an application encounters an unexpected condition, it might not be able to complete a requested operation. You want the application to inform the administrator that the operation has failed and why it failed so that the administrator can take the proper corrective action.

Application developers need to gather detailed information that relates to the path of a running application to determine the root cause of a failure that is due to a code bug. The facilities that are used for these purposes are typically referred to as *logging and tracing* and you access them as follows:

1. In the Administrative Console, select **Troubleshooting** → **Logs and Traces** → **app_server** → **Change Log Detail Levels**.
2. Under the configuration tab, specify a log detail level for a predefined group of components.

15.2 Problem avoidance checklist

A smooth deployment process is your main goal. However, things do not always go as planned. The checklists in this section are based on best practices to minimize the number of problems during the deployment phase. Check them before deploying an application and increase your chances for successful deployment.

15.2.1 Assembling an application

When you assemble an application, use this checklist:

- ☐ Ensure that the application is J2EE 1.4 compatible.
- ☐ The bean names must be unique within a given JAR file.
- ☐ Generate and deploy code using the proper assembly tools.

For detailed information about assembling applications, starting an assembly tool, and configuring an assembly tool, see the *WebSphere for z/OS Information Center* topic “Assembly tools” at:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

- ☐ Verify your data sources and create them for your application if you have not done it before.
- ☐ Configure your database, Enterprise Information System (EIS), and other components properly so that there is no subsystem problem.
- ☐ Disable automatic regeneration of the config files by updating the global.properties file with an editor and setting:

```
com.ibm.servlet.engine.disableAutoPluginCfg=true
```

The configuration information must be updated manually when servlets are added to a Web application so that the WebSphere for z/OS Web Server Plug-in can recognize the new servlets.

- ☐ Tune and set the JVM heap size using this path: **Servers** → **Application Servers** → **server_name** → **Process Definition** → **Java Virtual Machine**.

The best way to determine the necessary size is to run a series of tests, each time increasing the initial size. When you are tuning a production system where the working set size of the Java application is not understood, a good starting value for the initial heap size is 25% of the maximum heap size. The JVM then tries to adapt the size of the heap to the working set size of the application.

15.2.2 Deploying an application

When you are deploying an application, use this checklist:

- ☐ Check to see if the correct version of the EAR file and the full path name of the EAR on the server or on the client machine have been chosen.
- ☐ On the first Preparing for application installation page, specify the context root if you are installing a stand-alone WAR file.
- ☐ On the first Preparing for application installation page, specify whether default bindings for any incomplete bindings you might have should be generated.
- ☐ Specify whether to precompile JavaServer™ pages (JSP) files as a part of the installation. The default is not to precompile JSP files. This option can only be installed in a V6.x deployment target; otherwise, the installation is rejected.

- ❑ Specify the directory where the application EAR file is to be installed.

In a network deployment configuration, by default the application is installed in the APP_INSTALL_ROOT/network_cell_name directory. In a base configuration, it is installed in the APP_INSTALL_ROOT/base_cell_name directory.

- ❑ Choose **Deploy enterprise beans**, if:

- The EAR file was assembled with an assembly tool such as Rational Application Developer, and the EJBDeploy tool was not run during assembly.
- The EAR file was not assembled with an assembly tool.
- The EAR file was assembled using versions of the Application Assembly Tool (AAT) previous to Version 5.

This option allows the EJBDeploy tool to run during application installation and generates code that is required to run EJB files.

Note: Choosing this option might cause the installation program to run for several minutes.

- ❑ Ensure that the application name is unique in a cell and does not contain characters that are not allowed in object names.
- ❑ Select **Deploy WebServices** if the EAR file has modules that are using Web services and has not previously had the wsdeploy tool run on it.

The wsdeploy tool then can run during installation of the application and can generate the code that is required to run applications that use Web services.

15.3 Most common problems

This section is a collection of common problems that occur during application installation. It is not meant to be a comprehensive list, but rather to solve some typical problems that you might encounter:

- **Problem:** No response from the Administrative Console.

- **Solution:** Use SDSF to find which server is utilizing more CPU. One factor for high CPU is enabling Performance Monitoring Infrastructure (PMI), Logs and Traces. Remove the unwanted selection and save the configuration. Then stop and start the application server. Restart the Administrative Console with the specific port number.

- **Tools used:** Administrative Console message log, SDSF.

- **Problem:** Large applications to synchronize or memory is constrained.

- **Solution:** Adjust the JVM options to limit memory usage and therefore reduce the possibility of receiving Out Of Memory errors. The instance synchronization JVM uses default settings, unless you change the JVM options. Set the JVM options using the INSTANCE_SYNC_JVM_OPTIONS property:

```
asadmin set domain.node-agent.node_agent_name.property.INSTANCE_SYNC_JVM_OPTIONS="JVM_options"
```

An example is:

```
asadmin set domain.node-agent.node0.property.INSTANCE_SYNC_JVM_OPTIONS="-Xmx32m -Xss2m"
```

The node agent is *node0* and the JVM options are *-Xmx32m -Xss2m*.

For more information about JVM options, see the Web site at:

<http://java.sun.com/docs/hotspot/VMOptions.html>

Important: Restart the node agent after changing the `INSTANCE_SYNC_JVM_OPTIONS` property because the node agent is not automatically synchronized when a property is added or changed in its configuration.

- **Tools used:** Administrative console message log, SYSPRINT and job log of the server, and WebSphere for z/OS error log.
- **Problem:** The plug-in of the HTTP Server is unable to recognize the availability of another server because one server is down.
 - **Solution:** To avoid this problem, tune the WebSphere HTTP plug-in configuration parameters to fit your environment so that users can experience fewer delays and failover performance of the WebSphere environment improves.
 - **Tools used:** HTTP Plug-in config files, error logs.
- **Problem:** When a large application starts, your application server hangs and then shuts down. You get an ABEND SEC3.
 - **Solution:** You are experiencing a timeout in the HTTP transport. The recommended way to solve this is to find the reason for the timeout by analyzing the SVC dump. See “Analyze the SVCDUMP.” on page 54 and “SVC dumps” on page 247 for more information. You might have to increase or even disable the deployment manager timeout variables to circumvent the problem for a while or to enable the SVC dump.
 - **Tools used:** Administrative Console message log, system log, SYSPRINT and job log of the server, and WebSphere for z/OS error log.
- **Problem:** You are unable to install a large EAR file and the following message appears in the WebSphere for z/OS error log:
BB000271E HTTP REQUEST EXCEEDED 10485760 BYTE INPUT BUFFER
 - **Solution:** Increase the following variables as follows by selecting **Environment** → **Manage WebSphere Variables**:
 - `protocol_http_large_data_inbound_buffer` = 20485700 (or some other large number)
 - `protocol_http_large_data_response_buffer` = 20485700 (or some other large number)

After setting these variables, recycle the DMGR and install the EAR file. If you receive the same message with the larger byte input buffer that is referenced in the error message, increase the number in the variables again.
 - **Tools used:** Administrative Console message log, SYSPRINT and job log of the server, and WebSphere for z/OS error log.
- **Problem:** Session data integrity is lost when concurrent access to a session is made in different Web modules.
 - **Solution:** This problem can occur when two Web modules are installed on different servers. If this is the case, the applications might share session attributes between Web modules using distributed sessions, but session data integrity is lost. Also, the use of some session management features such as `TIME_BASED_WRITES` is severely restricted.
 - Install the Web modules in an enterprise application on one server to share session attributes to eliminate these problems as follows:

- i. Start the assembly tool.
 - ii. In the assembly tool, right-click the application (EAR file) that you want to share and select **Open With** → **Deployment Descriptor Editor**.
 - iii. In the application deployment descriptor editor of the assembly tool, select **Shared session context** under WebSphere Extensions. Make sure the class definition of attributes that are put in the session are available to all Web modules in the enterprise application. The shared session context does not fully meet the requirements of the specifications.
 - iv. Save the application (EAR) file. In the assembly tool, after you close the application deployment descriptor editor, confirm that you want to save the changes that you made to the application.
- **Tools used:** WebSphere Administrative Console for z/OS, Assembly tool.
- **Problem:** When you deploy an application using wsadmin, the plugin-cfg.xml is not updated.
 - **Solution:** You should install the application with a target of -cluster. After installation and before the application is saved, use \$AdminApp edit to add the additional mapping to the Web server. After the application is saved, plugin-cfg.xml is regenerated.
 - **Tools to use:** wsadmin.
 - **Problem:** Documentation for Rollout Update for cluster deployment is not clear for use in scripting with wsadmin.sh
 - **Solution:** The function for Rollout Update can be found under AdminTask for scripting with wsadmin.
- Refer to “Commands for the AdminTask object” in the Information Center at:
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>
- The updateAppOnCluster command can be used to synchronize nodes and restart cluster members for an application update that is deployed to a cluster. After an application update, this command can be used to synchronize the nodes without stopping all the cluster members on all the nodes at one time. This command synchronizes one node at a time by stopping the cluster members to which the application is targeted, by performing a node synchronization operation, and by restarting the cluster members. This command might take more time than the default connector timeout period, depending on the number of nodes that the target cluster spans. Be sure to set proper timeout values in the soap.client.props file when a SOAP connector is used and in the sas.client.props file when an RMI connector is used. This command is not supported in local mode.
- **Tools used:** wsadmin, Information Center.
- **Problem:** The pre-compile JSP phase failed during deployment of WebSphere Application Server V6 for z/OS. Example 15-1 shows the trace output.

Example 15-1 Trace output

```
Trace: 2005/07/19 15:08:35.410 01 t=8CE828 c=UNK key=P8 (0000000A) Description:
Log Boss/390 Error from filename: ./bborjtr.cpp at line: 932
error message: Compile complete for /jsp/
Errors compiling jsps in
/WebSphere/V6R0M0/DeploymentManager/profiles/default/
wstemp/app/ext/applic.war
Return code from jsp-compilation is: 1
Exception in jsp compile: com.ibm.websphere.management.
exception.AdminException: ADMA0021E: An error
```

```

    occurred in compiling JavaServer
    Pages (JSP) files - applic.war (rc=1)
ADMA6012I: Exception in run com.ibm.websphere.management.
exception.AdminException:
ADMA0021E: An error occurred in compiling JavaServer
    Pages (JSP) files - applic.war (rc=1)
Exception: com.ibm.websphere.management.exception.
    AdminException: ADMA0021E: An error occurred in
    compiling JavaServer Pages (JSP) files
    - applic.war (rc=1)
com.ibm.websphere.management.exception.AdminException:
ADMA0021E: An error occurred in compiling JavaServer
    Pages (JSP) files - applic.war (rc=1)
at com.ibm.ws.management.application.task.
    CompileJspTask.compileWar(CompileJspTask.java:152)
at com.ibm.ws.management.application.task.
    CompileJspTask.performTask(CompileJspTask.java:86)
at com.ibm.ws.management.application.SchedulerImpl.
    run(SchedulerImpl.java:253)
at java.lang.Thread.run(Thread.java:568)

```

- **Solution:** Generally, you see this error when the JAR file that is being used by the compiler is not in a readable or complete state. It could be truncated or malformed in some other fashion.

Check the disk space. The JAR file might be placed in /tmp/app_1052f9ddb0a/ear. Ensure that there is enough free space for that directory and for the WebSphere for z/OS temp space that the application server is using to compile.

- **Problem:** The following error occurs during synchronization when an EAR file is being deployed in WebSphere Application Server for z/OS V6:

EDC5129I No such file or directory.

The configuration synchronization completed successfully but there is an error message in the trace as shown in Example 15-2.

Example 15-2 EDC5129I error

```

Trace: 2005/06/29 08:42:59.948 01 t=AC44F8 c=UNK key=P2 (13007002)
ThreadId: 00000229
FunctionName: com.ibm.ws.management.repository.FileRepository
SourceId: com.ibm.ws.management.repository.FileRepository
Category: AUDIT
ExtendedMessage: BB000222I: ADMR0016I: User AHCPLX/ASCRIT modified
document cells/CELLDM1T/nodes/NODEB1T/servers/IMWEBPR6/plugin-
BSYS-V61-cfg.xml.
file:///Was601DB1T/V6R0/AppServer/properties/xsl/server.xsl; Line #489;
Column #138; Can not load requested doc: /Was601DB1T/V6R0/App
Server/profiles/default/config/cells/CELLDM1T/clusters/CLUSTER3/sib-eng
ines.xml (EDC5129I No such file or directory. (errno2=0x0562
0062))

```

- **Solution:** This message does not indicate any problem in the WebSphere environment. The `sib-engines.xml` file is missing in `/cells/CELLDM1T/clusters/CLUSTERxxx` after migration to V6.0.

See APAR PK07966 for more details:

<http://www-1.ibm.com/support/docview.wss?rs=404&uid=swg1PK07966>

After FixPack 6.0.2 is installed, the message does not appear.

- **Tools used:** None.

- **Problem:** When you run the admin script “\$AdminApp install” the process times out with the error message:

```
SOAPException: faultCode=SOAP-ENV: Client; msg=Read timed out;
targetException=java.net.SocketTimeoutException: Read timed out
```

- **Solution:** In most cases, this exception occurs because the timeout value is too small. To fix this, increase the timeout value specified by the `com.ibm.SOAP.requestTimeout` property in the `soap.client.props` file in the `/WebSphere/V6R0M0/AppServer/profiles/default/properties` directory. The value that you should choose depends on a number of factors such as the size and the number of the applications that are installed in the server, the speed of your hardware, and the capacity provided for the application. The default value of the `com.ibm.SOAP.requestTimeout` property is 180 seconds.

- **Problem:** When deploying an application that has database access, you receive this error message:

```
ADMA8019E: The resources that are assigned to the application are beyond the
deployment target scope. Resources are within the deployment target scope if
they are defined at the cell, node, server, or application level when the
deployment target is a server, or at the cell, cluster, or application level
when the deployment target is a cluster. Assign resources that are within the
deployment target scope of the application or confirm that these resources
assignments are correct as specified.
```

- **Solution:** Apply the latest maintenance to your WebSphere for z/OS environment. APAR PK08164 solves this problem.

- **Problem:** When deploying an application with DB2 access (in our case: TraderDB2 to test DB2 connection) into WebSphere for z/OS V6.0 (network deployment) a dump was thrown with a Java SQL exception with the message:

```
application are beyond the deployment target scope
```

- **Solution:** When enabling trace, we received the information in Example 15-3.

Example 15-3 Trace for DB2 resource access error

```
Trace: 2005/08/16 11:00:00.000 01 t=AC89C0 c=2.C key=P8 (13007002)
  ThreadId: 0000001a
  FunctionName: com.ibm.ejs.j2c.poolmanager.FreePool
  SourceId: com.ibm.ejs.j2c.poolmanager.FreePool
  Category: SEVERE
  ExtendedMessage: BB000220E: J2CA0046E: Method createManagedConnect
ionWithMCWrapper caught an exception during creation of the Managed Conn
ection for resource jdbc/TraderDB2, throwing ResourceAllocationException
. Original exception: com.ibm.ws.exception.WsException: DSRA8100E:
Unable to get a PooledConnection from the DataSource. with SQL State :
42505 SQL Code : -922
    at COM.ibm.db2os390.sqlj.jdbc.DB2SQLJConnection.setError(DB2SQLJConne
    at COM.ibm.db2os390.sqlj.jdbc.DB2SQLJConnection.<init>(DB2SQLJConnect
```

```

at com.ibm.db2.jcc.DB2PooledConnection.<init>(DB2PooledConnection.jav
at com.ibm.db2.jcc.DB2ConnectionPoolDataSource.getPooledConnection(DB
at com.ibm.db2.jcc.DB2ConnectionPoolDataSource.getPooledConnection(DB
at com.ibm.ws.rsadapter.DSConfigurationHelper$1.run(DSConfigurationHe
at com.ibm.ws.security.util.AccessController.doPrivileged(AccessContr
at com.ibm.ws.rsadapter.DSConfigurationHelper.getPooledConnection(DSC
at com.ibm.ws.rsadapter.spi.WSRdbDataSource.getPooledConnection(WSRdb
at com.ibm.ws.rsadapter.spi.WSManagedConnectionFactoryImpl.createMana
at com.ibm.ejs.j2c.poolmanager.FreePool.createManagedConnectionWithMC
at com.ibm.ejs.j2c.poolmanager.FreePool.createOrWaitForConnection(Fre
at com.ibm.ejs.j2c.poolmanager.PoolManager.reserve(PoolManager.java:2
at com.ibm.ejs.j2c.ConnectionManager.allocateMCWrapper(ConnectionMana
at com.ibm.ejs.j2c.ConnectionManager.allocateConnection(ConnectionMan
at com.ibm.ws.rsadapter.jdbc.WSJdbcDataSource.getConnection(WSJdbcDat
at com.ibm.ws.rsadapter.jdbc.WSJdbcDataSource.getConnection(WSJdbcDat
at edu.mayo.registration.amts.datamanager.ManageConnections.getConnec
at edu.mayo.registration.amts.objects.UserRequest.getDbConnection(Use
at edu.mayo.registration.amts.datamanager.UserMapper.accessRacfGroup(
at edu.mayo.registration.amts.businessfacade.ManageAMTSApplication.ch
at edu.mayo.registration.amts.presentation.TeamLogonServlet.performTa
at edu.mayo.registration.amts.presentation.TeamLogonServlet.doPost(Te
at javax.servlet.http.HttpServlet.service(HttpServlet.java:763)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:856)
Caused by: java.sql.SQLException: DB2SQLJConnection error in native
method: constructor: CONNECT 00F30085

```

Analyzing the trace (with Trace Analyzer for WebSphere Application Server and JDBC Trace) and the resource access, we concluded that this condition indicates a security violation. We had to verify the RACF and JDBC data source definitions. In our case, the resource requester's password could not be verified. The user password specified for the data source **jdbc/TraderDB2** was incorrect. We changed the definition and redployed the application.

- **Problem:** During the process of installing the Trade6 application in a WebSphere cell on z/OS v6, the Trade6 install scripts create all the SIBus resources that are needed to support the JMS part of the application, but at the messaging engine start up, the following error appears:

The messaging engine encountered an exception while starting. Exception:
 com.ibm.ws.sib.msgstore.PersistenceException: CWSIS1501E: The data source has
 produced an unexpected exception:java.lang.IllegalStateException: CWSIS1523E:
 Dynamic allocation of database objects in DB2 for z/OS is not allowed.
 com.ibm.ws.sib.utils.ras.SibMessage

- **Solution:** When you use DB2 for z/OS, the messaging engine cannot dynamically create the data store tables. This means that you must manually create these tables using the DDL statements produced by the **sibDDLgenerator** command. You can find instructions at:

http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.wesphere.pmc.zseries.doc\ref\rjm0630_.html

You can redirect the output of the command to a file so that you can submit it to DB2 for z/OS later. One way to do this is to use SPUFI, but before you do this, you must copy the file to an FB80 data set first. As the final step, you must clear the Create tables box (for every messaging engine that uses DB2 for z/OS) from the data store panel of the Administrative Console.

Important: You must create database tables manually for every messaging engine that uses DB2 for z/OS.

15.4 Related references

For information about installing EARs, EJBs, WARs, resource adapter (connector or RAR), and application client modules, search the WebSphere for z/OS Information Center for “ways to install applications or modules.”

For additional hints and tips about how to avoid problems in the deployment phase, search for “Troubleshooting deployment” in the WebSphere for z/OS Information Center.

The Web site for the WebSphere for z/OS information center is:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

See also *Disabling the Deployment Manager HTTP Timeout*, TD101703, available at:

<http://www.ibm.com/support/techdocs>

Refer to “Sample Scripts for WebSphere Application Server” at:

<http://www-128.ibm.com/developerworks/websphere/library/samples/SampleScripts.html>

Archived

Phase 3: Run applications

This chapter provides information about and guidelines for addressing application problems after an application has been successfully deployed to the hosting application server. When we say “deployed successfully,” we mean that the application runtime environment was correctly configured to map to all the components that the application must have to run. This chapter follows the flow of a request/response transaction. At each level in the tier, we identify the problems that can occur and suggest actions that you can take to fix them.

Consult this chapter to solve problems and to plan for and build better solutions. When you are designing the architecture, it is essential to know about the problems, the areas that they can be in, and their frequencies in your system. More time and resources can therefore be allocated to reinforce those areas that need them more during the design and test phases.

We summarize the chapter by revisiting some best practices, methods, and techniques that you can apply to build robust solutions and thus limit application problems.

For more in-depth information about addressing application problems after an application has been successfully deployed, see the WebSphere for z/OS support Web site at:

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

For additional learning and in-depth treatment of WebSphere components and related topics, consult the Information Center at:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb_plugincomp.html

16.1 Request process overview

When an application is perceived as not working properly, you must do preliminary diagnostics to verify reported problems. The first step is to familiarize yourself with the flow of the request/response model as it moves through the layers of the J2EE topology (Figure 16-1). Knowing this helps you pinpoint which problem area (tier) a particular error falls into and helps you quickly discard any area that is not relevant.

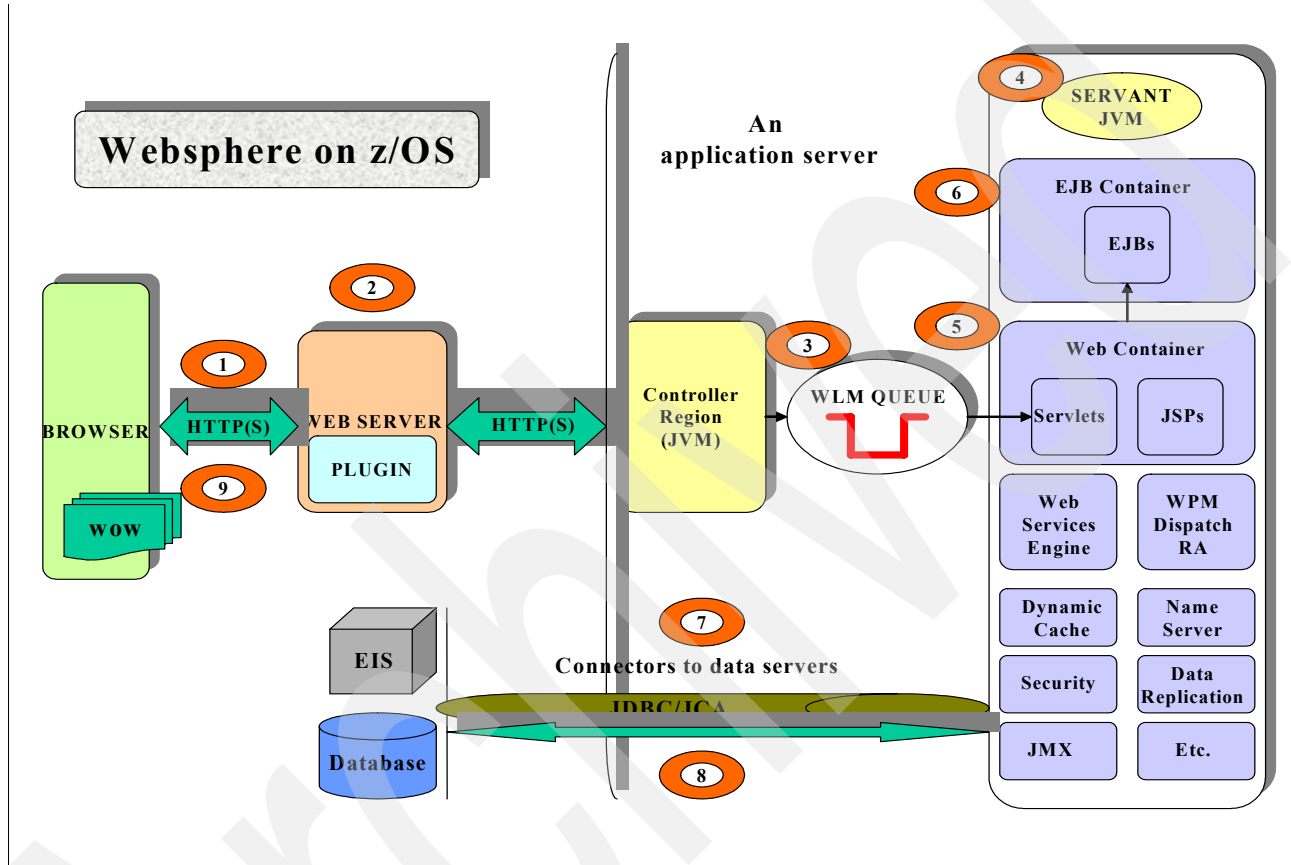


Figure 16-1 Request/response flow

A typical request flow (refer to Figure 16-1) might be:

1. The browser or client issues a request for a resource from the J2EE application.
2. After the request is cleared and authenticated by the z/OS security component, it is routed to the plug-in. The plug-in forwards the request based on directives in the plugin-cfg.xml file that are masked against the requested resource, the transport protocol (secured or non-secured), and the destination Plug-in. This task is done by the Web server plug-in.
3. The controller validates the request for resource access and puts it in the WLM queue. If there is no pre-allocated servant task running to pick the request from the queue and process it, the controller starts one. This scenario can happen if the number of servant tasks has reached the default maximum.
4. The J2EE application server, also called the servant region, loads the components from the required WebSphere class libraries into the runtime environment and invokes the application.

5. The request is routed to a container and a servlet is loaded to service the request. Common events that take place during the servicing of a request are:
 - If the servlet is not already loaded, it is loaded.
 - If the servlet is packaged with “load on initialize,” it is loaded when the server is started.
 - If not, it is loaded when the first request hits.
6. Requests for data from the servlet are classified depending on the types and intended activities (read-only, update). Based on this classification, the appropriate EJBs are invoked. EJBs act as internal application data brokers and shield the applications from the mechanics of having to model and format data every time they require them.
7. Physical data is located on data servers. Database software models data to a predefined design and access patterns (hierarchical, relational, sequential). Organizations choose database software that best fits their processing needs. J2EE connectors “connect” Java applications to data repositories with programming APIs. They do what EJBs do for internal applications that are accessing relational databases: they shield the clients from the mechanics of having to know the attributes of every piece of data needed. The two types of J2EE connectors are:
 - a. JDBC for relational databases (inside a WebSphere environment)
 - b. JCA (implemented as Resource Adapters) for EIS databases (outside a WebSphere environment)

For more information about J2EE connectors, visit:

<http://java.sun.com/j2ee/connector/>

8. Data is retrieved from the enterprise complex and returned to the EJB that is making the request. This data is processed (by the program code) and sent back to the requestor in HTML or XML format.
9. If the request has been serviced with no errors, it is posted back to the browser or client, and the HTTP return code is set to 200.

Knowing how far a request gets is critical in eliminating components that do not need to be addressed. To reinforce this concept, you take a request and superimpose it into your J2EE framework. From there, you can identify the layer or tier where the problem area might lie.

16.2 Model-view-control model for problem determination

J2EE programming adheres to the model-view-control (MVC) standard. We apply this model to analyze areas with common (and a potential for) problems. This enhances the understanding of the WebSphere for z/OS components together with thorough problem analysis and problem determination methodology. For the approach that we discuss in this section, keep Figure 16-1 on page 166 in mind.

16.2.1 Typical problems in the view tier

Based on the MVC model, anything that presents information to the user is located in the view tier, also called presentation tier or user interface layer. This tier is the starting point (where a request is initiated) and endpoint (response) of a Web transaction. It is most often the only relevant contact (interface) that we have with the underlying system that services our business needs.

Components that belong to this tier are browsers, thick clients, Web servers, and Web server plug-ins. See the components with the numbers 1, 2, 3, and 9 in Figure 16-1 on page 166.

The necessary logs for the IBM HTTP Server can be found in the <plugin_install_root>/logs/<web_server_name>/ directory.

The files are:

http_plugin.log	Confirms that the HTTP Server started and initialized
error.log	Records errors within the server
access.log	Records inbound and outbound requests

The Web server plug-in software handles communication between a Web server and the application modules in the Web container. It acts as a somewhat intelligent router for HTTP requests based on directives from its configuration file.

Some typical problems in the order of a request/response flow from the diagram in Figure 16-1 on page 166 are:

- ▶ **Problem:** You cannot get to an application or an application does not work.
 - **Solution:** Follow these steps to analyze and fix the problem:
 - i. Make sure that you do not have any typographical errors in the URI and that all HTTP 4xx codes are client error codes.
 - ii. Search for the HTTP error code at the HTTP Server Information Center or consult the W3 Consortium site:
<http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>
 - iii. Check to see if the Web server has been started. Go to SDSF, select SYSPRINT from the controller job (installation dependent), procstep BBOCTL, and look for this message:
PLGC0057I: Plug-in configuration service is started successfully.
You can also access the top level of the URL from a Web browser. For example:
`http://<web_server_name>/`
Or, you can ping the server:
`ping <web_server_name>`
 - iv. If the Web server does not open a default welcome or error page, check in SDSF to see if the Web server job is running. The name of the job is installation dependent, so verify it with yours. If the Web server is not running, start the Web server by issuing the **START <web_server_procname>** command and analyzing the syslog and job log for its success.
 - v. If the Web server is running, verify access to the application:
`http://<host name>/snoop`
 - vi. If you still have problems, check to see if the application can be accessed directly with the embedded HTTP server in the Web container. Select **Application servers** → **appserver link** → **HTTP transport**. The Web container default ports are listed here, one for non-SSL and one for SSL. Invoke the application again using this URL format:
`http://<host name>:port/snoop`
 - vii. If this did not work (you were not able to access the Web container), then the Web server and its plug-in have problems. Web servers and associated Plug-ins are simple and solidly built components. Once they are running, they work. It is rare that a bad code upgrade was released for the plug-in.

Most problems in this area are related to incorrect changes that were made to the plug-in file or files that were corrupted. With that in mind, incremental back-up of your plug-in configuration is recommended.

In USS, use the **-nostop** and **-nowait** options because it is not necessary to stop the server to back up or restore configuration files:

```
backupConfig <backup_file> [options]
```

For example, you can issue:

```
backupConfig myFile.zip -nostop
```

The **-nostop** option does a backup in place; the server does not have to be stopped for backup to be performed.

```
restoreConfig <backup_file> [options]
```

For example, you can issue:

```
restoreConfig -nowait
```

The **-nowait** option does a restore in place; the server does not have to be stopped for that restore to be performed.

- viii. To verify that the Plug-in is the problem, swap the configuration file that you think is in error with a good backup copy. Use the Administrative Console to apply changes to the Plug-in file. Although it is possible to edit it manually, it is not recommended.
- ix. If this works, then you only have to analyze the two plug-in files for differences to determine where the problem lies. When you know what causes the problem, you can fix it.
- x. You can also regenerate the plugin-cfg.xml from the Administrative Console if you suspect that the copy in the local server is bad. Select **Login** → **Servers** → **Web Servers** and click **Generate plug-in**. If you are running in a network deployment environment, this action replaces the plugin-cfg.xml copy at the node server with the master copy stored at the deployment manager node.

Note: The plug-in is an xml file. The directives that usually are changed when servers are remapped and reconfigured are: VirtualHostGroup, ServerCluster, VirtualHostGroup, and UriGroup.

- xi. After you get past the Web server and the application still does not respond to your requests, there are a few things that you can do to check on the server and its status.

Check for the servant job in SDSF (job name is installation dependent, look for **Procstep=BBOSR**). If it is not running, it does not show. In that case, you must start the application server as follows:

```
START <appserver_proc_name>,JOBNAME=<server_short_name>,  
ENV=<cell_short_name>.<node_short_name>.<server_short_name>
```

For example, we issued:

```
START WS6552C,JOBNAME=WS6552,ENV=CL6552.ND6552.WS6552
```

If you are running in a network deployment environment, you can start the application server from the Administrative Console. Select **Login** → **Applications** → **Enterprise Applications** → **Select Application (check box)** → **Start**.

- xii. If there is a servant job, select its SYSOUT (this log has the error trace turned on by default). Make sure that you do not see any exceptions logged. If the server is

running but the application is not responding, usually that means it has run into an Out Of Memory problem. This can be caused by a limited heap size or other resource constraints. Check which processes use most of your memory and whether this behavior is as expected or caused by configuration mistakes.

- xiii. You can also use the Administrative Console to verify the status of the application server as shown in Figure 16-2. Go to Administrative Console and select **Expand Troubleshooting → Logs and Trace**.

The green arrow confirms that the application server is running. The startup of the application server automatically starts all the applications that it hosts.

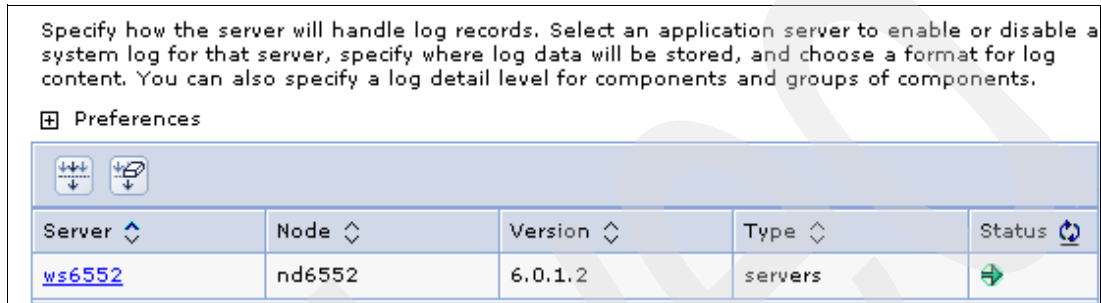


Figure 16-2 Verifying the status of an application server

- **Problem:** The static content that the Web browser is serving up is incorrectly rendered.
 - **Solution:** The source file is being transferred to z/OS in ASCII from workstations. HFS handles files in EBCDIC and any data at the presentation layer must be in ASCII, which might cause some confusion.

With your image, text, or source file available on your workstation, verify that the file has good contents. Use a browser to try to open it. If the file content is good and its association is correct, then it is viewable. If the file is not viewable, just retransfer the file in binary mode.

Tip: Issue the **bin** command to set the transfer mode to binary before transferring files between your workstation and HFS, and vice versa.

- **Problem:** You are experiencing erratic browser response and an inconsistent and quirky interface.
 - **Solution:** Sometimes, older (outdated) browsers also give erratic responses and render contents incorrectly. Check on updates, fixes, and for a list of supported browsers at:

<http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html>

16.2.2 Typical problems in the control tier

After your request moves past the Web server, it arrives at the application server, which is in the control tier (see the components with the numbers 4, 5, and 6 in Figure 16-1 on page 166). This tier controls data that is entering or leaving the enterprise.

The majority of application problems happen in this tier because it is where the WebSphere Application Server engine is located. It contains the two main containers for the Web and EJB components and these two containers make up the runtime environment for all applications deployed. They also act as the broker for all other required services.

Some common problems in the control tier are:

- ▶ **Problem:** You must log in and enter passwords from page to page even though you are using the same function in the same record in the same application.
 - **Solution:** This is the typical behavior of a session affinity problem. The Web container keeps track of sessions with cookies that carry session IDs that are passed back to the browser. Session data is maintained in the application server memory. The Plug-in configuration file is set up to enable session affinity by default and uses the CloneId parameter for session IDs. The parameter can look like this:

```
<Server CloneID="80mn51jkma" ...>
```

The Plug-in generation process adds the Clone ID parameter by default; this is how the Plug-in identifies each application server. When this ID is used, subsequent requests get routed back to the server that generated the session ID. This is the most efficient way of handling session affinity. Other ways to maintain session affinity throughout the servers in a cluster are:

- Persisting session information to a database
- Applying in-memory copying of objects between JVMs (domain replication)

The preferred and recommended way is to keep session objects on the server, the second option.

If you are experiencing a loss of session affinity, check to make sure that cookie writing is allowed in your browser and client firewall software. The browser software has different tabs or sub-menus for enabling cookie writing but, in general, it is in the privacy area.

- ▶ **Problem:** You suspect a program loop.
 - **Solution:** Program loops usually result in a request not responding, or a component hanging, usually followed by a timeout error. In some cases, things seem to work fine, but some tasks in the address space keep consuming system resources without producing a result. You can see this when CPU usage is high but nothing seems to justify this. A dump or trace might be necessary because the log information is not sufficient for determining the cause of the loop or the unusual high resource consumption.

Indications of a loop are:

- A repetitive message from a module waiting for work and nothing being done, such as:
ExtendedMessage: <component> waiting for next server work
- A repetitive message that a module is active and you are able to follow the executed address ranges, but the only thing changing is the time stamp.
- A repetitive message from a module processing work/requests but the thread ID stays the same.

Notice the ThreadID and FunctionName in Example 16-1. They might stay the same, but the trace header line with the time stamp changes if a loop is occurring. There might be several other messages between the repetitions. The shorter the loop cycle, the more likely it is that you can recognize the loop.

Example 16-1 Looping thread

```
Trace: 2005/08/19 21:23:41.232 01 t=7D19C0 c=UNK key=P8 (13007002)
ThreadId: 0000006d
FunctionName: com.ibm.etools.validation.validationbuilder
SourceId: com.ibm.etools.validation.validationbuilder.UserStateRegistry
```

ExtendedMessage: closeUser - found UserPrefs: UserPreferences: nodeName:nd6552, serverName: ws6552, userId: waspd2, refreshRate

If you suspect an application loop or hang, you can use:

- IPCS to format a trace and analyze it for recurring PSW addresses; see 20.1.2, “Viewing CTRACE and JRas data through IPCS” on page 242.
- Use the com.ibm.jvm.svc.dump.Dump utility to identify the thread under which a loop is occurring, the threads contending for resources or involved in a lockout, and a thread waiting for some operation that is external to the server.

See Chapter 20, “WebSphere for z/OS traces and dumps” on page 241.

Before you contact IBM for service, use these tools to identify the particular component or subcomponent that is responsible for the failure. In most cases, the components are the application program code rather than product code from IBM. Present the component name together with the class and method name from the trace to your application development team or IBM (in the case of IBM components). This enables them to fix the code quicker.

- **Problem:** You typed in a URL, such as `<host name>/hitcount`, and you expected an application page, such as that shown in Figure 16-3.

Hit Count Demonstration

This simple demonstration provides a variety of methods to increment a counter value.

Select a method of execution:

- ☐ Servlet instance variable
- ☐ Session state (create if necessary)
- ☐ Existing session state only
- ☐ Enterprise Java Bean (CMP)

Namespace lookup method for EJB:

- ☐ Global Namespace
- ☐ Local Namespace

Figure 16-3 Hit Count application Web page

Instead, you see a screen of source HTML in the browser window (Figure 16-4).

```
<HTML>
<HEAD><TITLE>IBM WebSphere Hit Count Demonstration</TITLE></H1>
<SCRIPT TYPE="text/javascript">
function enableLookupButtons() {
var myButtons = document.getElementsByName("lookup");
for (i = 0; i < myButtons.length; i++){
myButtons(i).disabled = false;
}
}
```

Figure 16-4 HTML source code instead of proper application

- **Solution:** Whenever the document root of the WebSphere Application Server is the same as the Web server document root, the Web server JSP source file is shown as plain text. Using the plug-in directives, you can tell the Web server that a request is to

be handled by the WebSphere Application Server. If an inbound request does not match any entry in these directives, the control returns to the Web server. In this case, the Web server searches the resource requested in the document root. Because the JSP file is stored in the document root, the Web server displays it as plain text.

To avoid the plain text display, move the application server JSP source file outside the Web server document root. When a request comes in with an unknown host header, the plug-in returns control to the Web server and if the Web server cannot find this JSP source file in its root document, it returns an HTTP 404 error message instead.

- **Problem:** The application server can be started from SDSF but does not start from the Administrative Console. The JCL error is:

```
IEF642I EXCESSIVE PARAMETER LENGTH IN THE PARM FIELD
```

- **Solution:** In the log, there is a variable appended to the start command:

```
//STARTING EXEC BB6S001,ENV=DTDCV6.I21A.BB6S001,  
// PARM='-Dwas.status.socket=1927'
```

The status socket being opened by the Administrative Console is actually opened by the node agent. The node agent uses this socket to monitor the progress of the server starting, and reports that progress to the Administrative Console. When you start the server from the MVS command line, it is not being monitored by the node agent, so there is no need to open the status socket (or pass in the `was.status.socket`). So, in this case, everything appears to be working correctly. The best solution here is to remove the hard coded `envvar` from your server JCL.

If you manually added the `_BPXK_SETIBMOPT_TRANSPORT` variable (for a multistack environment), you should move it out of the JCL. In the Administrative Console, select **Environment** → **Manage WebSphere Variables**.

For more information, search for “TcpMultistack” at the WebSphere for z/OS Information Center:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

- **Problem:** The application is deployed in WebSphere for z/OS in a cluster of two or more servers. Each server is on a different WebSphere node. When the application is redeployed, restarting the servers on each node fails. WebSphere for z/OS attempts to restart the controller region before the controller region has finished shutting down.
- **Solution:** Apply the latest PTFs (maintenance) to your WebSphere for z/OS environment. The problem was corrected in version 6.0.2.

16.2.3 Typical problems in the model tier

This is the last tier in the topology (see the components with the numbers 7 and 8 in Figure 16-1 on page 166). This model tier represents the data as the application would see it according to its access authorization (visibility and role). For example, a bank manager would see more details and have more options in a banking application account screen than a bank teller would.

If data is modeled properly, it has a built-in access control functionality (for example, a bank customer does not have the screen option, and therefore no access to it) and it maintains the state of the data (persistence), so that information such as account balances are accurate in real time.

The most common problem in this tier is that the application cannot access data. This condition has more than one possible cause:

- The user of a request has insufficient authorization granted for a certain resource.

If this is the case, you almost certainly find an authorization failure message (Example 16-2) in the job log. Request appropriate authorization from the security administrator so that the application can run with sufficient privileges.

Example 16-2 Sample failure message

```

ICH408I USER(WASPD2 ) GROUP(SYS1 ) NAME(USER1) /u/waspd2/.sh_history CL(FS0BJ )
FID(000000000000000004C03000000190000) INSUFFICIENT AUTHORITY TO OPEN
ACCESS INTENT(RW-) ACCESS ALLOWED(GROUP ---) EFFECTIVE UID(0000003174) EFFECTIVE
GID(0000000000)

```

- The connection to the back-end EIS or data sources has problems.

There are two types of connectors to back-end data:

- A JDBC driver that is embedded with WebSphere if you are using DB2
- EIS JCA connectors that are provided by IBM if you are using IMS or CICS.

Regardless of which connector you use, they all have traces and logs that can be collected to help diagnose errors.

An incorrect configuration usually results in an exception logged in the SYSLOG when the resource is being accessed. A timeout exception occurs when the back-end system server (EIS or DB2) is unavailable.

A JDBC trace is useful for diagnosing problems in the DB2 SQL for Java and JDBC (SQLJ/JDBC). The output goes to an HFS file that is specified in the JDBC properties file. JDBC trace information shows Java methods, database names, plan names, user names, or connection pools. Use the Administrative Console to check the configuration information and test the connection for DB2 JDBC:

- Go to **Login** → **Resources** → **JDBC Providers** → and look for the link with the name of the database. The center pane displays the configured JDBC provider as seen in Figure 16-5.

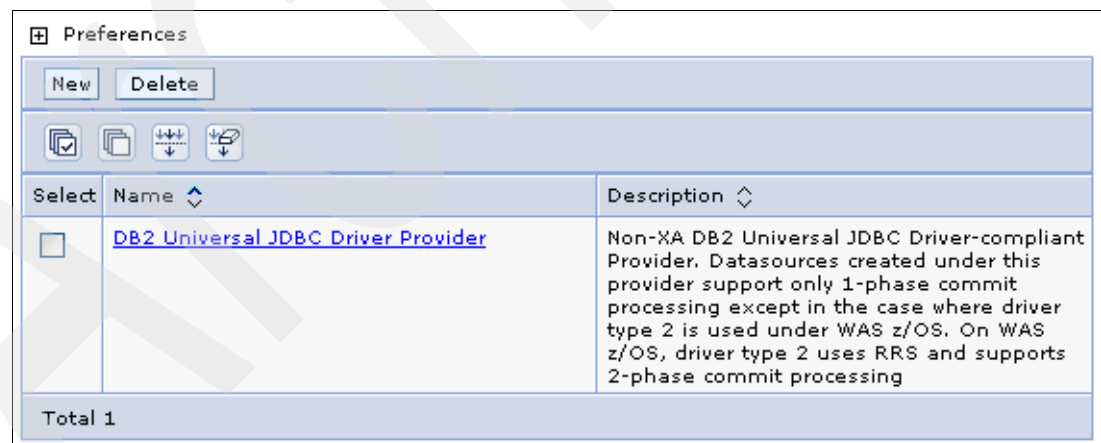


Figure 16-5 JDBC Data source configuration

- Click the link and verify the name and properties of the database.
- Test the connection to verify that the application can access the data source.
- Ensure that you have the right data source name and properties configured in the Administrative Console.
- Also check the data source names in the resources.xml configuration file.

IMS and CICS (two EIS products) also produce their own traces and logs. These subsystems very likely run in their own LPARs. Contact the administrator to get the traces and logs that are required for further analysis or ask them for help.

If your application uses EIS resource adaptors:

- a. Contact the EIS administrator to verify that the subsystem is up and available because there is no direct way to test the connection to an EIS resource from the WebSphere Administrative Console.
- b. Go to the Administrative Console and select **Login** → **Resources** → **Resource Adapters**.
- c. Drill down to the resource name link as you would with the JDBC providers. Verify the configuration properties (given by the administrator or developer of the application) such as spelling of resource names, class path information for libraries, and security information.

16.3 Problem avoidance

Good development standards and sound implementation methodology reduce the amount of time and resources being spent on problem resolution.

A good problem management system is indispensable because it can be used:

- ▶ To prevent problems from being introduced into steady state
- ▶ To feed solutions to problems found in steady state back into the repository of test cases.

It is outside the scope of this book to discuss design, build, and test philosophies in detail. Instead, we briefly highlight best practices for building your system and controlling its deployment. Your problem domain is a function of your design. Poor design leads to poor usability. However, knowing the principles to pay attention to in the construction phase of a system can help prevent poor design.

16.3.1 Designing, coding, and testing

Designing, coding, and testing best practices that can help you avoid problems are:

- ▶ Understanding the requirements
- ▶ Spending time on design

Understanding the requirements

The better you understand the requirements of a solution, the better your design. That helps you trap and plan for conditions that are not considered normal operating procedures. By foreseeing these abnormal conditions, you can design better error and warning mechanisms that help speed the debugging process. The mechanisms can range from meaningful messaging and to even possibly and programmatically recovering from these conditions.

Spending time on design

You should spend time at the beginning of the design phase to come up with good design, build solid and comprehensive test cases, and follow them through. Proper testing is better than quantity testing, and, of course, proper design is always better than proper debugging.

16.3.2 Change control

Good change control management can help you avoid problems. You should have robust change control procedures and enforce them. A robust change control process can only be as good as its implementation. Do not take shortcuts when implementing changes no matter how small a change it is.

Always have a log of the changes that are going into the production systems. Hold regular audits of changes and verify all changes. If a change was implemented on a certain date, then follow up with its result. Did it or did it not meet the requirements?

Keeping access logs to systems also helps identify unauthorized accesses, security problems, and so forth.

Note: Keeping change logs is a best practice, but do not use them as the sole source for identifying or considering where the root cause of a problem is. You must consider the problem from all aspects.

Phase 4: System run time

In this chapter, we present an overview of the various system components and subsystems that make up a typical system run time for WebSphere for z/OS V6. The system run time for WebSphere for z/OS V6 is fairly similar to WebSphere V5 for z/OS. So, if you have a good understanding of the system run time for WebSphere V5 for z/OS, you are well positioned to identify and solve problems in this phase.

We look at some key components and the underlying environment that provides the structure for running applications. We then give a brief description of common processes and some general sources of problems.

We also provide hints and tips that can help you avoid problems in a checklist. In the last sections of this chapter, we provide a list of a few common problems that are related to the system run time and to the applications deployed to this runtime environment.

We assume that the user application has been thoroughly tested either in the WebSphere test environment of the integrated development environment (IDE) or in WebSphere V6 on the distributed platform. We also assume that the application has been deployed without problems to the WebSphere for z/OS V6 environment.

Note: We assume that WebSphere for z/OS V6 has been installed and configured correctly and also that the latest maintenance has been applied. We focus on problems that you might encounter in the WebSphere for z/OS V6 runtime servers when running your applications.

17.1 The WebSphere for z/OS V6 runtime environment

WebSphere for z/OS V6 is a runtime environment implementation for the J2EE 1.4 application programming model. It also supports the J2EE 1.3 and J2EE 1.2 application programming models.

z/OS (or z/OS.e) Version 1 Release 4 is the prerequisite minimum release level for running WebSphere for z/OS V6. WebSphere for z/OS V6 interacts with a number of components of the z/OS operating system in ways that are similar to those of the previous versions of WebSphere for z/OS. The following z/OS components are prerequisites for WebSphere for z/OS V6:

- ▶ z/OS Communication Server (TCP/IP)
- ▶ z/OS USS and HFS
- ▶ Security Server (like RACF or another SAF-compliant security product)
- ▶ System Logger
- ▶ WLM (in goal mode)
- ▶ Resource Recovery Services (RRS)
- ▶ System SSL required when using SSL

Additional optional components can be important parts of the complete runtime implementation necessary for supporting your application environment. Such optional components typically include:

- ▶ IBM HTTP Server for z/OS
- ▶ Lightweight Directory Access Protocol (LDAP)
- ▶ DB2 Universal Database™ (UDB) for z/OS Version 7 or higher
- ▶ CICS TS V1.3 or higher
- ▶ WebSphere MQSeries® for z/OS, V5.3.1 or higher
- ▶ WebSphere MQ Integrator Broker for z/OS, V2.1 or higher

JDK 1.4 is now included with the WebSphere for z/OS V6 code base and also part of the WebSphere for z/OS maintenance stream. This is different from previous versions of WebSphere for z/OS where you had to obtain and install the JDK separately.

A new messaging engine is included in the WebSphere for z/OS V6 run time to provide a pure Java JMS provider. This function adds a new address space to the WebSphere for z/OS V6 runtime environment. This address space is called a control region adjunct (CRA) server. The CRA address space is started by WLM only when you are using messaging functionality. This new function helps avoid the possible installation of another product with its own prerequisites. The JMS messaging engine runs completely inside the JVM of the CRA.

Another new function in WebSphere for z/OS V6 is the option of administering your Web Server from the Administrative Console. From the Administrative Console, you can:

- ▶ View the Web server logs.
- ▶ View and edit the Web server configuration file (httpd.conf).
- ▶ Start and stop the Web server.
- ▶ Automatically build the plug-in configuration file (plugin-cfg.xml).

Figure 17-1 on page 179 shows the WebSphere for z/OS V6 runtime environment, including the underlying z/OS infrastructure and also some optional z/OS subsystems. Also refer to Chapter 14, “Phase 1: Installation, configuration, and migration” on page 141, for additional information about this structure.

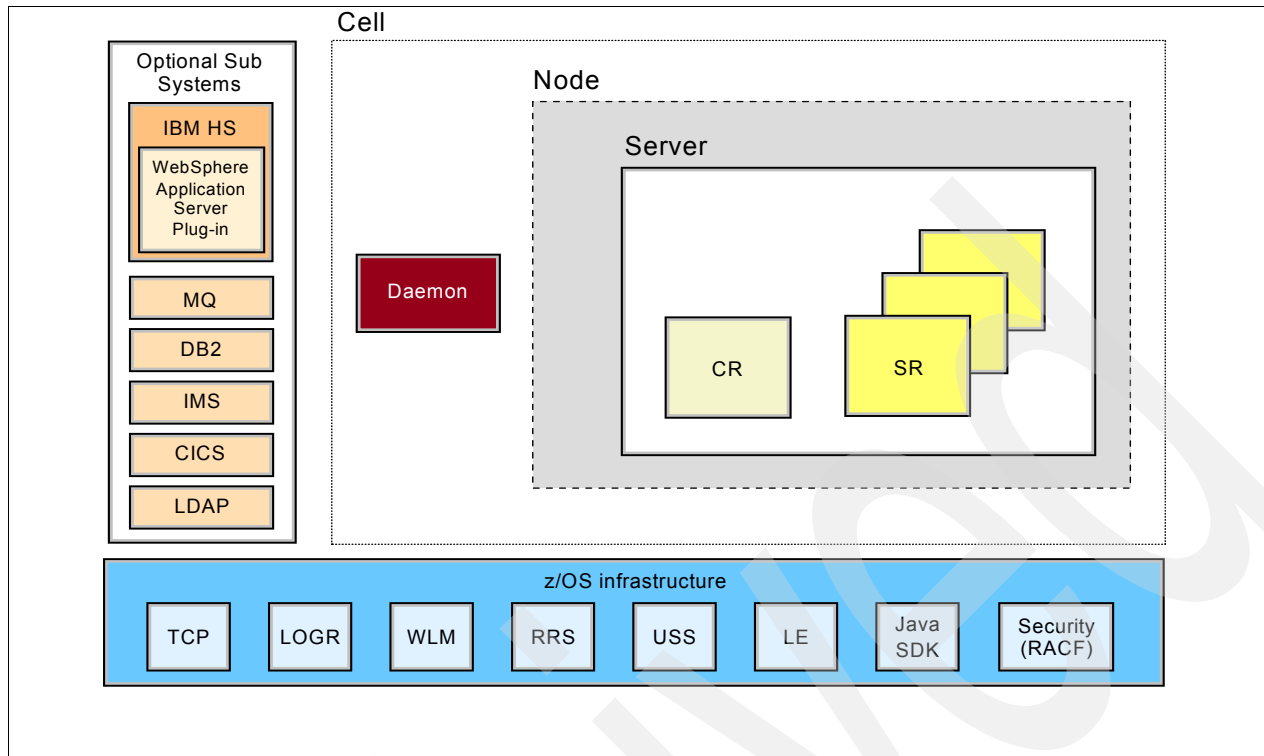


Figure 17-2 Stand-alone application server configuration

WebSphere for z/OS V6 also has a base application server configuration. It is called a stand-alone application server configuration because that is what it is. You can no longer add additional servers to the node like you did with WebSphere for z/OS V5. As shown in Figure 17-2, the stand-alone application server configuration is a node and a cell, and it has a daemon server. The rule is *one daemon per cell per z/OS image*.

In WebSphere for z/OS V6, the stand-alone application server configuration supports multiple servant regions (“SR” in the illustration). This was not supported in WebSphere for z/OS V5. You could only have one servant region in the WebSphere for z/OS V5 base application server configuration.

17.2 Problem categories in the runtime phase

Problems in this runtime environment can occur in many places. Problem determination is by no means a simple task in such an environment. We have categorized the problem areas as:

- **z/OS infrastructure problems**

The WebSphere for z/OS V6 runtime environment depends on the prerequisite z/OS components. Conscientious planning is necessary. Include experts from various areas such as RACF (or another SAF-compliant security product), TCP/IP, and WLM on your WebSphere for z/OS V6 administration team. A healthy z/OS operating system and subsystem infrastructure is necessary for WebSphere for z/OS V6 to run successfully. This helps avoid time consuming problem determination sessions later in your administration process.

- **Problems that are related to optional subsystems**

One or more optional subsystems might be present in your runtime environment. Most likely, some of your user applications that are deployed to WebSphere for z/OS V6 are

going to be accessing one or more of these subsystems. These subsystems can be in the same LPAR as your WebSphere for z/OS V6 run time (see Figure 17-1 on page 179), but this is not required. They can even be on a separate server such as an LPAR on a different zSeries platform. These subsystems (for example, CICS TS, DB2 UDB for z/OS, or IMS) normally provide vital back-end functionality for your user applications. When WebSphere MQ is involved, this subsystem might provide back-end and front-end functionality to your applications.

Problems connecting to the back-end subsystems usually create an application timeout or some incorrect output from the application. Problems in the subsystems themselves often impact the response times or the performance of your WebSphere for z/OS V6 user application. In many of these cases, the WebSphere for z/OS V6 system administrator works with subsystem experts to resolve these types of problems.

► WebSphere for z/OS environment problems

The WebSphere for z/OS environment basically consists of an administrative part and the application server run time part:

- The administrative part includes several tools like the Administrative Console, the WebSphere administrative scripting program (wsadmin), administrative commands, and possible administrative programs. Refer to the WebSphere for z/OS Information Center for further information and description. In addition to the tools, the deployment managers and the node agents are administrative components.

For the administrative part to work without problems, these tools must all be installed correctly and the components must be configured correctly and up and running. For example, in a network deployment configuration, the Administrative Console application, the deployment manager, the node agent, and the configuration repository (HFS or zFS) must all be present and working correctly for your administration to be carried out. Any mismatch between the tools and the components can result in your administration failing to work.

- The application server run time (either a stand-alone configuration or a network deployment configuration) consists of, at minimum, a daemon, a controller region, an optional CRA, and one or more servant regions for each controller region. Inside each servant region is a JVM where the J2EE applications are executed (see Figure 17-1 on page 179 and Figure 17-2 on page 180).

When you are analyzing problems in the WebSphere for z/OS V6 application server run time, it is important to understand the functions of each component. It is also very important to understand how the components interact with each other. Problems arising in the runtime part should be approached in the standard way:

- What is the symptom?
- When does the problem occur?
- Has anything been changed in the runtime part recently?
- Has a new version of the application been deployed recently?
- Look at logs starting with the most likely component
- Look at the error log, look at SYSOUT, and look at SYSPRINT

17.3, “Understanding your own runtime configuration” on page 182, describes this process in more detail.

For information about troubleshooting, search for “Troubleshooting and support” at the WebSphere for z/OS Information Center:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

17.3 Understanding your own runtime configuration

To be effective at problem determination, you need a good understanding of your current WebSphere for z/OS V6 runtime environment. Specifically, you need to understand:

- ▶ What system configuration files exist and what are their content
- ▶ What logs and error logs are available
- ▶ What variables and properties are available and what values they can take
- ▶ What traces and trace options are available
- ▶ What tools are appropriate to use for my current environment and symptoms
- ▶ Which functions, administration, and problem determination tasks you can perform with the Administrative Console

A good start is to take a close look at the job log for all your control and servant regions. Specifically, when these regions start, you can learn quite a lot about your configuration by examining the messages being issued.

Each WebSphere for z/OS V6 server has a configuration file called `was.env`. This file is a parameter file that contains important configuration settings. Each time you start your runtime environment servers, you must point to this file. The file allows the non-Java portion of the server to open, and it is located far down the HFS directory tree.

If you are planning to enable global security in your runtime environment, it is necessary to have a good understanding of the implications. Refer to *WebSphere Application Server for z/OS V5 and J2EE 1.3 Security Handbook*, SG24-6086. This relatively large publication contains a lot of good information. With your security administrator, review the subjects that apply to your environment. Troubleshooting security issues is no different from troubleshooting other kind of problems, but sometimes it is harder to identify the cause of the problem.

17.4 Troubleshooting tips for the runtime environment

When you are troubleshooting problems in a runtime environment that previously ran without any error symptoms, it is important to use a structured approach.

The most common sources of problems in this environment include:

- ▶ Recent changes
 - Was any z/OS maintenance performed lately?
 - Was any maintenance to any of the z/OS subsystems performed?
 - Were any new versions of the production applications newly deployed?
 - Was any maintenance performed for WebSphere for z/OS V6 lately?
- ▶ Recent operational activities
 - Are you running any diagnostic tools or traces?
 - Have you activated any performance monitors?
 - Have you restarted any of the z/OS components or subsystems?
- ▶ Changes in the volume of traffic
 - Has the number of WebSphere for z/OS V6 transactions increased?
 - Has the resource utilization increased?
 - Is there a lack of processor or memory capacity for the WebSphere for z/OS V6 LPAR?

Following such an approach leads you to the most likely cause of the problem in the shortest amount of time.

The WebSphere for z/OS V6 runtime environment differs from installation to installation. However, the initial troubleshooting process is almost identical and the authors recommend the following simple approach when you have identified a symptom:

1. Review the syslog, the job log for the CR and the SR, and the WebSphere for z/OS V6 error logs. Identify error messages.
2. If you can associate an error message directly with your initial symptom, take action as indicated by the error message.
3. If the error message cannot be directly associated with the symptom but requires you to obtain more diagnostic information, follow the indication in the error message.

If you still cannot identify the error messages that are associated with your symptom, we recommend that you use the flow charts for the symptoms that we have identified in Part 2, “Problem symptoms and their resolutions” on page 39, to find the root cause of the problem and solve it.

17.5 Security issues and problems

Security issues and problems are normally encountered and sorted out during the installation and customization phase, during the application deployment phase, or during the run application phase. However, problems can appear later during run time because of new functions that are being exercised or defects that were not discovered during the application testing period.

Troubleshooting security issues involves looking for symptoms in messages, in the controller region, the CRA, and servant region job logs and in the SYSLOG. If you cannot find any symptoms, turn on tracing and search the traces for symptoms. Additionally, you can use dumps. Security problems usually carry one or more of these symptoms:

- ▶ Security violation
- ▶ Access denied
- ▶ Permission bits problems
- ▶ Invalid user ID or password
- ▶ Not enough authority for the user ID

You should review security logs and security traces with your security administrator. Sometimes a dump of the security system is also necessary. Refer to *WebSphere Application Server for z/OS V5 and J2EE 1.3 Security Handbook*, SG24-6086, for more information.

17.6 Problem avoidance checklist

Because the WebSphere for z/OS V6 runtime environment and configuration differs from installation to installation, it is not possible to create a checklist that satisfies every configuration option. However, this checklist offers the following common and important problem sources:

- ☐ Software maintenance

Establish a procedure for applying software maintenance to your system if you do not already have one. This procedure involves testing the maintenance that is being introduced to a test system image prior to putting it in the production system. The procedure must include the base z/OS system, its components, all z/OS subsystems, and

WebSphere for z/OS V6. Although many problems are caused by software components other than WebSphere for z/OS V6, many product defects are also found and maintenance for those defects is made available. As a standard procedure, verify that your current runtime environment satisfies all the hardware and software prerequisites for the maintenance to be performed. Also verify that, when you perform maintenance for WebSphere for z/OS V6, you do not introduce any inconsistency between the z/OS load libraries and the HFS.

❑ Change management

In the same context, establish a total change management procedure for your environment. This procedure should outline the proper processes for making any kind of changes to your environment. This prevents problems that occur as a result of one group or person in the organization introducing changes to the environment that others were not aware of. With a procedure to eliminate these occurrences, the chances of unexpected problems is reduced.

❑ The test environment

Ideally, the test environment should be configured exactly like the production environment. However, this is sometimes not possible. When performing tests, the software maintenance level should be identical to your production environment. If the test environment and the production environment are kept in sync, you can easily do problem determination in the test environment if a problem occurs in the production environment. This is an advantage because it does not impact your production environment.

One of the reasons the authors recommend a testing environment with a configuration identical to the production environment is because it allows you to test performance without affecting your production environment.

❑ Application testing

Testing is the best strategy for preventing problems from occurring in your WebSphere for z/OS V6 production runtime environment. A detailed testing strategy should be in place for all applications and it should be followed every time a new version of the application is being introduced into the production environment. This procedure must include both functional testing and performance testing. Prior to deploying an application into your WebSphere for z/OS V6 runtime environment, the authors recommend that it be thoroughly tested in the WebSphere test environment of the IDE or in WebSphere V6 in a distributed environment.

❑ Monitoring

A good monitoring strategy can help you identify problems before users experience them. WebSphere for z/OS V6 includes an enhanced Tivoli Performance Viewer that is accessible from the Administrative Console. To use Tivoli Performance Viewer, you enable the Performance Monitoring Infrastructure, log on to the Administrative Console, select your application server, and select the PMI link. Tivoli Performance Viewer can give you clues about possible performance bottlenecks.

It is also wise to monitor the WebSphere for z/OS V6 application server logs. The SystemOut and SystemErr logs for each application server should be monitored. Informational messages, warning messages, and error messages are in these logs.

❑ Operational procedures

To avoid or reduce operational issues, make sure that you have created and documented your own start and stop procedures. They must clearly describe the sequence and commands for all components and subsystems. This can serve as a requirement specification for the automation of the operational process. Ideally, your operation should be fully automated to reduce human error.

❑ Large HFS

Ensure that you allocate a large HFS for the WebSphere for z/OS V6 configuration. Leave a large spare space for your file system so that you have room for adding new applications, upgrading to new application servers, upgrading to a new WebSphere for z/OS V6 maintenance level with a new version of the Administrative Console application, and so on.

❑ Network deployment configuration

In your network deployment configuration, ensure that the deployment manager and the node agents are running prior to any application deployment or configuration changes. This helps ensure that the changes are effectively synchronized across all nodes.

❑ Attributes and properties verification

For each server, using the Administrative Console or the job logs, verify the following attributes and properties:

- `protocol_http_timeout_persistentSession`
Use the default values unless you know that your application requires different values.
- `protocol_http_timeout_input`
Use the default values unless you know that your application requires different values.
- `protocol_http_timeout_output`
Use the default values unless you know that your application requires different values. This timeout might also cause abend EC3-0413000n in the case of a long running JSP compilation request.
- `transaction_maximumTimeout`
The default is 300 seconds. Check the value and increase it.
- `wlm_dynaplenv_single_server=1`
Use this value if you do not want to have multiple servant region instances per application server. Otherwise set it to 0.
- `wlm_maximumSRCount=1`
If you need multiple servant region instances to be started, make sure that this is set to a desired value.
- `wlm_minimumSRCount=1`
This determines how many servers to start as a minimum.
- `ras_trace_defaultTracingLevel=1`
For new applications, the recommendation is 1 and:
`ras_trace_outputLocation=BUFFER SYSPRINT`

❑ Correct settings

For each server, use the Administrative Console or the servant job log to verify that the following attributes and properties are set correctly:

- Verbose garbage collection is enabled.
Verbose garbage collection for the application server JVM is turned on. Turning on verbose garbage collection is not very costly and can help to monitor possible Java memory leaks.
- JVM maximum heap size is set.
You can set this value to 256, `-Xmx256m`

- JVM minimum heap size is set.
-Xms256m or set it to the same as for -Xmx
- -Xdebug is turned off.

17.7 Typical problems

In this section, we list a few common problems in the WebSphere for z/OS V6 runtime environment. These problems were discovered by WebSphere for z/OS V6 users and we document the solutions for them:

- **Problem:** After applying maintenance to WebSphere for z/OS V6, runtime servers do not start or they abend while running applications.

- **Solution:** Actually, the symptoms here can vary. We have seen this problem being caused by a mismatch in code levels between the z/OS load libraries and the HFS files. Typically, this happens if maintenance is applied to a service HFS and then later not copied to the production HFS.

This problem can also occur if you do not remove old copies of WebSphere load modules from LPA or you do not refresh LLA.

- **Problem:** After you perform maintenance to WebSphere for z/OS V6, applications do not respond as expected. Changes to the plugin-cfg.xml file are not being picked up.

- **Solution:** This can be caused by using the wrong plugin-cfg.xml file. After maintenance, files might be overwritten or the pointers to these files might have changed. Make sure the correct plugin-cfg.xml file is located at the path specified in the httpd.conf file.

- **Problem:** The instructions for empty managed node are wrong.

After an attempt to federate an empty managed node, it was discovered that the file ownerships were wrong; they were owned by UID=0. The federate completed rc=0, but there were RACF error messages where the node agent was unable to access parts of the HFS. It appears that the instructions that were generated for empty managed node creation and federation were wrong, because they indicated that a UID=0 user ID should be used, when in fact, the admin user ID probably should be used, such as when a non-empty node is federated.

- **Solution:** The soap.client.props file for the managed node was not being properly updated with the user-provided values from the Customization Dialog, specifically the section defining the SSL Configuration. The soap.client.props file can be fixed so that it is properly updated with the user-provided values if you apply the latest maintenance. APAR PK04464 is associated with Service Level (Fix Pack) 6.0.2.1 (Build Level cf10533.10) of WebSphere Application Server for z/OS V6.0.1.

- **Problem:** In the Administrative Console on troubleshooting, the users selected **Configuration Problems**. They found multiple entries that they believed were not caused by configuration work that they had performed and as a result, they thought there might be a problem with the product.

The message was:

```
CHKW2062E: The executable target of a java process definition is absent.
cells/CELLDM1T/nodes/NODEB2T/servers/nodeagent/server.xml
```

It occurred multiple times for server.xml documents.

They also received:

CHKW3706E: Validation of cells/CELLDM1T/nodes/NODEB1T/serverindex.xml failed with exception java.lang.NullPointerException.
cells/CELLDM1T/nodes/NODEB1T/serverindex.xml

They only received it once.

Another message was:

CHKW2130W: The server Network Deployment Server has no configured transaction service. cells/CELLDM1T/nodes/NODEDM1T/servers/dmgr/server.xml

- **Solution:** All of these CHKW messages are being issued by the configuration validator. The CHKW2062E messages are usually the result of Java process definitions that have blank fields, specifically the executable arguments field in the process definition for the JVM of a server. These messages communicate nothing except that the field is blank. They are benign. There is a good argument for why these messages are being generated, however.

The next message is the CHKW2130W. This is a warning message (note the W suffix on the message code) that appears to check the value of the enable attribute for the TransactionService element in the server.xml. This warning is benign and can safely be ignored.

For the last error, it is difficult to determine what is occurring in that CHKW3706E message. However, this is simply an error in the validation of the WebSphere document repository, and does not necessarily represent any runtime error in WebSphere that would affect operation.

- **Problem:** When you attempt to use the wsadmin script, you are not able to invoke the script or the error returned by wsadmin does not seem to apply to the command you entered. For example, you receive a WASX7023E, stating that a connection could not be created to host “myhost,” but you did not specify -host myhost on the command line.

- **Solution:** You are either in the wrong session and therefore not authorized to run wsadmin in this particular session, or you might have entered the wrong commands.

If you are not able to enter wsadmin command mode, try running **wsadmin -c "\$Help wsadmin"** for help in verifying that you are entering the command correctly.

If you can get the wsadmin command prompt, enter **\$Help help** to verify that you are using specific commands correctly.

Check your syntax and spelling and enter the command again. Also keep in mind that wsadmin.traceout is refreshed (existing log records are deleted) whenever a new wsadmin session is started.

If the error returned by wsadmin does not seem to apply to the command you entered, examine the properties files that are used by wsadmin to determine what properties are specified. If you do not know what properties files were loaded, look for the WASX7326I messages in the wsadmin.traceout file; there is one of these messages for each properties file that is loaded.

The wsadmin commands are a super-set of JACL, which is a Java-based implementation of the TCL command language. For details about JACL syntax beyond wsadmin commands, refer to the TCL developers site at:

<http://www.tcl.tk>

For specific details relating to the Java implementation of TCL, refer to the Web site at:

<http://www.tcl.tk/software/java>

- **Problem:** After you apply the cumulative fix cf20523.06 to your WebSphere for z/OS V6.0.1 system, you notice that an old problem regarding SMF had resurfaced. Separator

values were missing in the SMF 120 subtype 5 field, `AMCName`, that is used to parse the application, module, and class name.

For example, what used to appear as:

```
ECperfEAR::OrdersJAR.jar::OrderLineEnt
```

Now appears as:

```
ECperfEAROrdersJAR.jarOrderLineEnt
```

The APAR to fix the old issue was PQ85314.

- **Solution:** In WebSphere Application Server V4.0.1 for z/OS and OS/390, the Application-Module-Component (AMC) name was represented in SMF as the string `ApplicationName::ModuleName::ComponentName`, where each sub-field was separated by “::”. However, in V5, the AMC name returned with SMF recording is `ApplicationNameModuleNameComponentName`, where there is no separation of each sub-field.

The Application-Module-Component value that is returned by SMF should follow the `ApplicationName::ModuleName::ComponentName` format and can now be tokenized with “::” separating each sub-field. This value is now compatible with WebSphere Application Server V4.0.1 for z/OS and OS/390 SMF tooling. Apply the latest maintenance. APAR PK07137 is associated with Service Level (Fix Pack) 6.0.2.1 (Build Level cf10533.10) of WebSphere Application Server for z/OS V6.0.1.

- **Problem:** When you are running an application, a Storage Allocation Exception error occurs in the WebSphere for z/OS error log:

```
WAS Z fullMaterializedLobData give storage error- Websphere Application Server  
Allocation Error
```

- **Solution:** The current available resolution is either to use LOB locators versus fully materialized data (that is, setting the `fullyMaterialLobData=false`) or to define the target LOB columns with maximum sizes that the client-side system can allocate storage for.

- **Problem:** Server abends cause the Administrative Console session to become invalid. The following message is issued in the log:

```
BB000232W REQUEST FOR CLASS NAME REMOTEWBCONTAINER METHOD NAME HTTP REQUEST  
FROM IP ADDR HAS TIMED OUT
```

- **Solution:** If the server abends are caused by a timeout, you can change the `protocol_http_timeout_output_recovery` variable. Set the variable to `protocol_http_timeout_output_recovery=SESSION` so that there is no abend when the session timeout happens. This prevents invalidation of the existing session and allows the Administrative Console to be used immediately without waiting for the server to finish initialization.

If the timeout occurs because of server performance issues, then capture a console dump of the server and determine the cause of the poor performance:

```
DUMP COMM=(description of problem)
```

Reply to dump WTO, where `SERVERPROC` is the name of your WebSphere Server address spaces:

```
JOBNAME=(OMVS,Serverproc),DSPNAME=('OMVS'.SYSZBPX1,'OMVS'.SYSZBPX2),  
SDATA=(CSA,GRSQ,LPA,NUC,PSA,RGN,SQA,TRT,SUM)
```

Use IPCS to diagnose the dump and identify the cause of low performance. Chapter 13, “WebSphere for z/OS performance analysis” on page 117, and Chapter 12, “High CPU utilization” on page 109, can help you analyze and resolve performance problems.

- **Problem:** When an application in WebSphere for z/OS V6 (network deployment) was running, an application accessing DB2 resources timed out. The server address space did not respond anymore.
- **Solution:** We enabled trace and received the messages in Example 17-1.

Example 17-1 Messages after WebSphere for z/OS becomes unresponsive

```
Trace: 2005/08/15 13:34:14.427 01 t=AC2CF0 c=UNK key=P8 (13007002)
  ThreadId: 0000003a
  FunctionName: com.ibm.ws.runtime.component.ThreadMonitorImpl
  SourceId: com.ibm.ws.runtime.component.ThreadMonitorImpl
  Category: WARNING
  ExtendedMessage: BB000221W: WSVR0605W: Thread "WebSphere:ORB.thread.
pool t=00ac6cf0" (00000020) has been active for 700000 milliseconds and
may be hung. There is/are 9 thread(s) in total in the server that may
  Trace: 2005/08/15 13:34:14.428 01 t=AC2CF0 c=UNK key=P8 (0000000A)
  Description: Log Boss/390 Error
  from filename: ./bborjtr.cpp
  at line: 901
  error message: BB000221W: WSVR0605W: Thread "WebSphere:ORB.thread.
pool t=00ac6cf0" (00000020) has been active for 700000 milliseconds and
may be hung. There is/are 9 thread(s) in total in the server that may
```

When using the Trace Analyzer for WebSphere Application Server, we realized that the application ran out of JDBC connection threads. The application in WebSphere for z/OS could not process the next request as there were no more threads available. The request stayed in waiting condition. The application server issued a warning message about a potential hang.

This message is common when the connection between WebSphere for z/OS and DB2 database runs into problems, such as authorization failures or thread limitations for new DB2 resource connections. We increased the number of threads available for DB2 connections.

- **Problem:** The connection between z/OS and a Java-based Connector to WebSphere MQ does not start.
- **Solution:** First verify that WebSphere MQ is running and test to see if the listener is running and connected to the right port. The port must be the same as in those of your counterpart definitions for WebSphere MQ for z/OS. Use the following command to verify:

```
ps -ef | grep runmq1sr
```

If the name of the MQ manager that you suspect has trouble does not appear in the list, it is not started. Check the port numbers in the display. You must use a different port for every Queue Manager. Try to start the listener with the **runmq1sr** command if the ports are fine.

If you see a listener with no port information, this listener is using the default port 1414. In a system where more than one MQ listener is active, this has an impact on the port definitions of the counterpart as well. To start the listener with a different port, you must provide the port number with option **-p**, such as

```
runmq1sr -m QM1 -t tcp -p 1500
```

Note: When you are trying to connect from two different Queue Managers to a local queue, you might successfully transmit one or more messages only from the first Queue Manager. If that happens, it means that the local message counter of your local queue differs from the message counter of the second remote Queue Manager on the other side. In this situation, no further messages are accepted until the counters on both sides have been reset or have adjusted to each other.

- **Problem:** You find the following error message:

CTG9630E: IOException occurred in communication with CICS

- **Solution:** This is a configuration error. Review the JNI trace for the root cause. If you find:

- RRS register return code 0x300

The CICS Transaction Gateway has not been properly configured to use RRS. Consult your CICS or system administrator about changing the CICS Transaction Gateway configuration for RRS.

- EXCI reason code 403

The CICS Transaction Gateway is unable to contact the target CICS server because of an invalid pipe name. Consult your CICS or system administrator about correcting the pipe name in the CICS Transaction Gateway configuration.

- **Problem:** You find the following message:

CTG9631E: Error occurred during interaction with CICS. Error Code=ECI_ERR_NO_CICS minor code: 0 completed

- **Solution:** The CICS Transaction Gateway is unable to contact the target CICS server. If you also found EXCI reason code 203, then the target CICS server is not active or has not opened IRC communication. Consult your CICS or system administrator to verify that the CICS server region is up and running and accepting requests.

- **Problem:** You find the following message:

CTG9631E: Error occurred during interaction with CICS. Error Code=ECI_ERR_SECURITY_ERROR

- **Solution:** This is a security problem that is related to either validating a user ID or accessing RACF authorized resources. Review the JNI trace for the root cause. If you find:

- EXCI reason code 423

This means that RACF surrogate checking has failed. Contact your RACF administrator to analyze the problem and fix it. Verify that you are using the correct surrogates in CICS.

- RACF return code 143

The user ID is unknown or is not defined to RACF or it does not have an OMVS RACF segment.

- **Problem:** You find the following message:

Return code - 22

- **Solution:** The connection to the gateway was successful; however, the program does not exist. Make sure that you used the right program name or that the program exists.

- ▶ **Problem:** There is a hang after a connect to gateway message.
 - **Solution:** The port of the remote gateway cannot be found. Check your port definitions.
- ▶ **Problem:** You find the following message:


```
IC00079E:com.ibm.connector2.ims.ico.IMSTCPIPManagedConnection@3b1bf125.getOutputData (InteractionSpec) error. IMS returned DFS message.
```

 - **Solution:** The first eight characters of the input could not be recognized as a valid transaction, a logical terminal name, or a command. This usually means the transaction name that you specified in the input request is not recognized by the target IMS system. Make sure that you have defined the correct transaction name.
- ▶ **Problem:** You find the following message:


```
IC00001E:com.ibm.connector2.ims.icoIMSTCPIPManagedConnection@d6fd946.processOutputOTMAMsg(byte [], InteractionSpec, Record) error. IMS Connect returned the error: RETCODE=[8], REASONCODE=[SECFNPUI]. Security failure; no password and no user ID.
```

 - **Solution:** The application description specifies the res-auth application but the application did not provide a user ID or a password. Either correct the res-auth configuration or consider changing the authorization method from res-auth to container.
- ▶ **Problem:** You find the following message:


```
IC00003E:com.ibm.connector2.ims.ico.IMSTCPIPManagedConnection@5072da31.connect() error. Failed to connect to host [p390.poughkeepsie.ibm.com], port [3500]. [java.net.ConnectException: Connection refused: connect]
```

 - **Solution:** The IMS Connect task on the specified host is not accepting the connection request. Verify that the IMS Connect task is active in the target host system and is listening on the specified port.
- ▶ **Problem:** You find the following message:


```
IC00064E:com.ibm.connector2.ims.ico.IMSLocalOptionManagedConnection@12d18e56.processSubject(javax.security.auth.subject aSubject) error
```

 - **Solution:** IMS connect was unable to validate the user ID and password that were sent in your request with the external security manager. Either change the user ID and password for the access, provide the correct credentials, or determine if this is correct behavior because only authorized requests should be processed.

Archived

Problem Determination Means and Tools

For the day-to-day tasks that are involved in the process of IBM WebSphere Application Server for z/OS V6 administration, you need powerful and efficient tools for analyzing problems, for finding their root cause, and for solving them.

This part can be used as a reference for identifying the means and tools that are used to determine problems for WebSphere for z/OS.

We introduce and explain:

- ▶ Useful commands for the WebSphere for z/OS environment
- ▶ WebSphere for z/OS logs
- ▶ Traces and dumps
- ▶ Diagnostic tools for problem determination in WebSphere for z/OS
- ▶ Other helpful tools

For each of these means, we provide information about it, about when to use it, and about how to use it. We describe the output, discuss how to interpret it, and provide an example.

Note: Our focus in this redbook is on IBM products. This is no valuation; rather, it represents our experience. If you explore and use other products from various vendors for problem determination in the WebSphere for z/OS environment, the authors would welcome your comments and recommendations.

Archived

Commands

There are various commands that can be useful when you are trying to determine the root cause of problems in WebSphere for z/OS and are preparing to analyze logs, traces, dumps, and diagnostic tool output. This chapter is intended to be a quick reference guide to these commands.

Although they are not specific to problem determination for WebSphere for z/OS, we consider them very useful and powerful for performing day-to-day administrative tasks with WebSphere for z/OS and for identifying problems.

In the sections that follow, we introduce you to:

- ▶ A few command-line tools
- ▶ z/OS commands for WebSphere such as MODIFY, DISPLAY, and TRACE
- ▶ Some TCP/IP commands
- ▶ Related USS for z/OS (OMVS) commands such as **df**, **du**, **ps**, **pid**
- ▶ The WASgrep.sh command for searching string patterns,
- ▶ The Windows FTP command

18.1 Commands for administering WebSphere for z/OS

You can use several command-line tools to start, stop, and monitor WebSphere server processes and nodes. These only work on local servers and nodes and not on a remote server or node. To use them, go to the bin directory, and issue one of these commands:

```
START appserver_proc_name
STOP appserver_proc_name
START dmgr_proc_name
STOP dmgr_proc_name
START nodeagent_proc_name
STOP nodeagent_proc_name
addNode
serverStatus
removeNode
cleanupNode
syncNode
backupConfig
restoreConfig
EARExpander
GenPluginCfg
```

For more information about the command line tools and syntax, search for “Using command line tools” at the *WebSphere for z/OS Information Center*.

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

For other MVS system commands that can assist with the display, modification, and monitoring of z/OS subsystems, such as WLM, RRS, and the WebSphere for z/OS components, refer to *z/OS V1R5 MVS System Commands*, GC28-1781.

18.2 z/OS MODIFY commands

You can use the MODIFY (short command: F) command to send requests to the WebSphere for z/OS controller region. In this section, we discuss how to use MODIFY and show a combination of display commands for requesting information about the server and options for requesting trace and diagnostic data. You must use the server name, not the started task (STC) name, with MODIFY. You can find the correct server name either in the JCL for your WebSphere for z/OS server (Example 18-1), or you can issue the `d a,1` system command.

Example 18-1 WebSphere for z/OS server JCL

```
//PDACR PROC ENV=,PARMS=' ',Z=PDACRZ
// SET ROOT='/waspdconfig/pdcell'
// SET FOUT='properties/service/logs/applyPTF.out'
//APPLY EXEC PGM=BPXBATCH,REGION=OM,
// PARM='SH &ROOT./&ENV..HOME/bin/applyPTF.sh inline'
//STDOUT DD PATH='&ROOT./&ENV..HOME/&FOUT.',
// PATHOPTS=(OWRONLY,OCREAT,OAPPEND),PATHMODE=(SIRWXU,SIRWGX)
//STDERR DD PATH='&ROOT./&ENV..HOME/&FOUT.',
// PATHOPTS=(OWRONLY,OCREAT,OAPPEND),PATHMODE=(SIRWXU,SIRWGX)
//BBOCTL EXEC PGM=BBOCTL,COND=(8,EQ),REGION=OM,TIME=MAXIMUM,
// PARM='TRAP(ON,NOSPIE),ENVAR("_EDC_UMASK_DFLT=007") / &PARMS.'
//BBOENV DD PATH='&ROOT/&ENV/was.env'
// INCLUDE MEMBER=&Z
//
```

The message for your commands is issued to the job log of your WebSphere for z/OS server and to your z/OS system log. As with other z/OS system commands, you can issue them either on a system console or you can use a system command interface, such as SDSF or JES33 Spooler Interface (EJES).

The general syntax for the system commands is:

MODIFY <Server_Name>,DISPLAY,<Options>

The syntax is broken down as follows:

Server NAME	Server name as specified in the JCL
DISPLAY	Fixed keyword
Options	HELP, SERVERS, TRACE, WORK

The MODIFY command can be abbreviated by using the “F” character, for example:

f bboasr2a,display,trace

Note that the system commands are not case-sensitive.

MODIFY <Server_Name>,DISPLAY returns this information:

STC/server name	Started task name and server name
Status	ACTIVE
System name	SYSID of the system where WebSphere for z/OS is active
Level	Build level of your WebSphere for z/OS server

18.2.1 z/OS DISPLAY commands

We found the commands shown in the following examples especially useful.

MODIFY <Server_Name>,DISPLAY

Example 18-2 shows the result of this MODIFY command.

Example 18-2 MODIFY <Server_Name>,DISPLAY

```
F PDSR01A,DISPLAY
BB000173I SERVER PDSR01/PDSR01A ACTIVE ON SC49 AT LEVEL W510004.
BB000188I END OF OUTPUT FOR COMMAND DISPLAY
```

MODIFY <Server_Name>,DISPLAY,HELP

Example 18-3 shows the result of this MODIFY command.

Example 18-3 MODIFY <Server_Name>,DISPLAY,HELP

```
F PDSR01A,DISPLAY,HELP
BB000178I THE COMMAND DISPLAY, MAY BE FOLLOWED BY ONE OF THE FOLLOWING 042 KEYWORDS:
BB000179I SERVERS - DISPLAY ACTIVE CONTROL PROCESSES
BB000179I SERVANTS - DISPLAY SERVANT PROCESSES OWNED BY THIS CONTROL 044 PROCESS
BB000179I SESSIONS - DISPLAY INFORMATION ABOUT COMMUNICATIONS SESSIONS
BB000179I TRACE - DISPLAY INFORMATION ABOUT TRACE SETTINGS
BB000179I JVMHEAP - DISPLAY JVM HEAP STATISTICS
BB000179I WORK - DISPLAY WORK ELEMENTS
BB000179I ERRLOG - DISPLAY THE LAST 10 ENTRIES IN THE ERROR LOG
BB000188I END OF OUTPUT FOR COMMAND DISPLAY,HELP
```

MODIFY <Server_Name>,DISPLAY,SERVERS

Example 18-4 shows the result of this MODIFY command.

Example 18-4 MODIFY <Server_Name>,DISPLAY,SERVERS

```
F PDSR01A,DISPLAY,SERVERS
BB000182I SERVER          ASID  SYSTEM  LEVEL
BB000183I PDCELL   /SC49    3F5x  SC49    W510004
BB000183I PDAGNTB  /PDAGNTB  78x   SC42    W510004
BB000183I PDSR01   /PDSR01A  3F6x  SC49    W510004
BB000183I PDCELL   /SC42     72x   SC42    W510004
BB000183I PDDMGR   /PDDMGR   3EBx  SC49    W510004
BB000183I PDAGNTA  /PDAGNTA  3ECx  SC49    W510004
BB000188I END OF OUTPUT FOR COMMAND DISPLAY,SERVERS
```

MODIFY <Server_Name>,DISPLAY,SERVERS returns the following information:

SERVER	STC and server name for all active servers
ASID	ASID for all active servers
SYSTEM	System ID (SYSID) of the system on which the server is active
LEVEL	Build level of each active server

MODIFY <Server_Name>,DISPLAY,TRACE

Example 18-5 shows the result of this MODIFY command.

Example 18-5 MODIFY <Server_Name>,DISPLAY,TRACE

```
F PDSR01A,DISPLAY,TRACE
BB000224I TRACE INFORMATION FOR SERVER PDSR01/PDSR01A/STC05755
BB000197I LOCATION = SYSPRINT BUFFER
BB000197I AGGREGATE TRACE LEVEL = 1
BB000197I EXCEPTION TRACING = RAS(0), Common Utilities(1), COMM(3), 059
ORB(4), OTS(6), Shasta(7), OS/390 Wrappers(9), Daemon(A), Security(E),
Externalization(F), JRAS(J), J2EE(L), Logging(M)
BB000197I BASIC TRACING =
BB000197I DETAILED TRACING =
BB000197I TRACE SPECIFIC = NONE SPECIFIED
BB000197I TRACE EXCLUDE SPECIFIC = NONE SPECIFIED
BB000225I TRACE INFORMATION FOR SERVER PDSR01/PDSR01A/STC05755 064
COMPLETE
```

The **MODIFY <Server_Name>,DISPLAY,TRACE** command returns the following information:

SERVER/STC	Started task name and server name
LOCATION	Target location for trace data
LEVEL	Trace level
OPTIONS	Trace options

MODIFY <Server_Name>,DISPLAY,ERRLOG

You can use this command to display the last 10 messages in the error log even if you are not routing them to a log stream.

Example 18-6 on page 199 shows only the last three entries in the log.

Example 18-6 WebSphere for z/OS DISPLAY,ERRLOG command

```
F DISPLAY,ERRLOG
BB000188I END OF OUTPUT FOR COMMAND DISPLAY,WORK,ALL,SRS
F PDSR01A,DISPLAY,ERRLOG
BB000266I (STC05755) BossLog: { 0002} 2004/09/21 22:13:44.668 01 138
SYSTEM=SC49 SERVER=PDSR01A PID=OX03080069 TID=OX216AC930 00000000
c=UNK ./bborjtr.cpp+830 ... BB000222I TRAS0017I: The startup trace
state is *=all=disabled.
BB000266I (STC05755) BossLog: { 0003} 2004/09/21 22:13:53.044 01 139
SYSTEM=SC49 SERVER=PDSR01A PID=OX03080069 TID=OX216AC930 00000000
c=UNK ./bborjtr.cpp+830 ... BB000222I SECJ0231I: The Security
component's FFDC Diagnostic Module com.ibm.ws.security.core.SecurityDM
registered successfully: true
BB000266I (STC05755) BossLog: { 0004} 2004/09/21 22:13:53.485 01 140
9} 2004/09/21 22:14:51.908 01 145
SYSTEM=SC49 SERVER=PDSR01A PID=OX03080069 TID=OX2173C430 0X00001A
c=UNK ./bborjtr.cpp+842 ... BBOJ0087W MDB Workload Classification
Support is not enabled
BB000188I END OF OUTPUT FOR COMMAND
DISPLAY,ERRLOG
```

DISPLAY WLM,APPLENV=*

To display the WLM application environment names for WebSphere for z/OS V5, use this command as shown in Example 18-7.

*Example 18-7 DISPLAY WLM,APPLENV=**

```
D WLM,APPLENV=*
IWM029I 08.28.43 WLM DISPLAY 768
APPLICATION ENVIRONMENT NAME    STATE    STATE DATA
BBOASR1                         AVAILABLE
BBOASR2                         AVAILABLE
CBINTFRP                       AVAILABLE
CBNAMING                       AVAILABLE
CBSYSMT                         AVAILABLE
C1INTFRP                       AVAILABLE
C1NAMING                       AVAILABLE
C1OASR1                        AVAILABLE
C1OASR2                        AVAILABLE
C1SYSMT                         AVAILABLE
DBD7MWLM                       AVAILABLE
DBD8MWLM                       AVAILABLE
DB7AODBA                       AVAILABLE
DB7EUTIL                       AVAILABLE
DB7EWLM                        AVAILABLE
DB7LSQL                        AVAILABLE
DB7LUTIL                       AVAILABLE
DB7LWLM                        AVAILABLE
DB7LWLM2                       QUIESCED
DB7PDBUG                       AVAILABLE
DB7PJAVS                       AVAILABLE
DB7PODBA                       AVAILABLE
```

DISPLAY WLM,DYNAPPL=*

To list all the dynamic application environment names (for WebSphere for z/OS V6), use this command as shown in Example 18-8 on page 200.

Example 18-8 DISPLAY WLM,DYNAPPL=*

```
D WLM,DYNAPPL=*
IWM029I 08.39.26 WLM DISPLAY 854
DYNAMIC APPL. ENVIRON. NAME      STATE      STATE DATA
PDSR01                          AVAILABLE
ATTRIBUTES: PROC=PDASR           SUBSYSTEM TYPE: CB
SUBSYSTEM NAME: PDSR01A          NODENAME: PDCELL
PDDMGR                          AVAILABLE
ATTRIBUTES: PROC=PDASR           SUBSYSTEM TYPE: CB
SUBSYSTEM NAME: PDDMGR           NODENAME: PDCELL
CLU491                          AVAILABLE
ATTRIBUTES: PROC=WS5491S         SUBSYSTEM TYPE: CB
SUBSYSTEM NAME: WS491            NODENAME: CL491
```

Other DISPLAY commands

Consider the following DISPLAY, WORK commands to display specific information about active server components:

```
MODIFY <Server_Name>,DISPLAY,WORK,SERVLET
MODIFY <Server_Name>,DISPLAY,WORK,SERVLET,SRS
MODIFY <Server_Name>,DISPLAY,WORK,SUMMARY
MODIFY <Server_Name>,DISPLAY,WORK,SUMMARY,SRS
MODIFY <Server_Name>,DISPLAY,WORK,ALL
MODIFY <Server_Name>,DISPLAY,WORK,ALL,SRS
MODIFY <Server_Name>,DISPLAY,WORK,CLINFO
```

You can also use the commands in Table 18-1.

Table 18-1 Useful z/OS DISPLAY commands

Command	Explanation
d a,all	All jobs running on the system
d asm,page=all	Page data sets and utilization of page space
d grs,c	Global resource serialization - contention
d grs,res=(syszbbo,*)	GRS ENQs by WebSphere
d iplinfo	IPL time and bootstrap parms
d logger,l	Logger logstreams
d parmlib	PARMLIB data sets used for this IPL
d omvs,a=all	UNIX address spaces (processes)
d omvs,f o p	HFS file system in use config. PFS
d opdata	Operator command prefixes
d r,l	Outstanding WTORs (Write To Operator with Replys)
d smf	SMF recording data sets status
d symbols	System symbolics
d tcpip,,n,route home	TCP/IP routes, home
d tcpip,,n,portlist	TCP/IP ports

Command	Explanation
d trace[,comp=cname]	Trace settings
d xcf,cp1 str	XCF parameters and couple data sets structures
\$dsp1	JES2 spool utilization

18.2.2 Basic TRACE commands

Because trace or diagnostic data is not documented and is only useful to the IBM support team, you should only use these traces at the request of IBM. For this reason, we only list a few commands here and do not cover them in detail:

- ▶ To turn tracing to SYSPRINT on and off:
MODIFY <server_name>,TRACETOSYSPRINT=YES | NO
- ▶ To change the overall trace level (F is short for MODIFY):
F <server_name>,TRACEALL=0 | 1 | 2 | 3
- ▶ To turn on basic or detailed tracing for specified components (non-Java):
F <server_name>,TRACEBASIC=(0,1,2...)
F <server_name>,TRACEDetail=(0,1,2..)

Explanations of the codes are in Table 18-2.

Table 18-2 TRACEBASIC/TRACEDetail codes

Code	Explanation	Code	Explanation
0	RAS	A	Daemon
1	Common utilities	E	Security
3	COMM	F	Externalization
4	ORB	J	JRAS
6	OTS	L	J2EE
7	Shastat		
9	z/OS wrappers		

- ▶ To turn off all tracing:
F <server_name>,TRACENONE

Other useful trace commands are:

```
F <control_region_JOBNAME>,TRACESPECIFIC
F <control_region_JOBNAME>,TRACE_EXCLUDE_SPECIFIC
F <control_region_JOBNAME>,TRACETOTRCFILE
F <control_region_JOBNAME>,MDBSTATS
```

18.2.3 Dynamic Java TRACE

One of the most important tools is the MVS Modify command that dynamically turns Java tracing on and off. It is well documented at the *WebSphere for z/OS Information Center*, but can be overlooked among all the other tracing tools.

The z/OS MODIFY command does not require the server to be recycled.

- ▶ To turn on Java tracing for specified components such as com.ibm.ws.security, enter:
F <server_name>,TRACEJAVA='com.ibm.ws.security.yyy.*=all=enabled'
- ▶ To reset to trace settings in your configuration (such as in was.env), enter:
F <server_name>,TRACEINIT
- ▶ To turn off all tracing, enter:
F <server_name>,TRACENONE

For more information, search for “Dynamic Java Trace” at the *WebSphere for z/OS Information Center*.

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

18.3 TCP/IP related commands

We describe several TCP/IP commands that can be very powerful for performing day-to-day administrative and problem determination tasks for WebSphere for z/OS.

Because it is a multi-address-space application, WebSphere for z/OS is very dependent on communication services. Any interruption or wrong configuration prevents communication between the components (for example, between the node agent and deployment manager).

During the development test and deployment process, you also rely on the communication between the client on one platform and the WebSphere for z/OS environment. Tools such as Rational Application Developer must establish a connection between the workstation and z/OS for full functionality.

As with any communication scenario, you must check both sides that want to exchange information for functionality. If both are running on the same platform, you can use the same tool to check the server and client functionality.

Because TCP/IP is a standard protocol and its usage on various platforms is very similar, the commands and their returned information only vary slightly. There are many different commands for analyzing communication problems, but we are presenting only the ones that we consider particularly useful:

- ▶ netstat
- ▶ nslookup
- ▶ ping
- ▶ tracer

We have assumed that you use a TSO Command panel with z/OS and a simple Windows command prompt for the remote system.

Note: There are a number of tools available that issue the same commands under the cover, but provide the information in a more sophisticated way.

For a complete description of all tools, options, and return codes, refer to *TCP/IP V3.2 for MVS: Users Guide*, SC13-7136.

18.3.1 The netstat command

This command provides information about all external connections to your system and about all servers waiting for inbound connections.

To issue the netstat command, you use a TSO Command panel on z/OS and a Windows command prompt on the remote system. When you issue the command in TSO, you receive the following details:

User ID	The user ID of the local and remote task of the active connection.
Conn	The connection ID in TCP/IP.
Local Socket	The IP address and port number in the local system. An IP address of 0.0.0.0 refers to the default stack.
Foreign Socket	The IP address and port number of the remote system. The IP addresses of the local and the remote systems can be the same. For an ESTABLISHED connection in the same system, there are two lines with reversed local and foreign socket information, representing an active connection.
State	The state of the socket. Some important states are LISTEN (server is listening for incoming connections), ESTABLISHED (a connection is established), SYNC_SENT (actively attempting to establish a connection), TIME_WAIT (waiting after close for remote shutdown), FIN_WAIT1(socket is closed and shutting down), and so on.

You should be most interested in the normal session states of LISTEN (a server waiting for work), ESTABLISHED (a communication between client and server is in progress), and SYNC_SENT (usually means an attempt to establish a connection is being blocked by a firewall).

For an explanation of the other states, refer to *TCP/IP V3.2 for MVS: Users Guide*, SC13-7136.

Example 18-9 shows the output after we issued the netstat command in our z/OS system.

Example 18-9 The netstat command and its response.

```
>tso netstat
```

MVS TCP/IP NETSTAT CS V1R6	TCPIP Name: TCPIP	15:29:47
User Id Conn Local Socket Foreign Socket State		
-----	-----	-----
FTPMVS1 0000002A 9.12.4.28..21	0.0.0.0..0	Listen
FTPOE1 0000002B 9.12.4.29..21	0.0.0.0..0	Listen
INETD1 00000033 9.12.4.29..512	0.0.0.0..0	Listen
INETD1 00000030 9.12.4.29..23	0.0.0.0..0	Listen
INETD1 00000032 0.0.0.0..513	0.0.0.0..0	Listen
INETD1 00000031 9.12.4.29..514	0.0.0.0..0	Listen
NFSMVS 00000050 0.0.0.0..10001	0.0.0.0..0	Listen
NFSMVS 0000004F 0.0.0.0..10000	0.0.0.0..0	Listen
NFSMVS 00000055 0.0.0.0..2049	0.0.0.0..0	Listen
NFSMVS 00000052 0.0.0.0..10002	0.0.0.0..0	Listen
PMAP 00000028 0.0.0.0..111	0.0.0.0..0	Listen
REXECD 00000025 9.12.4.28..512	0.0.0.0..0	Listen
REXECD 00000026 9.12.4.28..514	0.0.0.0..0	Listen
TCPIP 00000016 127.0.0.1..1024	127.0.0.1..1025	Establish
TCPIP 000206E2 9.12.4.28..23	9.12.6.132..1438	Establish
TCPIP 0000001B 0.0.0.0..23	0.0.0.0..0	Listen

TCPIP	0000000C	127.0.0.1..1024	0.0.0.0..0	Listen
TCPIP	00000015	127.0.0.1..1025	127.0.0.1..1024	Establish
TCPIP	0001492A	9.12.4.28..23	9.12.6.136..2330	Establish
WASPDCTG	000017E3	0.0.0.0..2006	0.0.0.0..0	Listen
WS551	00000275	0.0.0.0..19080	0.0.0.0..0	Listen
WS551	0000024D	0.0.0.0..10003	0.0.0.0..0	Listen
WS551	00000249	0.0.0.0..18880	0.0.0.0..0	Listen
WS551	00000276	0.0.0.0..19443	0.0.0.0..0	Listen
WS551	0000024B	0.0.0.0..12809	0.0.0.0..0	Listen
WS551D	0000023B	0.0.0.0..15655	0.0.0.0..0	Listen
WS6552	0001385F	0.0.0.0..29080	0.0.0.0..0	Listen
WS6552	00013839	0.0.0.0..10044	0.0.0.0..0	Listen
WS6552	00013837	0.0.0.0..22809	0.0.0.0..0	Listen
WS6552	00013835	0.0.0.0..28880	0.0.0.0..0	Listen
WS6552	00013860	0.0.0.0..29443	0.0.0.0..0	Listen
WS6552D	0001381F	0.0.0.0..25655	0.0.0.0..0	Listen
WS6552S	00014871	9.12.4.28..10054	9.12.4.30..38050	ClosWait
NFSCNT	00000022	0.0.0.0..1017	*.*.*	UDP
NFSCNT	00000021	0.0.0.0..1018	*.*.*	UDP
NFSCNT	00000020	0.0.0.0..1019	*.*.*	UDP
NFSCNT	0000001F	0.0.0.0..1020	*.*.*	UDP
NFSCNT	0000001E	0.0.0.0..1021	*.*.*	UDP
NFSCNT	0000001D	0.0.0.0..1022	*.*.*	UDP
NFSCNT	0000001C	0.0.0.0..1023	*.*.*	UDP
NFSMVS	00000054	0.0.0.0..2049	*.*.*	UDP
NFSMVS	0000004E	0.0.0.0..10001	*.*.*	UDP
NFSMVS	00000051	0.0.0.0..10002	*.*.*	UDP
NFSMVS	0000004D	0.0.0.0..10000	*.*.*	UDP
NFSMVS	00000053	0.0.0.0..10003	*.*.*	UDP
PMAP	00000027	0.0.0.0..111	*.*.*	UDP
SYSLOGD4	00000024	0.0.0.0..514	*.*.*	UDP

18.3.2 The nslookup command

The name server lookup command tells you the name and the IP address of the registered name server and uses this connection to give you the IP address of a given server.

To issue the nslookup command, you use a TSO Command panel for z/OS and a Windows command prompt for the remote system. The command provides the following information:

Server	Host name of the DNS server
Address	IP address of the DNS server
Name	Host name of the requested server
Address	IP address of the requested server

Example 18-10 shows output from the nslookup command on a workstation.

Example 18-10 The nslookup command and its response

```

WASPD2 @ SC55:/u/waspd2>nslookup wtsc55.itso.ibm.com
Defaulting to nslookup version 4
Starting nslookup version 4
Server: itsodns.itso.ibm.com
Address: 9.12.6.7

Name: wtsc55.itso.ibm.com
Address: 9.12.4.28

```

18.3.3 The ping command

The ping command sends a request to a remote system and monitors the response time and the number of lost replies.

The ping command is mainly used on the client to check communication and the proper name and server configuration. If you can ping your WebSphere for z/OS system by using the name of the system, you can be almost certain that your client network configuration is fine. If you cannot ping your system, you should ping the IP address and use the nslookup command to check the name and server configuration.

To issue the ping command, you use a TSO Command panel for z/OS and a Windows command prompt for the remote system. The ping command provides the following information:

No. of responses	The ping command counts the number of responses. On z/OS, the ping command only sends one request. On other client systems, you should specify a count to limit the number of requests; otherwise, you must interrupt the process by pressing Ctrl+C.
Response time	How many seconds elapsed until the reply came back.
Successes	How many tries were successful and many requests were lost.

Example 18-11 shows sample output from the ping command on a workstation.

Example 18-11 The ping command and its response

```
C:\Documents and Settings\T0T188>ping wtsc55.itso.ibm.com

Pinging wtsc55.itso.ibm.com [9.12.4.28] with 32 bytes of data:

Reply from 9.12.4.28: bytes=32 time=4ms TTL=63
Reply from 9.12.4.28: bytes=32 time=4ms TTL=63
Reply from 9.12.4.28: bytes=32 time=4ms TTL=63
Reply from 9.12.4.28: bytes=32 time=7ms TTL=63

Ping statistics for 9.12.4.28:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 4ms, Maximum = 7ms, Average = 4ms
```

18.3.4 The tracert command

The trace route (**tracert**) command follows all intermediate network hops until it reaches a given IP address and informs you about any delays in this process.

The trace route command can be used from both ends of the communication to check for routers or bridges that are preventing communication or causing unnecessary delays.

Note: The trace route command in z/OS TSO/E is called TRACERTE and not tracert.

The trace route command provides the following information:

Address	Address of the target server
Delay	One line per hop, with IP address and delay in ms

Example 18-12 shows sample output from the `tracert` command on a workstation.

Example 18-12 The `tracert` command and its response

```
C:\Documents and Settings\TOT188>tracert plpsc.pok.ibm.com
```

```
Tracing route to plpsc.pok.ibm.com [9.56.214.1]
over a maximum of 30 hops:
```

```
  1      3 ms      3 ms      3 ms  pok6509r.itso.ibm.com [9.12.6.92]
  2      4 ms      4 ms      4 ms  9.56.1.189
  3      4 ms      4 ms      4 ms  pok-ud-2a-v993.pok.ibm.com [9.56.126.3]
  4      4 ms      4 ms      4 ms  pok-co-a-v808.pok.ibm.com [9.56.2.33]
  5      4 ms      4 ms      4 ms  pok-bd-b-ge0-4.pok.ibm.com [9.56.2.6]
  6      4 ms      4 ms      4 ms  pok-sc-a-v256.pok.ibm.com [9.56.1.6]
  7      7 ms      4 ms      5 ms  pok-sd-5b-ge2-1.pok.ibm.com [9.56.208.4]
  8      8 ms      5 ms      4 ms  plpsc.pok.ibm.com [9.56.214.1]
```

```
Trace complete.
```

18.4 USS and OMVS commands

Some USS and OMVS commands can be very powerful tools for performing day-to-day WebSphere for z/OS administrative tasks. We introduce five of them here:

- ▶ The `df` command for displaying a file system
- ▶ The `du` command for displaying usage of disk space
- ▶ The `ps` command for displaying process and thread information
- ▶ The `DISPLAY` command for thread details in a Servant Region
- ▶ The `WASgrep.sh` command for searching string patterns in XML files

18.4.1 Display file system with `df`

In a z/OS USS (OMVS) environment, you allocate a hierarchical file system (HFS) data set with a disk space allocation of a specific fixed size. You can mount the HFS data set to a certain mount point and make it available to the application.

Using `df` (display file) as a common UNIX command is particularly useful in a situation where you want to determine whether any HFS is full.

To use the command, first you go to a subdirectory that you would like to check. Then, enter:

```
df -k .
```

In the command, `-k` is the option flag request for the output to be in kilobytes and the “.” means “this directory.”

The `df` command displays:

- ▶ Mount-point name
- ▶ HFS data set name
- ▶ Total size of the file system and the remaining disk space that is still available

Figure 18-1 on page 207 shows the `df` command output when executed from the OMVS command line.

```

WASPD2 @ SC55:/waspdconfig/pdcell>df -k ./
Mounted on      Filesystem              Avail/Total    Files Status
/waspdconfig/pdcell (OMVS.WAS.PDCCELL.CONFIG.HFS) 139200/288000  4294949456
Available
/waspdconfig/rescell (OMVS.WAS.RESCCELL.CONFIG.HFS) 222224/288000  4294960903
Available

```

Figure 18-1 USS command *df* display

This display shows that there are two file systems mounted under the /waspdconfig subdirectory:

- ▶ ./pdcell has 143560 K available out of a total of 288000 K (50% full).
- ▶ ./rescell has 222224 K available out of a total of 288000 K (23% full).

18.4.2 Display disk space usage with *du*

In a USS environment, you often encounter disk full or file system full problems. In such situations, it is useful to know which file or directory has consumed most of the disk space. The disk usage (*du*) command is used to show the amount of disk space that is consumed by one or more directories (or directory trees).

In this section, we look at a simple tool that can display the files or subdirectories that use most of the disk space. To achieve this, you “pipe” the output from the *du* command into a sort command and then “pipe” the sorted results into a head command.

First, you go to the file system that encountered the file system full problem. Then, you issue:

```
du -k . | sort -r | head -n 20
```

In this command:

- ▶ *du -k .* displays list that consists of all the files and subdirectories.
- ▶ *sort -r* sorts the list in reverse or descending order.
- ▶ *head -n 20* displays only the top 20 rows of the sorted list.

These three commands are connected with the UNIX “|” piping function.

This tool displays a list of rows sorted in descending order. Each row consists of two columns: the size in kilobyte block and the name of the subdirectories as shown in Example 18-13.

Example 18-13 The *du* command output

```

WASPD2 @ SC55:/waspdconfig/pdcell>du -k | sort -r | head -n 20
119580 .
 84644 ./DeploymentManager
 43436 ./DeploymentManager/installedApps
 43428 ./DeploymentManager/installedApps/pdcell
 43276 ./DeploymentManager/installedApps/pdcell/adminconsole.ear
 42596 ./DeploymentManager/installedApps/pdcell/adminconsole.ear/adminconsole.war
 29576 ./DeploymentManager/installedApps/pdcell/adminconsole.ear/adminconsole.war/WEB-INF
 27000 ./AppServerNodeA
 16144 ./DeploymentManager/wstemp
 14316 ./DeploymentManager/config
 13900 ./AppServerNodeA/config
 13724 ./DeploymentManager/config/cells
 13708 ./DeploymentManager/config/cells/pdcell
 13128 ./DeploymentManager/config/cells/pdcell/applications
 12392 ./AppServerNodeA/config/backup

```

```

12384 ./AppServerNodeA/config/backup/base
12188 ./DeploymentManager/config/cells/pdcell1/applications/adminconsole.ear
11920 ./AppServerNodeA/config/backup/base/cells
11912 ./AppServerNodeA/config/backup/base/cells/pdcella
11660 ./AppServerNodeA/config/backup/base/cells/pdcella/applications

```

18.4.3 Display thread information with ps

You can use the UNIX `ps` command to learn how many threads there are in an application servant region. To do this, you have to learn the process ID (PID) of the servant region as follows:

1. Using the z/OS SDSF display, obtain the ASIDX (address space ID in hexadecimal).
2. With ASIDX, use the z/OS command from the console command to get the PID:

```
/d omvs,asid=xxx
```

3. From the OMVS command line, issue:

```
ps -p <pid> -m | wc -l
```

Example 18-14 shows the result.

Example 18-14 The ps command to find the number of threads

```

SDSF DA SC49 SC49 PAG 0 SIO 53 CPU 4/ 4 LINE 1-7 (7)
COMMAND INPUT ==> SCROLL ==> CSR
NP JOBNAME StepName ProcStep JobID Owner C Pos DP Real Paging SIO CPU% ASID ASIDX
PDDEMN PDDEMN BBODAEMN STC12605 WSDMNCR1 NS FE 7680 0.00 0.00 0.00 89 0059
PDAGNTA PDAGNTA BBOCTL STC12612 ASCR1 NS FE 50T 0.00 1.18 0.02 92 005C
PDSR01AS PDSR01AS BBOSR STC18036 ASSR1 IN FB 55T 0.00 0.00 0.00 94 005E
PDDMGRS PDDMGRS BBOSR STC15651 DMSR1 IN FE 125T 0.00 30.84 0.02 97 0061
PDSR01AS PDSR01AS BBOSR STC18038 ASSR1 IN FE 55T 0.00 0.01 0.02 101 0065
PDDMGR PDDMGR BBOCTL STC13312 ASCR1 NS FE 54T 0.00 1.22 0.02 1005 03ED
PDSR01A PDSR01A BBOCTL STC18031 ASCR1 NS FE 40T 0.00 0.01 0.04 1011 03F3

COMMAND INPUT ==> /D OMVS,ASID=65 SCROLL ==
RESPONSE=SC49
BPX0040I 18.21.20 DISPLAY OMVS 833
OMVS 000E ACTIVE OMVS=(5A)
USER JOBNAME ASID PID PPID STATE START CT_SECS
ASSR1 PDSR01AS 0065 84410478 1 HR---- 17.53.49 28.11
LATCHWAITPID= 0 CMD=BBOSR

From OMVS command line:
>ps -p 84410478 -m | wc -l
25

```

In this sample, you can see how we issued commands based on the result of the previous command:

1. **SDSF** shows the ASIDX for PDSR01AS as 0065.
2. **/D OMVS,ASID=65** shows the PID as 84410478.
3. **ps -p 84410478 -m | wc -l** gives the number of threads as 25.

18.4.4 Display thread details with DISPLAY

You can use the DISPLAY OMVS z/OS command to obtain detailed information about threads in a particular application servant region. This information includes the TCB address, whether it is a WLM or a non-WLM thread, and CPU time usage. To use the command:

1. Obtain the PID of the servant region. From the z/OS SDSF display, you can obtain the ASIDX (address space ID in hexadecimal). With ASIDX, use the z/OS command from the console command to get the PID: `/d omvs,asid=xxx`
2. From the OMVS command line, use `/d omvs,pid=yyy`

Figure 18-2 shows the result of these commands.

SDSF DA SC49 SC49 PAG 0 SIO 53 CPU 4/ 4 LINE 1-7 (7)														
COMMAND INPUT ==> DA										SCROLL ==> CSR				
NP	JOBNAME	StepName	ProcStep	JobID	Owner	C	Pos	DP	Real	Paging	SIO	CPU%	ASID	ASIDX
	PDDEMN	PDDEMN	BBODAEMN	STC12605	WSDMNCR1	NS	FE	7680	0.00	0.00	0.00	89	0059	
	PDAGNTA	PDAGNTA	BBOCTL	STC12612	ASCR1	NS	FE	50T	0.00	1.18	0.02	92	005C	
	PDSR01AS	PDSR01AS	BBOSR	STC18036	ASSR1	IN	FB	55T	0.00	0.00	0.00	94	005E	
	PDDMGRS	PDDMGRS	BBOSR	STC15651	DMSR1	IN	FE	125T	0.00	30.84	0.02	97	0061	
	PDSR01AS	PDSR01AS	BBOSR	STC18038	ASSR1	IN	FE	55T	0.00	0.01	0.02	101	0065	
	PDDMGR	PDDMGR	BBOCTL	STC13312	ASCR1	NS	FE	54T	0.00	1.22	0.02	1005	03ED	
	PDSR01A	PDSR01A	BBOCTL	STC18031	ASCR1	NS	FE	49T	0.00	0.01	0.04	1011	03F3	
COMMAND INPUT ==> /D OMVS,ASID=65										SCROLL ==				
RESPONSE=SC49														
BPXO040I 18.21.20 DISPLAY OMVS 833														
OMVS 000E ACTIVE OMVS=(5A)														
USER	JOBNAME	ASID	PID	PPID	STATE	START	CT_SECS							
ASSR1	PDSR01AS	0065	84410478	1	HR----	17.53.49	28.11							
LATCHWAITPID= 0 CMD=BBOSR														
COMMAND INPUT ==> /D OMVS,PID=84410478										SCROLL ==				
RESPONSE=SC49														
BPXO040I 18.22.14 DISPLAY OMVS 835														
OMVS 000E ACTIVE OMVS=(5A)														
USER	JOBNAME	ASID	PID	PPID	STATE	START	CT_SECS							
ASSR1	PDSR01AS	0065	84410478	1	HR----	17.53.49	28.11							
LATCHWAITPID= 0 CMD=BBOSR														
THREAD_ID	TCB@	PRI	JOB	USERNAME	ACC	TIME	SC	STATE						
2172D91000000000	006F4118				26.611		PTC	YU						
2172E62000000001	006E0170				.001		PTX	JY V	TCB address					
2173004000000002	006D04F0				.001		PTX	JY V						
21730D5000000003	006D0260				.003		RED	JY V						
2173277000000004	006DCE88				.005		CLO	JY V						
2173348000000005	006DCCF0				.085		STE	JY V						
21734EA000000006	006D58				.001		PTX	JY	Thread ID					
21735BB000000007	006DC9C0				.001		PTX	JY V						
217368C000000008	006DC690				.001		PTX	JY V						
217375D000000009	006DC360				.094		STE	JY V						
217382E00000000A	006DC1C8				.001		PTX	JY V						
21738FF00000000B	006CFE88				.443		STE	JY V						
21739D000000000C	006CFCF0				.004		PTX	JY V	CPU time					
2173AA100000000D	006CFA60				.003		RED	JY V						
21739D000000000C	006CFCF0				.004		RED	JY V						
2173AA100000000D	006CFA60				.003		RED	JY V						
2173B72000000011	006CF7D0				.001		PTX	JY V						
2173C43000000014	006CCE88				.001		PTX	JR V						
2173D14000000015	006CF180				.001		PTX	JR V						
2173DE5000000016	006C71F0	WLM			.001		PTX	JR V	WLM threads					
2177CC2000000017	006C7388	WLM			.001		PTX	JR V						
2178830000000018	006C7520	WLM			.001		PTX	JR V						
2178B74000000019	006C76B8	WLM			.001		PTX	JR V						
2178C4500000001A	006CC0B0	WLM			.001		PTX	JR V						
2178D1600000001B	006CC248	WLM			.001		PTX	JR V						

Figure 18-2 Using DISPLAY OMVS to show thread information

18.4.5 Search string patterns with WASgrep.sh

The WASgrep.sh command searches for a particular string pattern in all of the ASCII-based XML files in the current directory and all subdirectories as follows:

1. The script does a **find** at the current directory to obtain a list of all the XML files in the current directory and subsequent subdirectories.
2. It converts each ASCII-based XML file into an EBCDIC-based temporary file.
3. It searches for the string pattern in the temporary file.

Example 18-15 shows the contents of the shell script.

Example 18-15 WASgrep shell script

```
#!/bin/sh

ascii=${A2E_ASCII_CODEPAGE:-ISO8859-1}
ebcdic=${A2E_EBCDIC_CODEPAGE:-IBM-1047}

cmd="iconv -f $ascii -t $ebcdic"

if [ "$1" = "?" ]
then
    echo "This script searches for *.xml files in curent dir and subdirs"
    echo "This script will convert a file (ascii) to ebcdic and then"
    echo "search it for the specified string"
    echo "the original file(ascii) is untouched"
    exit 0
fi

tempfile=/tmp/checkfile
for file in `find . -name "*.xml"`
do
    echo '-----'
    echo $file
    $cmd $file > $tempfile    # convert the file
    grep $1 $tempfile
    rm $tempfile
done

exit
```

To use the tool, you **cd** to an appropriate directory. Then, you run the shell script with a string pattern that you want to search:

```
<script-directory>/WASgrep.sh jmsQCF2
```

This tool displays the XML file name and all the text lines that have the string pattern. Example 18-16 shows that only the server.xml file contains the was.wlmTimeout search string.

Example 18-16 Search string in server.xml file

```
cd /waspdconfig/pdcell/AppServerNodeB/config/cells/pdcell/nodes/pdnodeb/servers/pdsr02b/

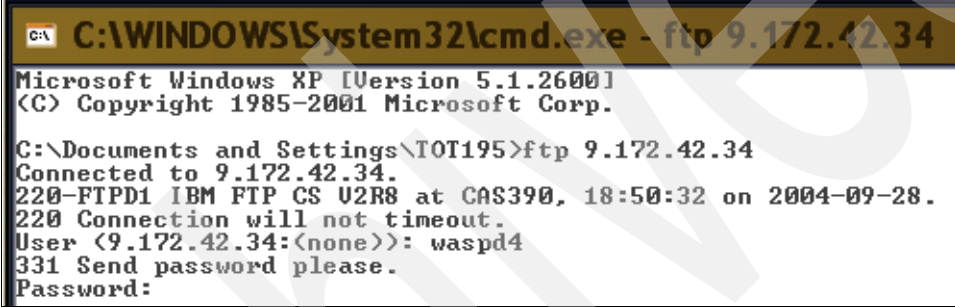
/u/waspd2/WASgrep.sh was.wlmTimeout
-----
./namestore-cell.xml
-----
./namestore-node.xml
```

```
-----  
./resources.xml  
-----  
./server.xml  
  <properties xmi:id="Property_24" name="was.wlmTimeout" value="300"/>  
-----  
./variables.xml  
-----  
./ws-security.xml  
-----
```

18.5 Windows FTP command

The FTP client that is delivered with Windows offers the functionality of a standard FTP client with a character-based user interface. You can use it to transfer files to and from z/OS systems. Figure 18-3 shows a DOS window with this command:

ftp 9.172.42.34



```
C:\WINDOWS\System32\cmd.exe - ftp 9.172.42.34  
Microsoft Windows XP [Version 5.1.2600]  
(C) Copyright 1985-2001 Microsoft Corp.  
  
C:\Documents and Settings\T0T195>ftp 9.172.42.34  
Connected to 9.172.42.34.  
220-FTPD1 IBM FTP CS U2R8 at CAS390, 18:50:32 on 2004-09-28.  
220 Connection will not timeout.  
User (9.172.42.34:(none)): waspd4  
331 Send password please.  
Password:
```

Figure 18-3 FTP client delivered with Windows

After supplying the user ID and password, you can issue PUT and GET commands to transfer files between the two systems.

Archived

Logs for problem determination in WebSphere for z/OS

In this chapter, we explain the following logs:

- ▶ Job logs and system log (syslog)
- ▶ WebSphere error log (BBORBLOG)
- ▶ First Failure Data Capture
- ▶ Java Logging API
- ▶ IBM HTTP Server logs and trace

For each log, we provide information about the nature of the log, discuss when to use it, and demonstrate how to use it. We describe the output, show how to interpret it, and provide an example.

19.1 Job logs and system log

WebSphere is a collection of server instances working together. Each server instance is made up of a controller region and some number of servant regions. Each of these regions is an address space that, as with any other address space, has a job log that you can view through SDSF. If you run JES3, then you might be using EJES, the JES2 spool.

The system log (SYSLOG) is a record of the console messages. WebSphere messages issued to SYSLOG also show up in the job log. In most problem cases, these logs show information about exceptions, information about abnormal situations, or simply warning messages in the system. These are normally the first logs that you should examine when software problems occur.

19.1.1 When to use system log and job logs

The JES2 spool writes the information to the job logs of the address spaces for the different regions. There are several types of regions that are of interest:

- ▶ For each server instance, there is one controller region and 0 or more servant regions (except for the daemon, for which there is only a controller region).
 - Controller regions, to put it simply, handle communication, receiving requests from clients and sending back responses.
 - Servant regions are given the requests to process, so they do the actual work.
- ▶ You might also have address spaces from local clients and your HTTP server.

Now, given all the job logs available, how do you know which ones have useful information? The answer depends on what is taking place at the time of the error.

In a typical situation, a client sends requests to the servant to process an action against an object. This client might be a local batch program, a remote Java application, a browser-based application, or the administrative console.

The requests are sent to the controller region of the appropriate application server. Then, the requests are processed in a servant. Therefore, the best place to look for messages that are related to the error is in the application servant region. This information might be supported by more detailed messages in the error log.

The application controller region is of most interest in communication failures. For naming registration failures, the application servant regions have messages regarding the failure. Again, the error log has more detailed messages that accompany those in the address space. Some failures result in a CEEDUMP being generated in the job log. Any regions with a CEEDUMP are always of interest. You should determine what is processing the work when the failure occurs.

In the SYSLOG, in addition to the system messages, there are WebSphere for z/OS messages, but there could be messages from other products that have been invoked by or that are related to WebSphere. Indications of dumps are also found in the SYSLOG.

19.1.2 How to set up system log and job logs

When the job output of an address space is viewed with a tool such as SDSF or EJES, it can be viewed either as one piece of output or as different sections.

Use the JOB statement MSGLEVEL parameter to request that the job control statements be printed in the job log output listing. Use **MSGLEVEL=(1,1)** to receive the maximum amount of information in the following order:

1. JES messages and job statistics
2. All job control statements in the input stream and procedures
3. Messages about job control statements
4. JES and operator messages about the processing of the job: allocation of devices; volumes, execution, and termination of job steps and the job; and disposition of data sets

19.1.3 System log and job log output and their interpretation

Each job log output section contains certain types of information:

- ▶ **JESMSGLG:** This section contains start-up messages, including a list of environment variable values and server settings, and the service level of WebSphere, for example:


```
BBOM0007I CURRENT CB SERVICE LEVEL IS build level cf20523.06 release 829  
WAS601.ZNATV date 06/07/05 10:24:12.
```


It also lists the Java service level in a J2EE servant region, for example:


```
BBOJ0011I JVM Build is J2RE 1.4.2 IBM z/OS Persistent Reusable VM 070  
build cm142sr1a-20050209 (JIT enabled: jitc).
```
- ▶ **JESJCL:** This section lists the JCL of the procedure that is running the address space. This is a useful place to look for incorrect STEPLIBs and other JCL related issues.
- ▶ **JESYSMSG:** This section might list more messages, dump information, and provide a list of environment variables and server settings.
- ▶ **CEEDUMP:** An exception in the address space might cause this section to be generated. It lists failure information including trace backs (a trace back shows which functions were last called prior to the program failure).
- ▶ **System output (SYSOUT):** During normal processing, the SYSOUT should be empty, but there are situations that cause output to be written to this section. If the error log stream cannot connect, the messages set to be written to the error log are written to CERR, which goes to SYSOUT. Trace from the JVM occurs when you set the *control_region_jvm_logfile*, *server_region_jvm_logfile*, and *server_region_use_java_g* environment variables to *SYSPRINT*.
- ▶ **SYSPRINT:** The WebSphere for z/OS trace output can be written to SYSPRINT if the environment variable is *ras_trace_outputLocation=SYSPRINT*.

Important: When BM support asks for a trace, always send the *entire* job output, because each section might have useful information that can help debug the problem.

The job log can have a variety of information based on environment variable settings. As a default, it obtains information about the job itself, such as life cycle messages (when it started, when it finished initiating, and so on), the JCL that was used to run the job, data set utilization, and other typical JES messages.

There are also WebSphere messages, which start with the BBO prefix. The messages in the console (what we refer to as SYSLOG messages) are typically related to configuration failures of other products, unrecoverable WebSphere configuration errors, and WebSphere life cycle messages. Messages written explicitly to the job log are more general failure and warning messages. Messages with more details that support these general failure and warnings can be found in the error log.

Other important pieces of useful information that always come out in the job log are the configuration messages. These list the values of the environment variables and the server properties.

You also have the option of using various traces and managing their output. For more information about traces, refer to Chapter 4 in *WebSphere Application Server for z/OS V6, Troubleshooting and support*, GA22-7964-03.

Example 19-1 shows a part of SYSLOG, where information generated at start-up for WebSphere for z/OS is captured. In this example, the release, build level, cell name, node name, and procedure name are displayed.

Example 19-1 SYSLOG sample at start-up

```
+BB000239I WEBSHERE FOR Z/OS SERVANT PROCESS c16422/nd6422/ws6422 IS
  STARTING.
+BBOM0007I CURRENT CB SERVICE LEVEL IS build level cf20523.06 release
  WAS601.ZNATV date 06/07/05 10:24:12.
```

For more information about the messages you can see in the SYSLOG and job log, refer to Appendix A, “Messages and codes” on page 311, and the WebSphere for z/OS Information Center.

19.2 WebSphere error log (BBORBLOG)

This section describes the WebSphere for z/OS error log. This error log is where WebSphere for z/OS saves the detailed error messages from its runtime servers. The WebSphere Application Server for z/OS error log stream is a system logger application. The system logger is a z/OS component that allows applications to log data in a sysplex.

The system logger creates and manages log streams that are written first to a coupling facility or local in-memory buffer and then transferred to log data sets on a DASD for longer term access. Log streams that are written to local buffers rather than to a coupling facility are called DASD-only log streams. For more information about the system logger and sysplex refer to *z/OS V1R6.0 MVS Setting Up a Sysplex*, SA22-7625-10.

Note: There is a significant performance penalty when DASD-only error logging is used.

19.2.1 When to use BBORBLOG

The WebSphere for z/OS error log is a log stream data set that uses the system logger to record messages. These messages are typically more detailed than those from the system console or from the job output of the server address space.

The advantage of the BBORBLOG log is that the error logs from the various WebSphere Application Server address spaces are consolidated into one log. The consolidation includes controller region, servant region, and daemon error logs. This is very useful when you are trying to compare time stamps and errors from the different servers.

If the error log is a DASD-only log stream, this provides single system-only error logging. If you are using the coupling facility for the system logger, then action is sysplex-wide and all the systems in the sysplex can log data in one place.

19.2.2 How to set up BBORBLOG

The BBORBLOG is set up during the server installation using the customization dialogs and all the jobs required for the setup are created for you. If the BBORBLOG was not set up during initial installation, then it can be enabled later. The detailed information can be found in *WebSphere Application Server for z/OS, Version 6, Installing your application serving environment*, GA22-7957. Here is a brief summary of the required steps:

1. Create a new Coupling Facility Resource Management (CFRM) policy with the new structure definition. Refer to sample job BBOWCFRM.
2. Define a WebSphere error log stream by updating the LOGR policy. Refer to sample job BBOERRLG.
3. Check for security authorization. Refer to sample jobs BBODBRAC and BBOWBRAC.

Update the `ras_log_logstreamName` environment variable using the Administrative Console by selecting **Environment** → **WebSphere Variables** → **Scope node**. Locate the variable `ras_log_logstreamName` with the logstream name. Once it is updated, click **OK**. On the tool bar, select **Save** → **Save**. The change takes effect when the server is recycled.

Note: Time stamps will appear in GMT unless they are changed by setting the WebSphere Application Server variable `ras_time_local` to 1.

During server initialization, there is an attempt to connect to the appropriate log stream data set. If this connection is successful, you see messages such as those in Example 19-2, which indicates the name of the data set that is being used.

Example 19-2 Successful connection messages

```
+BBOM0001I  ras_log_logstreamName: WAS.SC42.ERROR.LOG
+BB000024I  ERRORS WILL BE WRITTEN TO WAS.SC42.ERROR.LOG LOG STREAM FOR
JOB WS6422S.
+BB000153I  THE FOLLOWING NUMBER OF MESSAGES WERE WRITTEN TO CERR PRIOR
TO CONNECTING TO LOGSTREAM: 1.
```

If, however, the server cannot connect to the log stream, the message is instead written to CERR, which puts it in the SYSOUT of the job output. This is indicated by this message:

```
BB000024I  ERRORS WILL BE WRITTEN TO CERR FOR JOB <server name>
```

Important: Even if the server successfully connects to the log stream, there will still be a message saying that errors will be written to CERR. This is because during initialization, before the connection to the log stream is made, errors are written to CERR.

When they are written to CERR, or SYSOUT, messages have a header that looks like that shown in Example 19-3. Notice that it is prefaced with BossLog.

Example 19-3 BBORBLOG sample header

```
BossLog: { 0001} 2005/08/08 13:06:36.265 01 SYSTEM=SC42 SERVER=<none>
```

You can view the error log stream output using the BBORBLOG browser. To invoke the browser, go to ISPF option 6 and enter:

```
ex 'BBO.SBBOEXEC(BBORBLOG)' 'WAS.SC42.ERROR.LOG'
```

In this example, BBORBLOG resides in BBO.SBBOEXEC and WAS.SC42.ERROR.LOG is the LOG_STREAM_NAME that was configured in the administrative console. Other examples include:

```
ex 'BBO.SBBOEXEC(BBORBLOG)' 'WAS.SC42.ERROR.LOG 80'
ex 'BBO.SBBOEXEC(BBORBLOG)' 'WAS.SC42.ERROR.LOG noformat'
```

The browser creates a data set named USERID.LOG_STREAM_NAME (for example, WASPD3.WAS.SC42.ERROR.LOG), which contains the formatted contents of the log stream. When the browser is started, it:

1. Allocates the USERID.LOG_STREAM_NAME data set, overwriting any duplicates
2. Populates the data set with the contents of the log stream
3. Puts the user in browse mode on the data set

Important: Each time BBORBLOG is invoked, a static file is created that overwrites the existing file. To refresh the file, it is necessary to re-issue BBORBLOG. If you want to keep the last log, you must rename it before running the tool again.

You can also invoke the BBORBLOG utility from the OMVS shell or Telnet using the shell script shown in Example 19-4.

Example 19-4 Shell script to format error log stream from MVS shell or telnet

```
/* REXX */
/* trace r */
parse arg logstrm format .
if logstrm = '' then logstrm = "WAS.ERROR.LOG"
if format = '' then format = "80"
qual =userid()
file_name = "/tmp/" || qual || ".errorlog"
"rm " || file_name
"touch " || file_name
call syscalls 'SIGOFF'
call bpxwdyn "alloc fi(bbolog) path(' ' || file_name || ' ')"
address LINKMVS "BBORBLOG logstrm format"
call bpxwdyn "free fi(bbolog)"
"vi " || file_name
exit(0)
```

19.2.3 BBORBLOG output and its interpretation

The output of BBORBLOG looks similar to that shown in Example 19-5.

Example 19-5 Error logstream output with line numbers

```
1| 2005/08/08 20:11:04.658 01 SYSTEM=SC42 SERVER=WS6422 JobName=WS6422
2| ASID=0X0403 PID=0X0301014D TID=0X22172FA0 0X000019 c=UNK
3| ./bbooboot.cpp+3152 ... BB000011W The function
4| ACR_ExecutionThread::ProcessInboundRequest(acrwObj *, ThreadCleanUp
5| *, BOSS_Object_Key &, Internal_CORBA_Request &)+3152 received CORBA
6| system exception CORBA::INTERNAL. Error code is C9C2102F.
```

The line numbers together with the output listed in Table 19-1 can help you analyze and understand the output of this log.

Table 19-1 Parts of server log stream record output

Line number	Component	Description
1	2005/08/08 20:11:04.658 01	Date, time stamp, 2-digit record version number
1	SYSTEM=SC42	System name
1	SERVER=WS6422	Server name
2	ASID=0X0403	ASID
2	PID=0X0301014D	PID
2	TID=0X22172FA0 0X000019	Thread identifier (TID)
2	c=UNK	Request correlation information
Lines 1 and 2 help identify when and where the error occurred.		
3	./bbooboot.cpp+3152	File name and line
3	BB000011W	Log message number
The message number can provide detailed information about the error.		
3, 4, 5, 6	The function	Log message
It shows you which function was active in the moment of error; useful information for describing the problem to the IBM Support Center.		
6	Error code is C9C2102F	Error code
Sometimes, the error code is more meaningful than the message number.		

For further details about BBORBLOG see *WebSphere Application Server for z/OS V6, Troubleshooting and support*, GA22-7964-03, and the *WebSphere for z/OS Information Center* at:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

For assistance with error messages, see Appendix A, “Messages and codes” on page 311.

19.3 First Failure Data Capture

This section describes the First Failure Data Capture ((FFDC) facility that is included as part of WebSphere Application Server V5 and later for all platforms. With the FFDC tool, system administrators can identify runtime issues with their WebSphere Application Server in any environment. If an exception occurs, the FFDC tool traps and logs the exception, and then returns control to the thread that threw the exception.

In WebSphere for z/OS V6.0.2, FFDC is enabled by default; in previous versions, it was disabled.

Important: Because each WebSphere address space writes to its own FFDC file, the FFDC files can accumulate over time and take up a significant amount of space. We recommend that, if FFDC is enabled, you prune the FFDC directory periodically to remove old FFDC files that are no longer required.

19.3.1 When to use FFDC

Because the I/O of the information that is captured by the FFDC tool is infrequent, FFDC has virtually no impact on runtime performance. The artifacts produced by the FFDC tool are incremental rather than repeated, which results in very little use of the file system resources. The FFDC tool is delivered as part of any WebSphere Application Server Base or Network Deployment installation and consists of a set of components. These components are the error filter, the analysis engine, the knowledge base that feeds the analysis engine, the diagnostic engine, and a set of diagnostic modules (Figure 19-1).

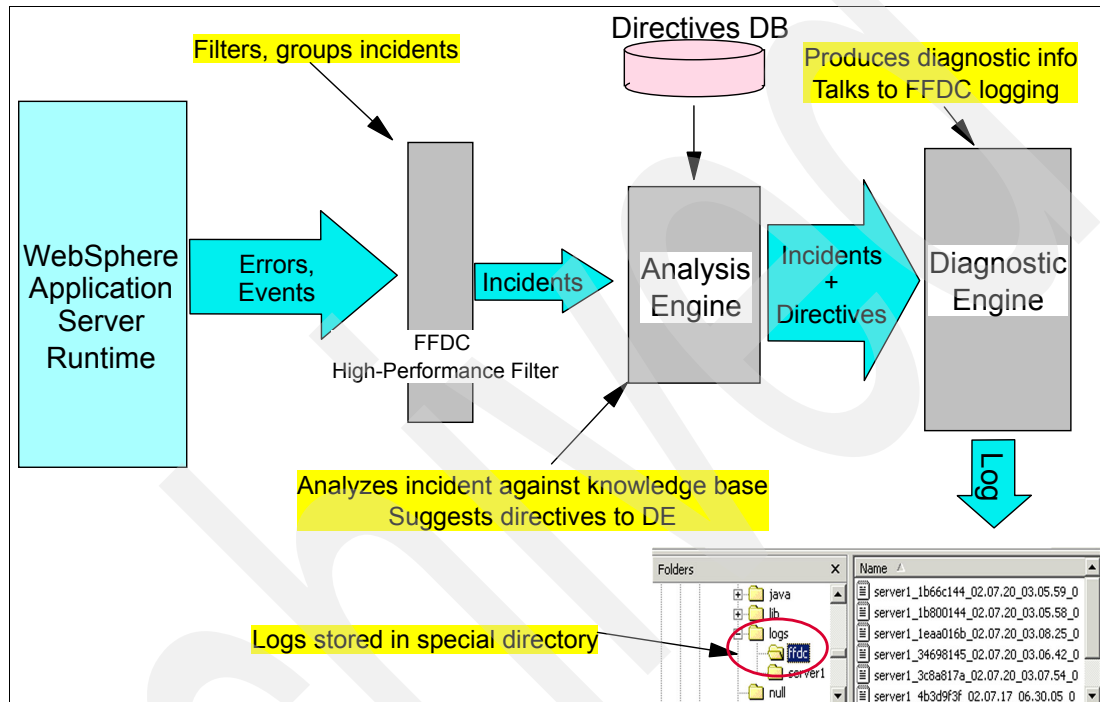


Figure 19-1 FFDC tool architectural overview

The servant regions of WebSphere for z/OS have been instrumented with calls to the FFDC tool. When an exception occurs, the event is passed through the error filter to the diagnostic engine. If the analysis engine is enabled, it retrieves any information that relates to the event from the symptom database.

Each component of WebSphere Application Server for z/OS contains a symptom database for the FFDC tool. This symptom database is located in the <install_root>/properties/logbr/ffdc/adv/ directory installed under all members of the cell, including the deployment manager and all installed nodes. When the analysis engine tool is run against an exception log, the symptom database, `ffdcdb.xml`, is used to try to provide a solution to the exceptions that are captured by the FFDC tool. The analysis engine uses the name, the source ID, and the probe ID from the exception as the key to the symptom database. If a match exists, the solution is displayed after the analysis engine is run.

19.3.2 How to set up the FFDC tool

There is no centralized control of the FFDC tool in WebSphere for z/OS. It must be activated for each member in the cell. This activation is done through the `ffdcRun.properties` file, which is found in the properties directory for all members of a z/OS cell. This includes the deployment manager in a Network Deployment installation and all nodes in the cell.

If a particular cell member is of interest during a runtime problem determination session, only that particular FFDC capability has to be activated. The only property that needs to be reset in the `ffdcRun.properties` file is the Level property. For a WebSphere for z/OS V5 installation, the Level property is set to a value of “0” by default. Figure 19-2 shows the possible Level values and their effects.

```
# Level of processing to perform
# 0 - none
# 1 - monitor exception path
# 2 - dump the call stack, with no advanced processing
# 3 - 2, plus object inspecting the current object
# 4 - 2, plus use DM to process the current object
# 5 - 4, plus process the top part of the call stack with DMs
# 6 - perform advanced processing the entire call stack
```

Figure 19-2 *ffdcRun.properties* level values

In this example, the default value is set to 4. The authors recommend that you use the same value as a starting point. However, if you want to set it to another logging level, use the following process:

1. Locate the `ffdcRun.properties` file for the address space of interest.
2. Open the `ffdcRun.properties` file for editing in any ASCII-capable editor.
3. Set the Level key of the `ffdcRun.properties` to the level of your choice (see Figure 19-2).
4. Save the `ffdcRun.properties` file.
5. Restart the address space.

Important: In z/OS, each address space produces its own set of FFDC files. Therefore, it is important to determine which address space is of interest during the runtime debug process to enable the FFDC tool for that specific address space.

19.3.3 FFDC output and its interpretation

The FFDC tool produces a set of files including:

- ▶ An index file that references all of the exceptions logged by FFDC (Example 19-6)
- ▶ An exception file for each exception type from each probe

Example 19-6 *FFDC index file*

Index	Occur	Time of last Occurrence	Exception	SourceId	ProbeId
ences					
1	1	04.09.27 14:58:58:371 GMT	java.lang.IllegalStateException		
			com.ibm.ws.webcontainer.servlet.ServletManager.doService	3891	
2	1	04.09.27 14:58:58:435 GMT	java.lang.IllegalStateException		
			com.ibm.ws.webcontainer.servlet.ServletInstance.service	2821	
3	1	04.09.27 14:58:58:391 GMT	java.lang.IllegalStateException		
			com.ibm.ws.webcontainer.servlet.StrictLifecycleServlet._service	1901	
4	334	04.09.27 14:58:59:577 GMT	java.lang.ClassNotFoundException		
			com.ibm.ws.classloader.CompoundClassLoader.loadClass	248	
5	1	04.09.27 14:58:58:553 GMT	java.lang.IllegalStateException		
			com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.handleWebAppDispatch	746 6	1
		04.09.27 14:58:58:590 GMT			
			com.ibm.ws.webcontainer.servlet.exception.UncaughtServletException		
			com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.dispatch	428	

The index file uses a naming convention of *<server name>_exception.log* and has the following columns:

Index	This column is used to determine the number of rows in the table. A plus sign (+) in front of this value signifies that this value has been updated since the last persistence of the information to the file system.
Occurrences	This column signifies the number of times that the exception has occurred.
Time of last Occurrence	A time stamp of the last occurrence of this exception.
Exception	The name of the Java exception class that was captured by the FFDC tool.
SourceId	The unique identifier for the exception source.
ProbeId	The unique identifier for the probe used in the data capture.

The exception file in Example 19-7 was produced by setting the Level to 4 in the *ffdcRun.properties* file. It has the stack trace for a *java.lang.IllegalStateException* and a dump of the *this* object and its properties. This was not captured in the z/OS system log, so the information captured by FFDC was over and above the normal logging.

Example 19-7 FFDC exception file

```

-----Start of DE processing----- = [04.09.27 14:58:58:371 GMT] , key = java.lang.IllegalStateException
com.ibm.ws.webcontainer.servlet.ServletManager.doService 3891
Exception = java.lang.IllegalStateException
Source = com.ibm.ws.webcontainer.servlet.ServletManager.doService
probeid = 3891
Stack Dump = java.lang.IllegalStateException: Context has not been prepared for next connection
    at com.ibm.ws.webcontainer.srt.Ni1SRPConnection.getHeaderNames(SRTConnectionContext.java:482)
    at com.ibm.ws.webcontainer.srt.SRTServletRequest.prepareHeader(SRTServletRequest.java(Compiled Code))
    at com.ibm.ws.webcontainer.srt.SRTServletRequest.getHeader(SRTServletRequest.java:307)
    at
    .
    .
    .
    at com.ibm.ws390.orb.ORBEJSBridge.invoke(ORBEJSBridge.java:170)

Dump of callerThis =
Object type = com.ibm.ws.webcontainer.servlet.StrictServletInstance
com.ibm.ws.webcontainer.servlet.StrictServletInstance@4045fee5

Exception = java.lang.IllegalStateException
Source = com.ibm.ws.webcontainer.servlet.ServletManager.doService
probeid = 3891
Dump of callerThis =
Object type = com.ibm.ws.webcontainer.servlet.StrictServletInstance
class$com$ibm$ws$webcontainer$servlet$StrictServletInstance =
    serialPersistentFields = {}
    serialVersionUID = 3206093459760846163
    allPermDomain = null
    getPDperm = null
    have_extensions = true
    _servicingCount = 0
    _servletClassname = com.ibm.ws.cache.servlet.ServletWrapper
    _servletName = action
    _servlet =
        class$com$ibm$ws$cache$servlet$ServletWrapper =

```

```

        serialPersistentFields =
this.class$.com.ibm.ws.webcontainer.servlet.StrictServletInstance.serialPersistentFields
        serialVersionUID = 3206093459760846163
        allPermDomain = null
        getPDperm = null
        have_extensions = true
        applicationUnAvailList =
        class$.java.lang.Object = null
        size = 0
        elementData = [Ljava.lang.Object;@3b07e96
        serialVersionUID = 8683452581122892189
        modCount = 0
        firstTime = true
        wrapsCacheableServlet = false
cacheEntrySet = true
        cacheEntry = null
        proxied =
            definitionsFactory = org.apache.struts.tiles.definition.ReloadableDefinitionsFactory@25807ee1
            lStrings = java.util.PropertyResourceBundle@743b3e83
            LSTRING_FILE = javax.servlet.http.LocalStrings
            HEADER_LASTMOD = Last-Modified
            HEADER_IFMODSINCE = If-Modified-Since
            METHOD_TRACE = TRACE
            METHOD_PUT = PUT
            METHOD_POST = POST
            METHOD_OPTIONS = OPTIONS
            METHOD_GET = GET
            METHOD_HEAD = HEAD
            METHOD_DELETE = DELETE
config = this._config
        tc =
            ivLogger = null
            ivResourceBundleName = com.ibm.ws.cache.resources.dynacache
            ivDumpEnabled = false
            defaultMessageFile = com.ibm.ejs.resources.seriousMessages
            ivEntryEnabled = false
            ivEventEnabled = false
            ivDebugEnabled = false
            ivName = com.ibm.ws.cache.servlet.ServletWrapper
        tc =
            ivLogger = null
            ivResourceBundleName = com.ibm.ejs.resources.seriousMessages
            ivDumpEnabled = false
            defaultMessageFile = com.ibm.ejs.resources.seriousMessages
            ivEntryEnabled = false
            ivEventEnabled = false
            ivDebugEnabled = false
            ivName = com.ibm.ws.webcontainer.servlet.StrictServletInstance
syncObject = java.lang.Object@40457ee5
servicingCount = 1
_implementsSTM = false
_config =
    _servletName = action
_initParams =
    hexDigit = [C@523dfb5
    whiteSpaceChars =
    specialSaveChars = =:
    #!
    strictKeyValueSeparators = =:
    keyValueSeparators = =:

```

```

defaults = null
serialVersionUID = 4112578634029874840
class$java$util$Hashtable$Entry = java.lang.Class@68cbe4b
emptyIterator = java.util.Hashtable$EmptyIterator@522cbeb5
emptyEnumerator = java.util.Hashtable$EmptyEnumerator@522d3eb5
ENTRIES = 2
VALUES = 1
KEYS = 0
values = null
entrySet = null
keySet = null
modCount = 11
loadFactor = 0.75
threshold = 17
count = 10
table = [Ljava.util.Hashtable$Entry;@40657ee5
_servletContext = com.ibm.ws.webcontainer.webapp.WebApp@16503e92
_unavailableUntil = -1
_servicingState =
  _instance = this._servicingState
_state =
  _instance = this._state
PERMANENTLY_UNAVAILABLE_FOR_SERVICE_STATE =
  _instance = this.PERMANENTLY_UNAVAILABLE_FOR_SERVICE_STATE
UNAVAILABLE_FOR_SERVICE_STATE =
  _instance = this.UNAVAILABLE_FOR_SERVICE_STATE
AVAILABLE_FOR_SERVICE_STATE = this._servicingState
ERROR_STATE =
  _instance = this.ERROR_STATE
DESTROYED_STATE =
  _instance = this.DESTROYED_STATE
DESTROYING_STATE =
  _instance = this.DESTROYING_STATE
STM_SERVICING_STATE =
  _instance = this.STM_SERVICING_STATE
SERVICING_STATE = this._state
IDLE_STATE =
  _instance = this.IDLE_STATE
INITIALIZING_STATE =
  _instance = this.PRE_INITIALIZED_STATE
PRE_INITIALIZED_STATE =
  _instance = this.PRE_INITIALIZED_STATE

```

The information that was dumped from the *this* object provides extra context information for the stack, including the member variables, calling objects, and so on. This information is of limited value when you are doing a problem determination by inspection. However, it can provide the analysis engine with valuable information when the exception log can be correlated to the symptom database. The real value of this exception log is the capture of the exception, during runtime, of the exception stack trace.

The combination of the exception, source ID, and the probe ID form an index key that is used to identify the exception log that has the FFDC exception information.

The exception file for each exception uses a naming convention of *<server name>_<thread id>_yy.MM.dd_HH.mm.ss_<unique id>.txt* and contains information that is relative to the value of the `ffdcRun.properties` Level property value. The higher the value of the Level property, the greater the amount of information in the exception file. Refer to 19.3.2, “How to set up the FFDC tool” on page 220 for an explanation of the information that is produced by each logging level.

After it is enabled, the FFDC tool produces the index and exception logs that are associated with the address space and persist to the <install base>/logs/ffdc directory. Retrieval of these log files can be done by using an FTP client from any other environment. For example, the index and exception logs could be retrieved with the ASCII setting for the FTP client on a Microsoft Windows host. Because the index and exception logs are text files, they can be viewed in any ASCII-capable text editor or viewer.

19.3.4 Example: Using the FFDC tool for problem determination

This section describes a real-life scenario where the FFDC tool was used to resolve an issue at a customer site.

We encountered the following problem: When multiple administrators were logged on to the administrative console, a synchronization error for a shared resource occurred as a result of contention over a shared context object. The symptom was an HTTP 500 error on one of the administrative consoles.

We ran the FFDC tool and captured this exception:

```
java.lang.IllegalStateException: Context has not been prepared for next connection
```

This information signified that an investigation of a thread synchronization issue was appropriate.

The FFDC tool was enabled, as described in 19.3.2, “How to set up the FFDC tool” on page 220. In particular, the `ffdcRun.properties` file was edited and the `Level` property was changed from a value of `0` to a value of `4` for this problem determination session. After the value was changed, the edited file was put in place and the deployment manager address space was restarted.

Preliminary investigation

Having retrieved the ASCII FFDC index file and exception logs, we followed several steps to determine that a `java.lang.IllegalStateException` had occurred:

1. We opened the `dmgr_exception.log` with WordPad to format the data and performed a visual inspection. It was clear that the `IllegalStateException` was of interest. We noted that the `java.lang.IllegalStateException` was originating from multiple classes. This is because the exception was being trapped by the FFDC tool as it traversed the call stack.
2. We began to inspect the exception logs that the FFDC tool had produced and that we had retrieved from the z/OS system. Here is where we noted a weakness between the index file and the naming of the exception logs: It was not clear which exception log contained the `java.lang.IllegalStateException` without opening each file and inspecting them.
3. Once again, we used WordPad on a Windows client to open each of these files in a formatted manner. The set of files had the names shown in Figure 19-3.








 dmgr_4c1efeb2_04.09.27_14.22.51_0.txt	45 KB	Text Document
 dmgr_4c18beb2_04.09.27_14.58.58_0.txt	9 KB	Text Document
 dmgr_4c18beb2_04.09.27_14.58.58_1.txt	9 KB	Text Document
 dmgr_4c18beb2_04.09.27_14.58.58_2.txt	15 KB	Text Document
 dmgr_4c18beb2_04.09.27_14.58.58_3.txt	15 KB	Text Document
 dmgr_4c18beb2_04.09.27_14.58.58_4.txt	13 KB	Text Document
 dmgr_exception.log	19 KB	Text Document

Figure 19-3 Index and exception Logs

As you can see, the dmgr_4c8beb2_04.09.27_14.58.58_0.txt file had the first instance of the java.lang.IllegalStateException.

Note: The authors edited Figure 19-3 to eliminate all occurrences of the com.ibm.ws.classloader.CompoundClassLoader.loadClass exception. Only the last of these exceptions is shown.

The information in the exception log also includes class and state information. This interpretation of the information beyond the stack trace is meant to be used by IBM to debug the problem and has little meaning outside of the IBM support network. Therefore, forward the exception logs to IBM support to obtain further information about the captured exception.

Running the FFDC analysis engine on the exception log

In Example 19-8, you can see the results of running the analysis engine against the dmgr_4c18beb2_04.09.27_14.58.58_0.txt exception log with the java.lang.IllegalStateException exceptions.

Example 19-8 Analysis engine output

```
C:\ffdclogs>java -classpath c:/WebSphere/AppServer/lib/ffdc.jar -Djava.
ext.dirs=c:/WebSphere/AppServer/lib com.ibm.ws.ffdc.AnalysisEngineTool
dmgr_4c18beb2_04.09.27_14.58.58_0.txt

c:/WebSphere/AppServer/properties/logbr/ffdc/adv/ffdcdb.xml
-----
Key Value : java.lang.IllegalStateException com.ibm.ws.webcontainer.servlet.ServletManager.doService 3891
Exception Name : java.lang.IllegalStateException
Solution : ***** NOT FOUND -- See attached call stack for more details about the problem.
java.lang.IllegalStateException com.ibm.ws.webcontainer.servlet.ServletManager.doService 3891
Stack Dump = java.lang.IllegalStateException: Context has not been prepared for next connection
at com.ibm.ws.webcontainer.srt.NilSRPConnection.getHeaderNames(SRTConnectionContext.java:482)
    at com.ibm.ws.webcontainer.srt.SRServletRequest.prepareHeader(SRServletRequest.java(Compiled
Code))
    at com.ibm.ws.webcontainer.srt.SRServletRequest.getHeader(SRServletRequest.java:307)
.
.
.
    at com.ibm.ws390.orb.ORBEJSBridge.invoke(ORBEJSBridge.java:170) *****
```

In this case, the analysis engine did not find a solution in the symptom database, as reported by this statement:

Solution: ***** NOT FOUND

The conclusion of the analysis, by inspection, resulted in further investigation by the WebSphere development team. Based on the java.lang.IllegalStateException, the team studied the synchronized code that was responsible for managing the administrative console and determined that a race condition existed for the shared connection resource. The code was then updated so that the appropriate synchronization occurred for the shared resource.

It took less than one hour to run the FFDC tool in the customer environment and, in this case, it provided an important clue in resolving the problem.

19.4 The Java Logging API

Developing applications and maintaining them are complex tasks. When a running application encounters an unexpected condition, it might not be able to complete a requested operation. In such a case, you want the application to inform the administrator that the operation has failed and explain why. Those who develop or maintain applications need to gather detailed information relating to the execution path of a running application to determine the root cause of a failure that is caused by a code bug. The facilities that are used for these purposes are typically referred to as message logging and diagnostic trace. The Java Logging API implements this facility.

Previous versions of WebSphere for z/OS exposed an API called JRas. Java logging and JRas have similar functionality, but the Java Logging API makes application logging portable to other compliant containers. The JRas API is still exposed to applications.

Internally, WebSphere for z/OS still uses JRas in conjunction with Java Logging, although that functionality is deprecated in V6.

WebSphere also supports Jakarta Commons-Logging, which defines a common programming model for logging and framework that enables applications to bind different logger implementations. It provides the APIs plus thin configuration file implementations.

For more detailed information about Jakarta Commons-Logging, refer to “Jakarta Commons Logging” at the WebSphere for z/OS Information Center:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

19.4.1 What is the Java Logging API and when to use it

The Java Logging API is a standard logging API for your applications that is provided by the `java.util.logging` package.

One of the drawbacks of using `System.out.println` statements is that the application only has two channels: Standard output or Standard error. Piping all debugging messages into a single point results in a deficiency of error granularity. Once an application goes into production, many of the logging messages can be either redundant or unnecessary, causing production logs to become extremely large.

Java Logging does not distinguish between tracing and message logging, although previous versions of WebSphere for z/OS have made a clear distinction between them. In WebSphere for z/OS, the differences between tracing and message logging are:

- ▶ Tracing messages are messages with lower severity.
- ▶ Tracing messages generally are not localized.
- ▶ When tracing is enabled, a much higher volume of messages is produced.
- ▶ Tracing messages provide information for problem determination.

WebSphere for z/OS and application code use the `Logger` object to put data into the logstream. The `Logger` objects can use one or more `Handler`. Each `Handler` is an output device. The `Filters` are used to reduce the amount of information sent to the stream. The `Formatter` is used to format log data. Figure 19-4 on page 228 shows the elements of the Java Logging Architecture in context.

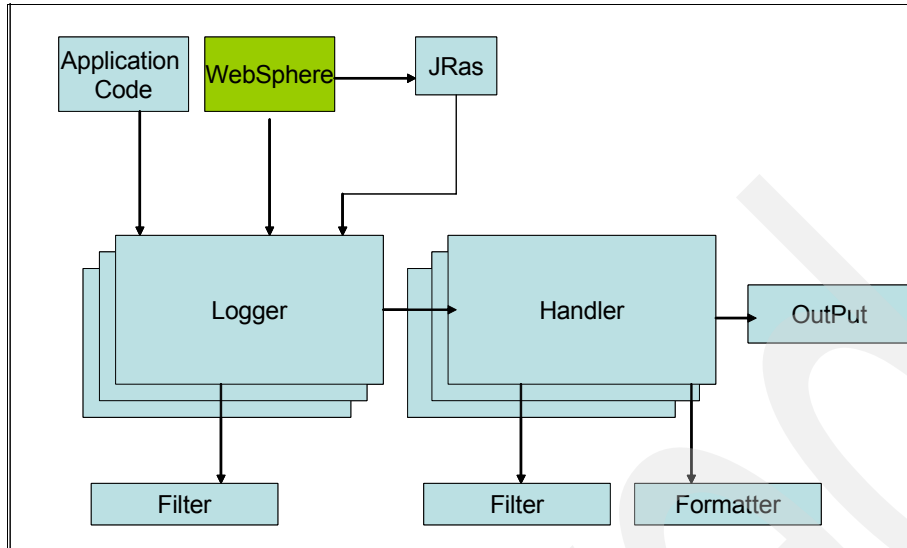


Figure 19-4 Java logging architecture

19.4.2 Setting up the Java Logging API

To collect traces and logs of WebSphere for z/OS using the Java Logging API:

1. Log in to the Administrative Console.
2. In the navigation pane, click **Servers** and **Application Servers**.
3. Click the name of the server that you want to work with.
4. Click **Diagnostic Trace Service**.
5. Select **Enable Log**.
6. Select either **Memory buffer** or **file**.
7. Specify your configurations.

Figure 19-5 on page 229 shows **Enable log in Diagnostic Trace Service** and options to enable logs and trace.



Figure 19-5 Enable log in Diagnostic Trace Service

The difference between V6 and older versions of WebSphere for z/OS is that the log level string has its own panel in V6. To set traces and log level details:

1. In the navigation pane, click **Servers** and **Application Servers**.
2. Click the name of the server that you want to work with.
3. Under Troubleshooting, select **Logging and tracing**.
4. Click **Change Log Detail levels**.
5. To make a static change to the configuration, click the Configuration tab. A list of well-known components, packages, and groups is displayed.
6. To change the configuration dynamically, click the Runtime tab.
7. Select a component, package, or group to set a logging level.
8. Click **Apply**.
9. Click **OK**.

Figure 19-6 on page 230 shows the Administrative Console panel for changing Log Levels Details. The list of components, packages, and groups shows all the components that are currently registered to the running server.

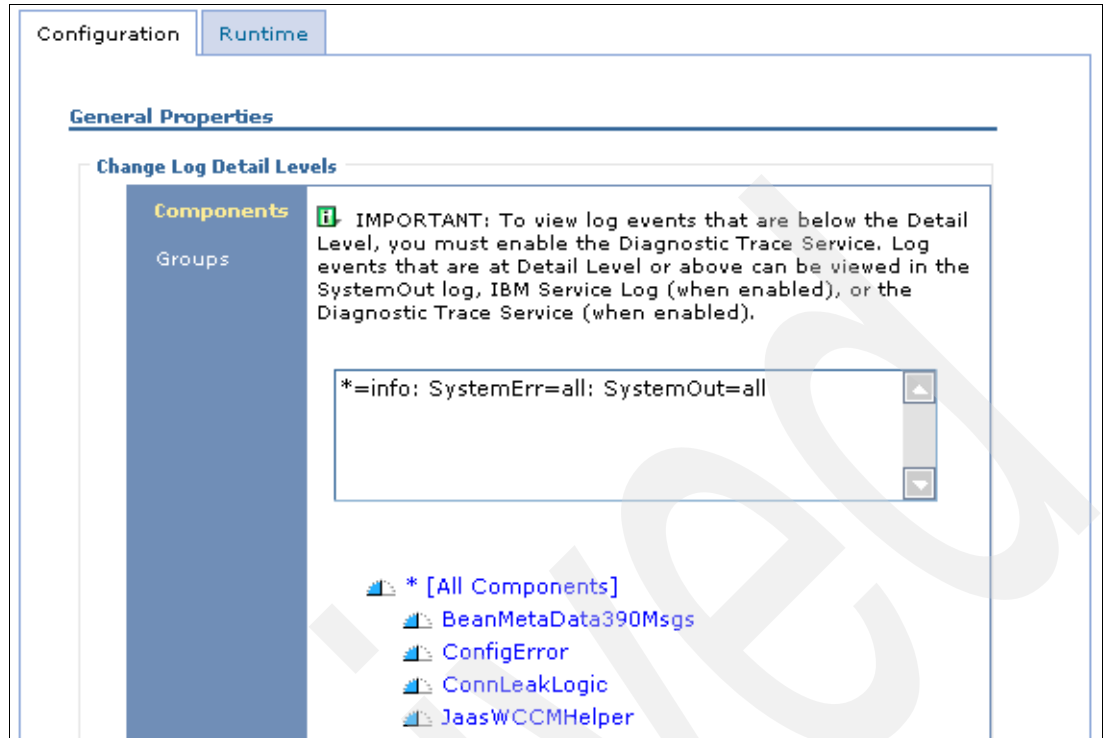


Figure 19-6 WebSphere Change Log Detail Levels

The default log level is *=info. To modify it, you can type another level or set it using the graphical menu.

Table 19-2 describes the fields in the first line of the trace.

Table 19-2 Log Details Level

Level	Consequence
Off	No events are logged
Fatal	Task cannot continue and component cannot function
Severe	Task cannot continue but component can still function
Warning	Potential error or impending error
Audit	Significant event affecting server state or resources
Info	General information outlining overall task progress
Config	Configuration change or status
Detail	General information detailing subtask progress
Fine	Trace information: General trace
Finer	Trace information: Detailed trace
Finest	Trace information: A more detailed trace (includes all the details that are needed to debug problems)
All	All events are logged. Inclusive custom logs

The syntax of strings that are used in the log detail level is specified by the Java Logging specification. The string is defined by the component or group that you want to trace, followed by an equal sign (=) and the level for detail. For example, to enable fine trace level for all classes in the com.ibm.ws.classloades package, use:

```
com.ibm.ws.classloader.*=fine
```

To enable detailed trace level for all components in the EJBContainer group, use:

```
EJBContainer=finest
```

Note: Tracing components have an impact on performance. A trace of all components (com.ibm.*=all) causes the system to run very slowly.

19.4.3 Java Logging output and interpretation

The logging output differs depending on the parameters that you set in the administrative console. It has a heading (first line) and a brief description (second line) in each entry of the trace. Example 19-9 shows the records of a sample entry in a trace.

Example 19-9 An entry in a trace

```
Trace: 2005/08/03 13:43:44.723 01 t=7CB4F8 c=UNK key=P8 (0000000A)
  Description: Log Boss/390 Error
  from filename: ./bborjtr.cpp
  at line: 932
  error message: BB000222I: TRAS0018I: The trace state has changed. The new trace state is
  *=config.
```

You can see the fields listed in Table 19-3 in the first line.

Table 19-3 First line in trace sample

2005/08/03 13:43:44.723	Date and hour of the entry in the trace.
01	Version ID
t=7CB4F8	TCB
c=UNK	Represents correlation information that consists of internal runtime information (session ID and request ID) that is used to identify trace entries related to a particular client request
key=P8	Represents which state and key the code is running in, for example, code running in a control process is running in supervisor state, key 2 (s2) and code running in a servant process is in problem state, key 8 (p8)
(0000000A)	Trace point ID that is used to locate trace in code and which follows the cmmmttt, structure, where cc is the component ID from include/private/bborras.h, mmm identifies the module in the component in include/private/ras/bboXcrd.h, and ttt is the unique trace point within the source (this value is in hex; the value in the code is decimal)

In the second line, you can see a brief description of the entry recorded in the trace:

```
Description: Log Boss/390 Error
```

Table 19-2 on page 230 lists the events and description.

19.4.4 Java Logging API example

In Example 19-10, you can see the log of a change detail trace level when the Java Logging API was used.

Example 19-10 Change log detail level

```
Trace: 2005/08/03 14:58:22.257 01 t=7C9690 c=UNK key=S2 (13007002)
  ThreadId: 000001d7
  FunctionName: com.ibm.ejs.ras.ManagerAdmin
  SourceId: com.ibm.ejs.ras.ManagerAdmin
  Category: INFO
  ExtendedMessage: BB000222I: TRAS0018I: The trace state has changed. The new trace state
is *=config:com.ibm.ws.db2.logwriter=fine:
com.ibm.ws.database.logwriter=fine:itso.db2.cmp.j2ee.*=fine:com.ibm.ws.webcontainer.servlet
.ServletWrapper=fine:com.ibm.ws.webcontainer.srt.SRTServletRequest=fine:com.ibm.ws.webconta
iner.srt.SRTServletResponse=fine.
```

19.5 IBM HTTP Server logs and trace

IBM HTTP Server writes various kinds of logs into HFS files. Depending on the specific function, various logs are provided:

- ▶ Error log
- ▶ Access log
- ▶ Fast Response Cache Accelerator (FRCA) access log
- ▶ Proxy and cache access logs
- ▶ Agent log
- ▶ Referrer log
- ▶ CGI error log

We focused on the server error log and the server access log, which are particularly useful for WebSphere for z/OS problem determination.

It is also possible to activate the IBM HTTP Server trace called very verbose trace (-vv trace) to collect the activity of the server. This trace is written directly in the job log of the task started by IBM HTTP Server.

The Web server plug-in in IBM HTTP Server is used to forward requests to a WebSphere for z/OS Web container. This HTTP plug-in component usually writes out logs and traces for informational or problem diagnosis use.

Figure 19-7 on page 233 shows the HTTP server logs and traces in relation to WebSphere for z/OS.

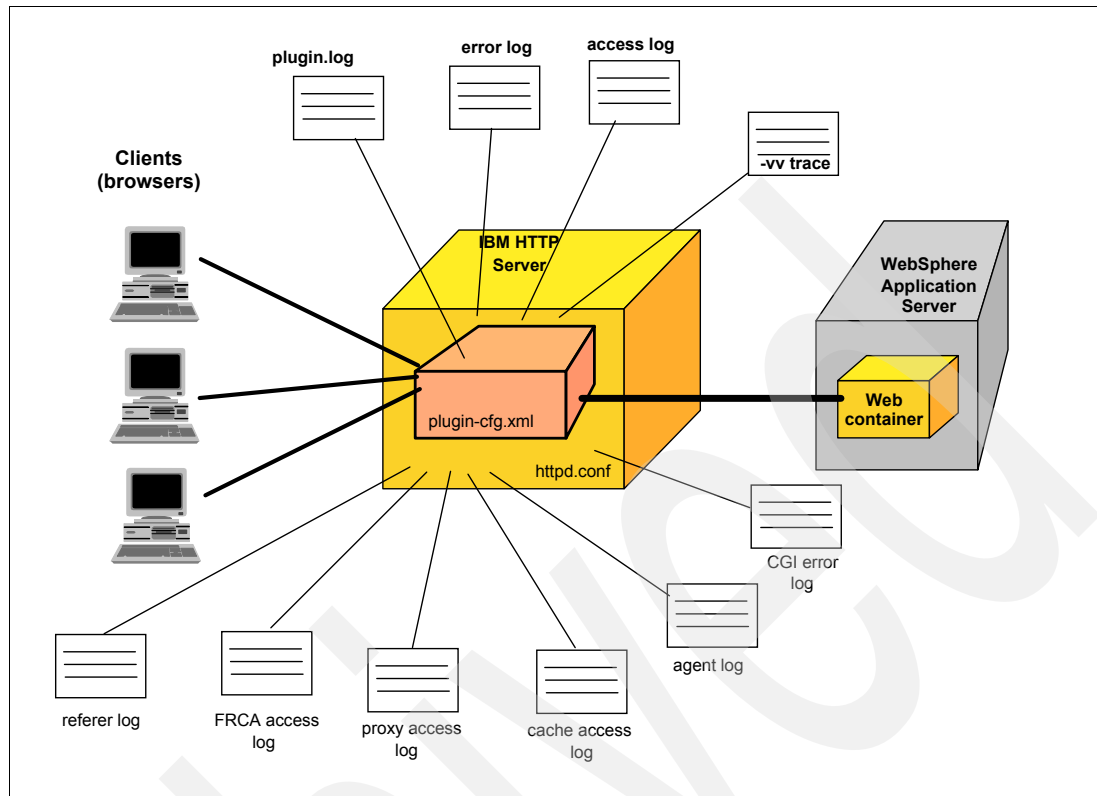


Figure 19-7 IBM HTTP Server logs and trace

19.5.1 Server error log

The IBM HTTP Server creates an error log that includes errors that were encountered by the clients for the server, such as timing out or not getting access. After you determine that there are problems (for example, the client and the server are not communicating), you should refer to this log because it might indicate what is wrong. The server saves the error log in an HFS file. You can use a logging directive to direct the path and the name of this file to the area where you want to log internal server errors. It is called `httpd.conf` by default. The name of the directive is `ErrorLog` as follows:

```
ErrorLog /web/logs/errorlog
```

When it creates the file, the server uses the file name that you specify and appends a date suffix. Figure 19-8 illustrates a section of a server error log with each number representing a different field.

```
[30/Sep/2004:11:47:26+0400] [IMV0193I OK] [host: 9.12.6.160] /
[30/Sep/2004:11:47:38+0400] [IMV0210E MULTI FAILED] [host: 9.12.6.160] /mytest
[30/Sep/2004:11:59:41+0400] [IMV0210E MULTI FAILED] [host: 9.12.6.160] /testapp
[30/Sep/2004:12:00:44+0400] [IMV0210E MULTI FAILED] [host: 9.12.6.160] /mytest/
[30/Sep/2004:13:01:43+0400] [IMV0193I OK] [host: 9.12.6.160] /IBMTods/testapp
[30/Sep/2004:13:02:32+0400] [IMV0210E MULTI FAILED] [host: 9.12.6.160] /IBMTod/testapp
[30/Sep/2004:13:06:49+0400] [IMV0193I OK] [host: 9.12.6.160 referer: http://wls049.itso.ibm.com:9508/IBMTods/] /IBMTods/EBizHiCount
[30/Sep/2004:13:06:59+0400] [IMV0193I OK] [host: 9.12.6.160 referer: http://wls049.itso.ibm.com:9508/IBMTods/] /IBMTods/EBizSuperSnoop
```

The fields (delineated by numbers in the illustration) in the server error log represent the following information:

1. Date and time when the entry of the request was recorded in the server
2. Error message (a description of this message is provided in *IBM HTTP Server Planning, Installing, and Using*, SC34-4826)
3. The IP address of the client that accessed the server
4. The URL that the client requested
5. The context root and the file requested by the client

Note: If you access the server from a PC client, the IP address in the server error log might not be the same as the IP address of your PC. This depends on the network configuration that you use (proxies, gateways, and so on).

19.5.2 Server access log

The IBM HTTP Server records activities in the access log files and stores them each day. Each day at midnight, the server closes the current access log and creates a new access log file for the next day. The access log has entries for page requests that were made to the server.

For each access request your server receives, an entry is made in the access log showing:

- ▶ What was requested
- ▶ When it was requested
- ▶ Who requested it
- ▶ The method of the request
- ▶ The type of file that your server sent in response to the request
- ▶ The return code, which indicates whether the request was honored

To enable the server access log, set another logging directive in the httpd.conf file called `AccessLog` that points to the file where you want the access log to be saved as follows:

```
AccessLog /web/logs/accesslog
```

The server then creates the file in `ServerRoot/web/logs/`:

```
accesslog.datesuffix.file_extension
```

`ServerRoot` is a path that can be configured with another directive in the httpd.conf, and `datesuffix.file_extension` is the date when the log was created with the Web server-generated file extension.

The server records one line per request that arrives. Figure 19-9 on page 235 illustrates the fields that each line contains in an example.

```

9.12.6.160 -- [30/Sep/2004:11:47:26 +0400] "GET / HTTP/1.1" 403 282
9.12.6.160 -- [30/Sep/2004:11:47:38 +0400] "GET /mytest HTTP/1.1" 404 375
9.12.6.160 -- [30/Sep/2004:11:51:45 +0400] "GET /IBMTools/ HTTP/1.1" 500 310
9.12.6.160 -- [30/Sep/2004:11:51:53 +0400] "GET /IBMTools/ HTTP/1.1" 304 0
9.12.6.160 -- [30/Sep/2004:11:51:53 +0400] "GET /IBMTools/rbhome.gif HTTP/1.1" 304 0
9.12.6.160 -- [30/Sep/2004:11:57:49 +0400] "GET /IBMTools/EBizHitCount HTTP/1.1" 200 1070
9.12.6.160 -- [30/Sep/2004:11:59:41 +0400] "GET /testapp HTTP/1.1" 404 375
9.12.6.160 -- [30/Sep/2004:12:00:02 +0400] "GET /IBMTools HTTP/1.1" 500 308
9.12.6.160 -- [30/Sep/2004:12:00:44 +0400] "GET /myTest/ HTTP/1.1" 404 375
9.12.6.160 -- [30/Sep/2004:12:01:06 +0400] "GET /IBMTools/EBizHitCount HTTP/1.1" 200 1175
9.12.6.160 -- [30/Sep/2004:13:01:43 +0400] "GET /IBMTools/testapp HTTP/1.1" 404 -
9.12.6.160 -- [30/Sep/2004:13:02:32 +0400] "GET /IBMTool/testapp HTTP/1.1" 404 375
9.12.6.160 -- [30/Sep/2004:13:03:11 +0400] "GET /IBMTools/EBizSuperSnoop HTTP/1.1" 200 14023
9.12.6.160 -- [30/Sep/2004:13:06:49 +0400] "GET /IBMTools/EBizHitCount HTTP/1.1" 404 -
9.12.6.160 -- [30/Sep/2004:13:06:59 +0400] "GET /IBMTools/EBizSuperSnoop HTTP/1.1" 404 -

```

Figure 19-9 Server access log sample

The numbered fields represent the following information:

1. The IP address of the client that made the request
2. The date and time of the request
3. The method of the request
4. The file that the client requested
5. The protocol and version
6. The value of the HTTP return code
7. The size of the file (in bytes) being requested

For more information about the server logs, see *IBM HTTP Server Planning, Installing, and Using*, SC34-4826, especially Chapter 15.

19.5.3 Very verbose trace

The server trace has several levels of debugging (verbose, much too verbose, verbose cache, and debug). The most common is the -vv (very verbose) trace.

Note: Activation of the -vv trace results in a large amount of information that is recorded in the job log. It directly impacts server performance.

There are two ways to start the -vv trace:

- Use the -vv parameter in the started procedure of the server, as shown in Example 19-11.

Example 19-11 Configuration of -vv trace in started procedure in PROCLIB

```

//IMWPROC PROC LEPRM=,ICSPARM='-vv -r /web/httpd.conf'
//*****
//WEBSRV EXEC PGM=IMWHTTPD,REGION=OK,TIME=NOLIMIT,
// PARM=(@&LEPRM/&ICSPARM@)
//*****
//SYSIN DD DUMMY
//OUTDSC OUTPUT DEST=HOLD
//SYSPRINT DD SYSOUT=*,OUTPUT=(*.OUTDSC)
//SYSERR DD SYSOUT=*,OUTPUT=(*.OUTDSC)

```

```
//STDOUT DD SYSOUT=*,OUTPUT=(*.OUTDSC)
//STDERR DD SYSOUT=*,OUTPUT=(*.OUTDSC)
//SYSOUT DD SYSOUT=*,OUTPUT=(*.OUTDSC)
//CEEDUMP DD SYSOUT=*,OUTPUT=(*.OUTDSC)
```

The server is in a very verbose mode when it is restarts. To stop the trace, you either change the parameter in the procedure and restart the server or issue a MODIFY command.

- Alternatively, dynamically modify the server with the following console command:

```
/f imwebsrv,appl=-vv
```

In the command, *imwebsrv* is the name of your IBM HTTP Server.

The following message appears in the console log:

```
30Sep04 10:28:08: IMW3518I Second level tracing (-vv) enabled.
```

To stop the trace, launch this command:

```
/f imwebsrv,appl=-nodebug
```

The following message appears in the console log:

```
30Sep04 10:38:38: IMW3508I Debug has been disabled for all modules.
```

Important: Because of the large amount of data that the -vv trace generates and the impact on performance, the authors recommend that you start the trace dynamically, reproduce the error, and then stop the trace. That way, you have a short -vv trace, which makes it easier to find the section of the log that relates to the problem.

The -vv trace provides more detailed information than the server error log or the access log. For this reason, the trace is more helpful if you determine that the problem occurred inside IBM HTTP Server and you need detailed step-by-step processing information to rectify the problem. Example 19-12 displays only a portion of the trace, showing a request of the /IBMTools/EBizSuperSnoop file from a browser with the following information:

- The method of the request
- The file requested (GET //IBMTools/EBizSuperSnoop)
- The protocol and version (HTTP/1.1)
- The browser and the operative system of the client (Mozilla 4.0; compatible Microsoft Internet Explorer 6.0; Microsoft Windows NT® 5.1)
- The IP address and port of the host (wtsc49.itso.ibm.com:9508)

Example 19-12 Very verbose trace sample

```
[21646C48 30/Sep/2004:14:06:12.854728]: 30Sep04 14:06:12: IMW3518I Second level tracing (-vv) enabled.
[21660778 30/Sep/2004:14:06:22.556968]: Read 460 bytes from socket 10.
[21660778 30/Sep/2004:14:06:22.557027]: After AcceptEx nAcceptThds: 75 and nSSLAcceptThds: 0.
[21660778 30/Sep/2004:14:06:22.557046]: server_loop... Accepted socket: 10.
[21660778 30/Sep/2004:14:06:22.557131]: KEEPALIVE... set.
[21660778 30/Sep/2004:14:06:22.557156]: HTSession... starting for socket=10; STHD=21932DD8
[21660778 30/Sep/2004:14:06:22.557187]: Keep-Alive.. Starting HTTPD 1.1 loop.
[21660778 30/Sep/2004:14:06:22.557218]: HTTimer... setting timer off->set (1) on socket 10.
[21660778 30/Sep/2004:14:06:22.557236]: HTTimer... set, old=0, cur=0, new=1
[21660778 30/Sep/2004:14:06:22.557314]: Client sez.. GET /IBMTools/EBizSuperSnoop HTTP/1.1
[21660778 30/Sep/2004:14:06:22.557340]: Protocol version.... 1.1
[21660778 30/Sep/2004:14:06:22.557355]: Persistent Connection has been established
[21660778 30/Sep/2004:14:06:22.557378]: Client sez.. Accept: /*/*
[21660778 30/Sep/2004:14:06:22.557395]: Accept..... /*/* (q=1.00,mxb=0.0,mxs=0.0)
```

```
[21660778 30/Sep/2004:14:06:22.557437]: Client sez.. Referer: http://wtsc49.itso.ibm.com:9508/IBMTools/
[21660778 30/Sep/2004:14:06:22.557456]: Referer..... http://wtsc49.itso.ibm.com:9508/IBMTools/
[21660778 30/Sep/2004:14:06:22.557476]: Client sez.. Accept-Language: en-us
[21660778 30/Sep/2004:14:06:22.557492]: Language.... en-us (q=1.00)
[21660778 30/Sep/2004:14:06:22.557517]: Client sez.. Accept-Encoding: gzip, deflate
[21660778 30/Sep/2004:14:06:22.557533]: Encoding.... gzip (q=1.00)
[21660778 30/Sep/2004:14:06:22.557552]: Encoding.... deflate (q=1.00)
[21660778 30/Sep/2004:14:06:22.557580]: Client sez.. If-Modified-Since: Thu, 30 Sep 2004 17:03:11 GMT
[21660778 30/Sep/2004:14:06:22.557600]: Format..... Wkd, 00 Mon 0000 00:00:00 GMT
[21660778 30/Sep/2004:14:06:22.557621]: TimeZone.... 04 hours from GMT
[21660778 30/Sep/2004:14:06:22.557638]: Time string. Thu, 30 Sep 2004 17:03:11 GMT; offset = 0 seconds
[21660778 30/Sep/2004:14:06:22.557662]: Parsed..... to 1096563791 seconds, Thu Sep 30 13:03:11 2004
[21660778 30/Sep/2004:14:06:22.557687]: Give only... if modified since (localtime) Thu Sep 30 13:03:11 2004
[21660778 30/Sep/2004:14:06:22.557722]: Client sez.. User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows
NT 5.1; .NET CLR 1.1.4322)
[21660778 30/Sep/2004:14:06:22.557746]: User-Agent.. Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;
.NET CLR 1.1.4322)
[21660778 30/Sep/2004:14:06:22.557771]: Client sez.. Host: wtsc49.itso.ibm.com:9508
[21660778 30/Sep/2004:14:06:22.557789]: Host..... wtsc49.itso.ibm.com
[21660778 30/Sep/2004:14:06:22.557804]: Host Port... 9508
```

19.5.4 HTTP plug-in log

The HTTP plug-in log usually has error messages regarding forwarded requests to WebSphere for z/OS and configuration errors in the plugin-cfg.xml file. In addition, in the plugin-cfg.xml file, you can turn on tracing to provide more details about the routing of a request, such as:

- ▶ URL and URI matching
- ▶ Virtual host matching
- ▶ Request header and session affinity cookie

These logs can help you determine where the problem exists during an initial search. It is useful to follow the request from a browser on the client side to WebSphere for z/OS, and also in the other direction, from WebSphere for z/OS to the client. By looking at the logs, you can determine if the requests from the client arrive correctly at IBM HTTP Server, and if they are mapped correctly according to the httpd.conf directives and plugin-cfg.xml routing table.

The -vv trace provides a level of messaging that can provide information about the problem if it is occurring in the server. If it is not occurring inside the server, the -vv trace can provide detailed information about what is “traveling” through it.

For more information about the server logs and the server trace, see *IBM HTTP Server Planning, Installing, and Using*, SC34-4826.

For more information about the HTTP plug-in log, see *WebSphere Application Server for z/OS V5.1: Servers and Environment*, GA22-7958.

To enable logging for the WebSphere z/OS plug-in, specify the location of the plugin-cfg.xml file in the httpd.conf file:

```
ServerInit /<...>/ihs390WAS50Plugin_http.so:init_exit /<...>/plugin-cfg.xml
```

If your plug-in initialized successfully, the following messages appear in SYSOUT, indicating which plugin-cfg.xml file is used by IBM HTTP Server:

```
WebSphere HTTP Plug-in for z/OS and OS/390 initializing with configuration file :
/web/andy1/plugin-cfg.xml
WebSphere HTTP Plug-in for z/OS and OS/390 initialization went OK :-)
```

Then, in the plugin-cfg.xml file, specify the LogLevel and Name of the plug-in log file (plugin.log) where all logging output should go, as shown:

```
<Log LogLevel="Error" Name="plugin.log"/>
```

Plug-in logging allows logging at many levels of detail to suit various situations. You can specify one of the following levels:

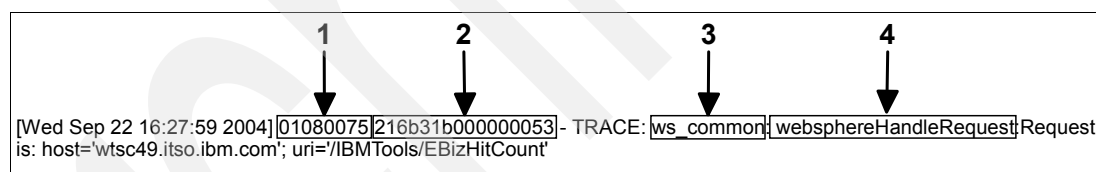
Trace	All of the steps in the request process are logged in detail.
Stats	The server selected for each request and other load balancing information that is related to request handling is logged.
Warn	All warning and error messages that result from abnormal request processing are logged.
Error	Only error messages that result from abnormal request processing are logged.

The default level of logging is Error.

Note: Specifying **LogLevel="Trace"** generates a large amount of data that might impact performance. The authors recommend that you specify **LogLevel="Error"**.

The server records one line per request that arrives. Figure 19-10 illustrates the fields in each record line:

1. Process ID
2. Pthread ID
3. IBM Software source code file name
4. Function name



```
[Wed Sep 22 16:27:59 2004]01080075216b31b000000053- TRACE: ws_common:websphereHandleRequest Request
is: host='wtsc49.itso.ibm.com'; uri='/IBMTools/EBizHitCount'
```

Figure 19-10 A plug-in trace record and some of the important fields

Figure 19-11 on page 238 shows the plug-in trace records that resulted from a request to find matches for the virtual host group (VhostGroup) and URI group (UriGroup).

```
TRACE: ws_common: websphereVhostMatch: Comparing 'wtsc49.itso.ibm.com:9508' to 'wtsc49.itso.ibm.com:9508' in VhostGroup: default_hos
TRACE: ws_common: websphereVhostMatch: Found a match 'wtsc49.itso.ibm.com:9508' to 'wtsc49.itso.ibm.com:9508' in VhostGroup: default
TRACE: ws_common: websphereVhostMatch: Comparing '*:9559' to 'wtsc49.itso.ibm.com:9508' in VhostGroup: default_host
TRACE: ws_common: websphereVhostMatch: Comparing '*:9549' to 'wtsc49.itso.ibm.com:9508' in VhostGroup: default_host
TRACE: ws_common: websphereVhostMatch: Comparing '*:9548' to 'wtsc49.itso.ibm.com:9508' in VhostGroup: default_host
TRACE: ws_common: websphereVhostMatch: Comparing '*:80' to 'wtsc49.itso.ibm.com:9508' in VhostGroup: default_host
TRACE: ws_common: websphereVhostMatch: Comparing 'wtsc49.itso.ibm.com:9519' to 'wtsc49.itso.ibm.com:9508' in VhostGroup: default_hos
TRACE: ws_common: websphereVhostMatch: Comparing 'wtsc49.itso.ibm.com:9518' to 'wtsc49.itso.ibm.com:9508' in VhostGroup: default_hos
TRACE: ws_common: websphereVhostMatch: Comparing '*:9519' to 'wtsc49.itso.ibm.com:9508' in VhostGroup: default_host
TRACE: ws_common: websphereVhostMatch: Comparing '*:9518' to 'wtsc49.itso.ibm.com:9508' in VhostGroup: default_host
TRACE: ws_common: websphereUriMatch: Comparing '/admin' to '/IBMTools/EBizHitCount' in UriGroup: default_host_dmgr_pddmnode_Cluster_
TRACE: ws_common: websphereUriMatch: Comparing '/admin/' to '/IBMTools/EBizHitCount' in UriGroup: default_host_dmgr_pddmnode_Cluste
TRACE: ws_common: websphereUriMatch: Comparing '/adminservlet' to '/IBMTools/EBizHitCount' in UriGroup: default_host_dmgr_pddmnode_C
TRACE: ws_common: websphereUriMatch: Comparing '/FileTransfer' to '/IBMTools/EBizHitCount' in UriGroup: default_host_dmgr_pddmnode_C
TRACE: ws_common: websphereUriMatch: Comparing '/adminservlet/' to '/IBMTools/EBizHitCount' in UriGroup: default_host_dmgr_pddmnode
TRACE: ws_common: websphereUriMatch: Comparing '/FileTransfer/' to '/IBMTools/EBizHitCount' in UriGroup: default_host_dmgr_pddmnode
TRACE: ws_common: websphereUriMatch: Failed to match: /IBMTools/EBizHitCount
```

Figure 19-11 Plug-in traces

The request that was used was:

`http://wtsc49.itso.ibm.com:9508/IBMTools/EBizHitCount`

The plug-in first attempts to find a match for the wtsc49.itso.ibm.com virtual host and port 9508 in the defined virtual host group. Then, it compares the /IBMTools/EBizHitCount URI with the defined URI group entries. The last line shows that there was no matching URI definition.

Archived

Archived

WebSphere for z/OS traces and dumps

In this chapter, we explain the following traces and dumps:

- ▶ CTRACE for WebSphere
- ▶ JDBC trace
- ▶ SVCDump
- ▶ CEEDUMP
- ▶ Java Transaction Dump
- ▶ Javadump
- ▶ Heapdump

For each trace or dump, we provide information about its nature, information about when to use it, and information about how to use it. We describe the output, show how to interpret it, and provide an example.

20.1 CTRACE for WebSphere

WebSphere for z/OS uses z/OS CTRACE facilities to manage the collection and storage of trace data. Unless you configure specific CTRACE controls, WebSphere for z/OS records trace data in address space buffers in private (pageable) storage. This data is not accessible unless a dump of the address space is taken.

CTRACE data is primarily output for the IBM support team to use, but you might be asked to provide IBM with CTRACE output. Therefore, it is important for you to know how to use CTRACE in your environment to obtain additional trace data that is available when a problem first occurs.

CTRACE efficiently uses system resources so that you can collect valuable trace data with a minimal impact on performance. WebSphere for z/OS identifies itself to CTRACE with the component name that is determined by the short cell name. With CTRACE, you can:

- ▶ Use one or more data sets for capturing trace data so that you can manage I/O more effectively.
- ▶ Merge multiple traces, including other components such as TCP/IP and z/OS USS, using IPCS.
- ▶ Write trace data to a data set rather than SYSPRINT, keeping spool space use to a minimum.
- ▶ Wrap trace data for better management of system resources.
- ▶ Funnel trace data from multiple address spaces to one data set, or have CTRACE send the trace data from each address space to separate data sets or the same one.
- ▶ Start and stop tracing without stopping and restarting WebSphere for z/OS address spaces.

20.1.1 Setting up and taking a CTRACE

The procedure for setting up and taking a CTRACE is available in *WebSphere Application Server for z/OS V6, Troubleshooting and support*, GA22-7964-03

More information is available if you search for CTRACE at the *WebSphere for z/OS Information Center*.

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

CTRACE output contains important information that helps IBM provide a high level of support. IBM might ask you to provide a CTRACE when the IBM Support Center requires a deeper level of information to analyze your problem.

20.1.2 Viewing CTRACE and JRas data through IPCS

After CTRACE is activated, WebSphere for z/OS writes trace data into memory buffers. The number and size of these buffers is controlled with WebSphere variables. You can get this trace data from a dump, which might be captured by the system or requested by the operator with DUMP or SLIP commands. To view messages or application trace data from a component trace, you must use the IPCS to format the data.

To use the IPCS dialog to format application trace data for error analysis, follow these steps:

1. From the IPCS Primary Option Menu panel, select option **6** (COMMAND).
2. In the IPCS Sub-command Entry panel:
 - a. Issue the SETDEF sub command to determine the default values for routing displays.
 - b. Enter the CTRACE command, with the following required parameters:

CTRACE COMP(cell_short_name)

cell_short_name is the value that is specified through the ISPF Customization Dialog to identify the location of server configuration files (eight or fewer characters and all uppercase).

If you are interested in only JRas data, enter the following command and specify additional parameters as necessary:

CTRACE COMP(cell_short_name)USEREXIT(JRAS)

For more details about CTRACE, see *z/OS MVS IPCS Commands*, SA22-7594.

3. View your application CTRACE data based on the options that you chose for the location of the data. Example 20-1 shows WebSphere for z/OS CTRACE output.

Example 20-1 WebSphere for z/OS CTRACE output

```
SY1      OBOAT008  04000002  00:14:57.268258  Dispatch Method
ASID.... 0039
TCB..... 009E34A0 PSW1.... 078D2400 SESS.... 00000008 REQI.... 0000006C
Class Name      = JPolicyEmSQLMO
Method Name     = _get_policyNo
object          = 0x260ED1F8
objectPtr refcount = 3          0x00000003
objectPtr classname= JPolicyEmSQLMO
```

An entry contains an undefined ID: 13007002 , hex format will be used.

```
SY1      N/A      13007002  00:14:57.272682  N/A
0002009E 34A0078D 24000039 00000008 | ..... |
0000006C 000C0302 97969389 83A8D596 | ...%....policyNo |
00120402 6DD1D796 938983A8 C2D6C994 | ...._JPolicyB0Im |
97930009 0A02C1E4 C4C9E300 2E0B02C2 | p1....AUDIT....B |
C2D6D1F0 F0F0F240 D7969389 83A84095 | B0J0002 Policy n |
A4948285 9940F3F3 6BF3F3F3 409682A3 | umber 33,333 obt |
81899585 844B4040 40 | ained. |
```

Trace: 2004/10/12 00:14:57.268 01 t=9E34A0 c=8.6C key=P8 (04000002)

```
Description: Dispatch Method
Class Name: JPolicyEmSQLMO
Method Name: _get_policyNo
object: 260ED1F8
objectPtr refcount: 3
objectPtr classname: JPolicyEmSQLMO
```

Trace: 2004/10/12 00:14:57.272 01 t=9E34A0 c=8.6C key=P8 (13007002)

```
FunctionName: policyNo
SourceId: _JPolicyB0Impl
Category: AUDIT
ExtendedMessage: BBOJ0002 Policy number 33,333 obtained.
```

To navigate through the trace data in the Dump Display Reporter panel, use the commands and PF keys listed in *z/OS MVS IPCS User's Guide*, SA22-7596.

For information about viewing CTRACE and JRas data through IPCS, refer to the *WebSphere Application Server for z/OS V6, Troubleshooting and support*, GA22-7964-03.

Also, you can visit the *WebSphere for z/OS Information Center* at:
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

20.2 JDBC trace

Tracing the JDBC is useful when you are experiencing problems with applications that use a WebSphere for z/OS data source to connect to databases that support traces.

There are two traces that you can use. The first is based on the Java Logging API for WebSphere for z/OS, and it is a JVM trace. The other is specifically for DB2.

20.2.1 Setting up the JDBC trace

You use the WebSphere for z/OS administrative console to turn on the JVM trace. Turn on JDBC tracing by using the trace strings that are shown in Table 20-1. For information about how to enable trace strings to look for error messages, see 19.4, “The Java Logging API” on page 227.

Table 20-1 JDBC trace strings

Trace string	Description
com.ibm.ws.database.logwriter	Trace string for databases that use the GenericDataStoreHelper that can also be used for unsupported databases
com.ibm.ws.db2.logwriter	Trace string for DB2 databases
com.ibm.ws.oracle.logwriter	Trace string for Oracle databases
com.ibm.ws.cloudscape.logwriter	Trace string for Cloudscape™ databases
com.ibm.ws.informix.logwriter	Trace string for Informix® databases
com.ibm.ws.sqlserver.logwriter	Trace string for Microsoft SQL Server databases
com.ibm.ws.sybase.logwriter	Trace string for Sybase databases

If this trace cannot help you and you are connecting to DB2 databases, the DB2 SQLJ/JDBC trace might. The following steps describe the procedure for obtaining this JDBC DB2 trace:

1. Log in to the Administrative Console, expand the Environment item in the menu, and select **WebSphere Variables**. Select the scope for which you want to enable the trace and click **Apply**.
2. Click **New**. Add this variable name and its value:
DB2SQLJPROPERTIES=/mydb2dir/wscdb_db2sqljjdbc.properties
3. In the properties file called /mydb2dir/wscdb_db2sqljjdbc.properties, set up the variable DB2SQLJ_TRACE_FILENAME to enable the SQLJ/JDBC trace and specify the name of the file to which the trace is written:
DB2SQLJ_TRACE_FILENAME=/tmp/IVP2_jdbctrace
4. The JDBC trace produces two HFS files:
 - /tmp/IVP2_jdbctrace: This file is in binary format. You must format it using the db2sqljtrace command (as shown in the following step).
 - /tmp/IVP2_jdbctrace.JTRACE: This file contains readable text.

5. To format the binary trace data, use the following db2sqljtrace command in the USS environment:

```
db2sqljtrace fmt|flw TRACE_FILENAME > OUTPUT_FILENAME
```

The **fmt|flw** sub commands ensure that the output trace contains:

fmt A record every time a function is entered or exited before a failure

flw The function flow before a failure

OUTPUT_FILENAME The name of the file that the new formatted trace is written to.

To run **db2sqljtrace** correctly, ensure that the JDBC path and libraries variables are PATH and LIBPATH. You can change them with the following commands:

```
export PATH=$PATH:/usr/lpp/db2/db2810/bin
export LIBPATH=$LIBPATH:/usr/lpp/db2/db2810/lib
```

Note: The IBM default path is /usr/lpp/db2/db2810/, but you might have another path, depending on your installation.

You can verify that they are correct with the following commands:

```
echo $PATH
echo $LIBPATH
```

For more information, refer to DB2 documentation.

20.2.2 JDBC trace output and interpretation

JDBC trace information shows Java methods, database names, plan names, user names, or connection pools. The file name in this example (Example 20-2) is /tmp/IVP2_jdbctrace.JTRACE.

Example 20-2 JDBC trace sample

```
Timestamp> <Trace Point> <Method Name> <Class/ObjectId> <Thread Name> <Optional Parms>
<2004.10.04 20:15:02.810> <Entry> <printHeader> <COM.ibm.db2os390.sqlj.util.DB2SQLJTrace>
<P=253767:0:0:CT>
-- <p#1=Start of DB2 SQLJ/JDBC Tracing <2004.10.04 20:15:02.810>>
-- <p#2=DB2 for OS/390 SQLJ/JDBC Driver build version is: DB2 7.1 UQ85384 JDBC 2.0>
<2004.10.04 20:15:02.949> <Entry> <Constructor> <COM.ibm.db2os390.sqlj.jdbc.DB2SQLJConnection@5254a29a>
<P=253767:0:0:CT>
-- <p#1=source=DSN7>
-- <p#2=parser=COM.ibm.db2os390.sqlj.jdbc.parser.DB2JDBCParser@5c10229d>
-- <p#3=planname=DSNJDBC>
-- <p#4=pooledConnection=com.ibm.db2.jcc.DB2PooledConnection@6c54e29a>
<2004.10.04 20:15:02.9
<92> <Entry> <setTransactionIsolation> <COM.ibm.db2os390.sqlj.jdbc.DB2SQLJConnection@5254a29a>
<P=253767:0:0:CT>
-- <p#1=Current Isolation=2>
-- <p#2=New Isolation=2>
-- <p#3=COM.ibm.db2os390.sqlj.jdbc.DB2SQLJConnection@5254a29a[pCONN=3767fec8]> <2004.10.04
20:15:03.123> <Exit> <setTransactionIsolation> <COM.ibm.db2os390.sqlj.jdbc.DB2SQLJConnection@5254a29a>
<P=253767:0:0:CT>
-- <p#1=COM.ibm.db2os390.sqlj.jdbc.DB2SQLJConnection@5254a29a[pCONN=3767fec8]>
<2004.10.04 20:15:03.139> <Exit> <Constructor> <COM.ibm.db2os390.sqlj.jdbc.DB2SQLJConnection@5254a29a>
<P=253767:0:0:CT>
-- <p#1=COM.ibm.db2os390.sqlj.jdbc.DB2SQLJConnection@5254a29a[pCONN=3767fec8]>
<2004.10.04 20:15:03.157> <Entry> <getConnection> <com.ibm.db2.jcc.DB2PooledConnection@6c54e29a>
<P=253767:0:0:CT>
-- <p#1=com.ibm.db2.jcc.DB2PooledConnection@6c54e29a>
```

```

<2004.10.04 20:15:03.164> <Entry>    <constructor> <com.ibm.db2.jcc.DB2LogicalConnection>
<P=253767:0=0:CT>
<2004.10.04 20:15:03.164> <Exit>     <constructor> <com.ibm.db2.jcc.DB2LogicalConnection>
<P=253767:0=0:CT>
-- <p#1=com.ibm.db2.jcc.DB2LogicalConnection@4ba6629c[mClosed=false;mConnection=5254a29a]>

```

Example 20-3 shows a JDBC trace that was formatted with the fmt sub command.

Example 20-3 JDBC trace formatted with fmt sub command

```

Trace Version      : DB2 7.1
Driver Build Version : DB2 7.1 UQ85384 JDBC 2.0
Trace Captured at   : Mon Oct  4 20:15:02 2004
Trace buffer size    : 262144 bytes
Records to keep      : LAST
Trace truncated      : NO
Trace wrapped        : NO
Shared Memory Address : 0x1E5CA568
First empty slot     : 7604
Trace Table Address  : 0x1E681030
Size of trace        : 7592 bytes
Records in trace     : 134
1  SQLJ fnc_entry    sqlj_JDBC_Driver DB2SQLJ_sqlj_driver_native_init (2.1.7.1)
   pid 0x007fb620; tid 0x007fb620; time 1096935302; tpoint 0
   0000 0000          ....
2  SQLJ fnc_entry    sqlj_JDBC_AttachMgr sqlj_Attach_Global_Init (2.1.14.1)
   pid 0x007fb620; tid 0x007fb620; time 1096935302; tpoint 0
   0000 0000          ....
3  SQLJ fnc_data     sqlj_JDBC_AttachMgr sqlj_Attach_Global_Init (2.3.14.1)
   pid 0x007fb620; tid 0x007fb620; time 1096935302; tpoint 1
   0000 0001 0000 0004 37ac 75d0          .....}
4  SQLJ fnc_entry    sqlj_Native_Util sqlj_memAlloc (2.1.3.1)
   pid 0x007fb620; tid 0x007fb620; time 1096935302; tpoint 0
   0000 0000          ....

```

Example 20-4 shows a JDBC trace formatted with the flw sub command.

Example 20-4 JDBC trace formatted with flw sub command

```

Trace Version      : DB2 7.1
Driver Build Version : DB2 7.1 PQ56655
Trace Captured at   : Wed Sep 18 17:12:00 2002
Trace buffer size    : 262144 bytes
Records to keep      : LAST
Trace truncated      : NO
Trace wrapped        : NO
Shared Memory Address : 0x236D3568
First empty slot     : 184452
Trace Table Address  : 0x2377C030
Size of trace        : 184440 bytes
Records in trace     : 2298
pid = 0x007f9358;
1  DB2SQLJ_sqlj_driver_native_init fnc_entry ...
2  |sqlj_Attach_Global_Init fnc_entry ...
3  |sqlj_Attach_Global_Init fnc_data ...
4  | |sqlj_memAlloc fnc_entry ...
5  | |sqlj_memAlloc fnc_data ...
6  | |sqlj_memAlloc fnc_retcode 0

```

20.3 SVC dumps

SVC dumps are useful in diagnosing many WebSphere for z/OS problems, including:

- ▶ An abend issued by the application server
- ▶ An abend in the operating system or subsystem component
- ▶ An application server timeout
- ▶ An application server hang
- ▶ An application server that is using high CPU

An SVC dump is generally initiated by the z/OS operating system when a programming exception occurs. An SVC dump stores data in dump data sets that you preallocate or that the system allocates automatically as needed.

An SVC dump is an unformatted dump and is not readable without the use of a formatting tool. The z/OS dump formatting tool is the IPCS.

20.3.1 Capturing an SVC dump

There are three ways to capture an SVC dump:

- ▶ The system initiates the dump because an exception or problem occurred.
- ▶ You initiate the dump from the MVS console to gather diagnostic data using the MVS dump command. SVC dumps that are initiated this way are called *console dumps*.
- ▶ You set SLIP to trigger an SVCDUMP when a particular condition is met.

Console dump

A console dump is an SVC dump that is captured using the MVS DUMP command and run from the console or SDSF log. It is referred to as a console dump because of how it is triggered. You initiate a console dump when:

- ▶ You want an SVC dump of a servant region or a dump of the servant controller region.
- ▶ You suspect a particular servant region to be the source of a problem. Dump the controller region and all of its servant regions.
- ▶ You detect a hang or high CPU utilization for a particular address space.

A sample PARMLIB member that determines the information to be included in a console dump can be found in SBBOSLIB(BBODMCCB). The sample contains instructions about its installation and use.

The standard SDATA expected in an SVC dump is:

```
SDATA=(ALLNUC,CSA,GRSQ,LPA,LSQA,PSA,RGN,SQA,SUM,SWA,TRT)
```

SLIP dump

A SLIP dump is an SVC dump, but it is called a SLIP dump because it is triggered by the MVS SLIP command. You can use a SLIP dump when there is an error and no SVC dump is being produced because the SVC dump is probably being suppressed by the Dump Analysis and Elimination facility. You can also use a SLIP dump when you want a dump to be triggered when a certain error message occurs or when IBM support has asked you for one

An example of s SLIP dump that we used to capture an EC3/04130007 is shown in Example 20-5.

Example 20-5 Slip dump for capturing EC3/04130007

```
SLIP SET,A=SVCD,COMP=EC3,REASON=04130007,ID=WEC3,
MATCHLIM=20,ASIDLST=(O,H,I,P,S),
SDATA=(ALLNUC,CSA,GRSQ,LPA,LSQA,PSA,RGN,SQA,SUM,SWA,TRT),END
```

20.3.2 Problems capturing an SVC dump

If you cannot find an SVC dump for a specific abend, your installation might be using Dump Analysis and Elimination to suppress the dump. If this is the case, you can change Dump Analysis and Elimination to let the dump be taken, or you set a SLIP on the specific abend for a particular job name if the timeout is happening consistently.

If SYSLOG has a message that indicates that the maximum space limit was reached for this dump, the SVC dump might be a partial one. The partial SVC dump might not contain the data that you need to diagnose the timeout. This limit means that the data set that is used for the SVC dump is not large enough, and you must change the size to capture a complete dump. For example:

```
IEA043I SVC DUMP REACHED MAXSPACE LIMIT - MAXSPACE=00000500MEG
```

To change the size of dump storage, issue:

```
CHNGDUMP SET,SDUMP,MAXSPACE=nnnnnM
```

20.3.3 Formatting an SVC dump using IPCS

IPCS is used to format an SVC dump. The formatting commands that you use depend on the problem being investigated. We have listed some IPCS commands that we have found to be especially useful for formatting an SVC dump in Table 20-2.

Table 20-2 Useful IPCS commands for formatting an SVC dump

IPCS commands	Explanation
ip st regs faildata worksheet	This command formats the dump title, date, and time and shows the default ASID) for the dump, registers, and psw information.
ip systrace time(local)	This command formats the System Trace entries for the TCBs in the ASID.
ip select list all	This generates a list of all the address spaces in the system at the time of the dump with ASID and JOBNAME.
ip summary format	This command formats all the address space and task-related control blocks for a particular ASID. Some of the formatted control blocks are the ASCB, TCBs, RBs, and RTM2WA.
ip verbx ledata 'nthreads(*)'	This command formats all the C-stacks (DSAs) for threads in the process for the default ASID.
ip verbx ledata 'nthreads(*) asid(aaaa)'	This command formats all the C-stacks (DSAs) for threads in the process for the ASID specified.
ip verbx ledata 'tcb(tttttt) nthreads(*)'	This command formats all the C-stacks (DSAs) for a particular TCB and includes traceback information.

IPCS commands	Explanation
ip omvsdata process detail asid(x'hhhh')	This command generates a report for the process that shows the thread status from a USS perspective.
ip analyze resource	This command generates a report showing resource contention. This is useful in a hang situation.
ip verbx vsmdata 'summary'	This command generates a report that shows the virtual storage usage for the system. This is useful when the system is experiencing storage problems.
ip verbx logdata	This command formats the available erep detail reports.
ip verbx mtrace	This command formats the available master trace, which holds syslog information.

20.3.4 Related references

For additional information about Dump Analysis and Elimination, see *z/OS V1R6.0 MVS Diagnosis: Tools and Service Aids*, GA22-7589.

Refer to *z/OS V1R6.0 MVS System Commands*, SA22-7627 for details SLIP and DUMP.

For information about how to use IPCS refer to *z/OS V1R6.0 MVS IPCS Commands SA22-7594-05* and *z/OS V1R2.0 MVS IPCS User's Guide*, SA22-7596-01.

20.4 CEEDUMP

Generally, a CEEDUMP is generated if a region fails or there is an abend (for example, an error occurring in the z/OS Language Environment® or Java Runtime Environment). Typically, CEEDUMP can be found in the job logs of the different servers. CEEDUMP can help you identify the failing module in the *Traceback* section of the dump.

Search for Traceback at the top of the CEEDUMP. The result is a sequence of modules as shown in Figure 20-1.

CEE3DMP V1 R3.0: Condition processing resulted in the unhandled condition.09/18/02 5:19:06 PM

Page: 1

Information for enclave main

Information for thread 23B00F1000000000

Traceback:

DSA Addr	Program Unit	PU Addr	PU Offset	Entry	E Addr	E Offset	Statement	Load Mod	Service	Status
236D4768		06CB6B48	+00000806	__zerros	06CB6B48	+00000806		CEEV003		Call
236D3C08	CEEHDSP	06E7C2B0	+00002BE6	CEEHDSP	06E7C2B0	+00002BE6		CEEPLPKA		Call
		/src/share/java/runtime/jni.c								
236D37B8		26091830	+00000528	JNI_CreateJavaVM	26091830	+00000528	4432 *PATHNAM	Exception		
236D32C0		1C2FBEE0	+00001270	loadAndInitVM(JavaVM_**,JNIEnv_**,SOMEException*)						
					1C2FBEE0	+00001270	411 BBOLRT	CB30038	Call	
236D3210		1C301E88	+000002BE	getJavaEnv(SOMEException*)						
					1C301E88	+000002BE	1679 BBOLRT	CB30036	Call	
236D30F8		1C302860	+00000092	buildJavaClass(const char*,SOMEException*)						
					1C302860	+00000092	1921 BBOLRT	CB30036	Call	
236D3030		1C30A5F0	+000001A4	__cdecl _NewObject(SOMClassRef*,SOMEException*)						

Figure 20-1 CEEDUMP sample

The last modules in action are at the top, and underneath them are the oldest, in order. Look for the term *Exception* in the Status column. Usually, an exception is in one of the last modules in action, so it is likely to be near the top of the Status column.

The name of the entry with the exception (*JNI_CreateJavaVM* in the Entry column) is the most important string in this CEEDUMP, because it is the search argument that you use for researching known problems and their solutions in APARs, PMRs, or on your favorite problem search site on the Internet.

Note: See Chapter 1, “Problem determination methodology” on page 3, for information about IBM resources and using the exception entry to search problem databases.

A CEEDUMP is a formatted dump and therefore IPCS is not required to read it. Depending on the problem, a CEEDUMP might not have enough formatted information and IBM support might require an SVC dump. For CEEDUMP parameter settings, see *z/OS V1R6.0 Language Environment Debugging Guide*, GA22-7560

If you have an SVC dump, it is possible to view CEEDUMP contents in an SVC dump using the *IPCS verbexit LEDATA* with the CEEDUMP or *NTHREADS* options. This formats the Language Environment control blocks to help in analysis. For additional information, see the *z/OS V1R6.0 Language Environment Debugging Guide*, GA22-7560, to learn more about using IPCS to format and analyze dumps.

20.5 Java Transaction Dump (TDUMP)

When an out of memory error (*OutOfMemoryErrors*) occurs, a Java Transaction Dump (TDUMP) is produced. The TDUMP is generated from the IEATDUMP MVS service by default when there is a program check or exception in the JVM.

You can use IPCS to inspect a TDUMP. You can also inspect a TDUMP using a Java application such as *SVCDUMP* if the dump data set has been transferred in binary mode to the inspecting system. A TDUMP can have multiple address spaces. It is important to work with the correct address space associated with the failing Java process.

In the servant region job log and syslog, you see messages such as IEA822I indicating that a transaction dump has been taken as shown in Example 20-6.

Example 20-6 Transaction dump message in job log

```
IEA822I COMPLETE TRANSACTION DUMP WRITTEN TO ASSR1.JVM.TDUMP.WS6422S.D050809.T190153
```

The job log also shows the JVM messages in the trace part of the job log as shown in Example 20-7.

Example 20-7 Java OutOfMemoryErrors shown in job log

```
JVMDG217: Dump Handler is Processing OutOfMemory - Please Wait.
JVMHP002: JVM requesting System Transaction Dump
JVMHP012: System Transaction Dump written to ASSR1.JVM.TDUMP.WS6422S.D050809.T1
JVMDG315: JVM Requesting Heap dump file
JVMDG318: Heap dump file written to /SC42/tmp/HEAPDUMP.20050809.190559.16843095
JVMDG303: JVM Requesting Java core file
JVMDG304: Java core file written to /SC42/tmp/JAVADUMP.20050809.190613.16843095
JVMDG274: Dump Handler has Processed OutOfMemory.
JVMST109: Insufficient space in Javaheap to satisfy allocation request
```

```
...Trace: 2005/08/09 19:06:19.729 01 t=7CC148 c=11.1 key=P8 (13007002)
ThreadId: 00000029
FunctionName: com.ibm.ws.webcontainer.servlet.ServletWrapper
SourceId: com.ibm.ws.webcontainer.servlet.ServletWrapper
Category: SEVERE
ExtendedMessage: BB000220E: SRVE0068E: Could not invoke the service() method
on servlet MemLeak. Exception thrown : java.lang.OutOf
MemoryError: JVMXE006:OutOfMemoryError, stAllocArray for executeJava failed
```

In the case of an OutOfMemory error, looking for the memory leak using the transaction dump in IPCS is not useful. Java tools such as Heaproots or the Memory dump diagnostic for Java are more effective in this case.

You can disable the generation of a TDUMP, but IBM does not recommend it.

For more information about TDUMP, refer to *IBM Developer Kit and Runtime Environment, Java 2 Technology Edition, Version 1.4.2, Diagnostics Guide*, SC34-6358-01, for your version of Java on z/OS.

20.6 Javadump

A Javadump produces files with diagnostic information that is related to the JVM and a Java application captured at a point while it is run. For example, the information can be about the operating system, the application environment, threads, native stack, locks, and memory. The exact contents depend on the platform that you are running.

By default, a Javadump occurs when the JVM terminates unexpectedly. A Javadump can also be triggered by sending specific signals to the JVM.

Note: Javadump is also known as Javacore. This is *not* the same as a core file (that is, an operating system feature that can be produced by any program, not just the JVM).

For more information about the Javadump, refer to *IBM Developer Kit and Runtime Environment, Java 2 Technology Edition, Version 1.4.2, Diagnostics Guide*, SC34-6358-01, for your version of Java on z/OS.

20.7 Heapdump

Heapdump is an IBM JVM facility that generates a dump of all of the live objects that are on the Java heap, that is, those that are used by the Java application. It shows the objects that are using large amounts of memory on the Java heap and what is preventing them from being collected by the Garbage Collector.

For more information about the Heapdump refer to *IBM Developer Kit and Runtime Environment, Java 2 Technology Edition, Version 1.4.2, Diagnostics Guide*, SC34-6358-01, for your version of Java on z/OS.

Archived

Diagnostic tools for WebSphere for z/OS

In this chapter, we explain the following diagnostic tools for WebSphere for z/OS that can support you in the problem determination process:

- ▶ Collector tool
- ▶ JVM dump and heap analysis tools
- ▶ Memory Dump Diagnostic Tool for Java
- ▶ Trace Analyzer for WebSphere Application Server
- ▶ Java Garbage Collection Formatter
- ▶ dumpNameSpace tool
- ▶ Rational Application Developer V6
- ▶ Tivoli Performance Viewer
- ▶ OMEGAMON® XE

For each tool, we provide information about its nature, discuss when to use it, and explain how to use it. We describe the output, demonstrate how to interpret it, and provide an example.

21.1 Collector tool

WebSphere Application Server, Version 6.0.x on AIX, HP-UX, Linux, Solaris, and Windows provides a collector tool that you can also use with z/OS. Run it for all application servers and the deployment manager. The collector tool gathers extensive information about your WebSphere Application Server environment and packages it in a JAR file that you can send to IBM support to assist in determining and analyzing your problem.

Information in the JAR file includes logs, property files, configuration files, operating system and Java data, and the absence or level of each software prerequisite. The collector program runs to completion despite any errors that it might find. Errors might include missing files or commands. The collector tool collects as much data in the JAR file as possible.

There are two phases of using the collector tool. The first phase runs against your WebSphere for z/OS environment and produces the JAR file. The IBM Support team performs the second phase, which is analyzing the file. For more information about the collector tool and how to run it search for “Gathering information with the Collector tool” at the WebSphere Application Server, Version 6.0.x, Information Center:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

21.2 JVM dump and heap analysis tools

The JVM sits at the core of the WebSphere runtime environment. The many benefits of using Java for development and a JVM as a targeted run time also bring with them a few issues. The diagnosis of production environment problems can seem more difficult than and very remote from the original development environments, where real-time, GUI-based debugging is standard. However, this real-time approach can transfer to System z, zSeries, and production servers, at least from a technical, “it is possible” perspective. For example, you can load a server with a JVM in debug mode and attach a remote debugger such as WebSphere Studio Integrated Edition.

There are three problem analysis tools for the WebSphere for z/OS production environment:

- ▶ Svcdump.jar
- ▶ HeapRoots
- ▶ Dumpviewer GUI and jformat

There are other tools available that make use of the various JVM interfaces, such as JVM Profiling Interface (JVMP) and JVM Monitoring Interface (JVMMI), to provide real-time monitoring and profiling of the JVM that is attached to WebSphere servant processes. These techniques work well when the transaction throughput is low, or when you want to target behavior of a server that is being used in z/OS in development or testing before deploying it to a production server. They are usually unsuitable for production usage, for a variety of reasons:

- ▶ There is a requirement to load a debug version of the JVM. The debug version is built with many asserts for invalid conditions turned on. This means that the performance of the JVM is not adequate for production usage.
- ▶ The quantity of data that is generated by profiling tools from a server with a high transaction throughput is generally much larger than can be accommodated by the system or than can be meaningfully post-processed from the profiling tool.
- ▶ Profiling tools are typically fairly platform agnostic and therefore not sensitive enough to platform peculiarities or problems where the problem needs to be researched through middleware or OS components.

The analysis of heap-related issues, such as OutOfMemoryError and other crashes, hangs, or loops in WebSphere for z/OS address spaces can be similar to that for the level that is possible with the earlier deployment environments, such as CICS and IMS. You use the relatively low impact tools of unformatted dumps and JVM internal information for analysis, which means that you must draw together knowledge of the JVM internals with the less invasive diagnostic approaches (such as SDUMP) that are typically used to diagnose problems in z/OS production environments. These tools can be used to diagnose problems that affect your important production workload, but can only be recreated in these high-transaction environments.

These approaches should be driven initially by the systems programming staff, because they have authority to access the SVC dumps or transaction dumps that are taken during failures or have the authority to request console dumps of hung or looping servers. After the unformatted dumps are available, the post processing and interpretation of the data can be done by either systems programming or by development staff, because the tools are Java-based and therefore not tied to z/OS.

21.2.1 Svcdump.jar

The svcdump.jar file allows direct access to the binary SVC dump or transaction dumps that are created in z/OS without the need for intermediate software such as IPCS. There are three packages that are shipped in svcdump.jar:

- ▶ Dump utility: com.ibm.jvm.svcdump.Dump package

This formats native and Java stacks for threads in dumped processes that include an instantiated JVM. The dump utility includes a function to print out other useful information, such as in core trace buffers maintained by the JVM and the system trace that mimic or extend the information that can be obtained with IPCS.

- ▶ FindRoots utility: com.ibm.jvm.findroots.* package

This provides multiple ways of formatting the object graphs that are present in the Java-managed heap. This is critical for the sometimes difficult tasks of pinning down object leaks and making sense of heap occupancy.

- ▶ Java API

This can be used to write ad hoc utilities. For example, you could write small programs that report on objects in the Java heap that maintain state data about a business application.

Using svcdump.jar

The svcdump.jar file is available from this Web site (requires IBM registration):

<https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=diagjava>

Note: This tool is in active development, so it would be helpful to IBM if you provide feedback about your experiences with it.

You need three files:

- ▶ The svcdump.jar file
- ▶ The doc.jar file, which contains documentation for the exposed API
- ▶ libsvcdump.so, a DLL that allows the Java code to access an unformatted dump in an MVS data set rather than in the HFS (when you use this DLL, you do not have to provide a large HFS data set to copy the dump to, meaning that in z/OS, the original dump can be analyzed instead)

Attention: The authors used Version 20041012 of the code. Later versions might offer additional functions or different output.

Copy the three files in binary format to a suitable location in the HFS. In our example, the files are in /u/dclarke. Use the following command to confirm the version of the utility that you are running:

```
java -cp svcdump.jar com.ibm.jvm.svcdump.Dump -version
```

Example 21-1 shows the results that we obtained after we used the command.

Example 21-1 Determining the dump utility version

```
You are using
jar:file:/C:/Documents%20and%20Settings/Administrator/My%20Documents/SVCDumps/svcdump200410
07.jar!/com/ibm/jvm/svcdump/Dump.class
which was last modified on Tue Oct 12 14:53:29 BST 2004
```

This uses introspection to identify when this code was last modified.

Parameters and options for setting up the dump utility

Table 21-1 is an overview of the parameters and options for the dump utility. The most important parameters for WebSphere for z/OS problem determination are described in *WebSphere for z/OS V5 JVM Dump and Heap Analysis Tools*, REDP-3950.

Table 21-1 Parameters and options for the dump utility

Option	Description
-debug	Print internal debugging information.
-verbose	Print extra information.
-heap	Print a table that shows which classes have the most objects allocated.
-cache	Print alloc cache.
-exception	Print old exception objects.
-dis <addr> <n>	Disassemble <n> instructions starting at <addr> (hex).
-dump <addr> <n>	Dump <n> words of storage starting at <addr> (hex).
-dumpapars	Print the APARs installed.
-dumpclasses	Print information about all classes.
-dumpclass <addr>	Print information about class at given address.
-dumpobject <addr>	Print information about object at given address.
-dumpmdata <addr>	Print information about mdata at given address.
-dumpprops	Print the system properties.
-dumpnative	Dump all the native methods.
-dumpverbosegc	Dump the verbosegc (verbose garbage collection).
-heapstats	Print stats about heap usage.
-tcbsummary	Print a summary of what the TCBs are doing.
-systrace	Print the system trace.

Option	Description
-hptrace	Print the HPI trace.
-caa <addr>	Specify the CAA to use when disassembling.
-r<n>	Include saved register <n> in stack trace.
-args	Print first four function arguments
-verifysubpools	Verify subpools.
-verifyheap	Verify heap.
-printdosed	Print pinned and dosed objects.
-printroots	Print garbage collection roots.
-version	Print the version and exit.
-fullversion	Print the version of the jvm in the dump and exit.
-title	Print title of the dump and exit.
-time	Print time of the dump and exit.
-dis <addr> <n>	Disassemble <n> instructions starting at <addr> (hex).

To use the dump utility with these options, issue:

```
java com.ibm.jvm.svcdump.Dump [options] <filename>
```

Example 21-2 shows a simple shell script that can be used to run the utility.

Example 21-2 Shell script for the dump utility

```
#!/bin/sh
#TZ=EST5EDT
set -x
DUMPNAME=$1
SVCDUMPJARFILE=/u/dclarke/svcdump20041007.jar
SVCDUMPLIBPATH=/u/dclarke

java -Xmx348m -Dsvcdump.libpath=$SVCDUMPLIBPATH
-Xbootclasspath/p:$SVCDUMPJARFILE \
  -Dsvcdump.default.jvm=0 \
  com.ibm.jvm.svcdump.Dump -exception \
  $DUMPNAME \
  >>$DUMPNAME.svcdump.txt

java -Xmx348m -Dsvcdump.libpath=$SVCDUMPLIBPATH
-Xbootclasspath/p:$SVCDUMPJARFILE \
  -Dsvcdump.default.jvm=0 \
  com.ibm.jvm.svcdump.Dump -hptrace \
  $DUMPNAME \
  >>$DUMPNAME.svcdump.txt

java -Xmx348m -Dsvcdump.libpath=$SVCDUMPLIBPATH
-Xbootclasspath/p:$SVCDUMPJARFILE \
  -Dsvcdump.default.jvm=0 \
  com.ibm.jvm.svcdump.Dump -systrace \
  $DUMPNAME \
  >>$DUMPNAME.svcdump.txt
```

The shell script can then be invoked as shown in Example 21-3.

Example 21-3 invoke shell script

```
/u/dclarke:==>svcdump.sh ONTOP.GS031.P10316.C724.JVMDMP
+ DUMPNAME=ONTOP.GS031.P10316.C724.JVMDMP
+ SVCDUMPJARFILE=/u/dclarke/svcdump20041007.jar
+ SVCDUMPLIBPATH=/u/dclarke
+ java -Xmx348m -Dsvcdump.libpath=/u/dclarke -
Xbootclasspath/p:/u/dclarke/svcdump20041007.jar -
Dsvcdump.default.jvm=0 com.ibm.jvm.svcdump.Dump -exception
ONTOP.GS031.P10316.C724.JVMDMP
+ 1>> ONTOP.GS031.P10316.C724.JVMDMP.svcdump.txt
```

Analysis of the dump can take some time, especially for the first execution. The tool stores some heap information in a small .cache file that make subsequent executions faster. Use this simple JCL to run the utility in batch:

```
//STEP1 EXEC PGM=BPXBATCH,REGION=0M,
// PARM='SH /u/dclarke/svcdump.sh ONTOP.GS031.P10316.C724.JVMDMP'
```

What can I learn from com.ibm.jvm.svcdump.Dump?

If your application crashed, you can use the dump utility to analyze the problem by establishing:

- ▶ The thread that the crash occurred on
- ▶ The native stack for the failing thread
- ▶ The Java stack for the failing thread

If you experience an application loop or hang, you can use the utility to identify:

- ▶ The thread under which a loop is occurring
- ▶ The threads that are contending for resources or that are involved in a lockout
- ▶ A thread waiting for some operation that is external to the server

In all of these cases, the reports can help attach the failure to a particular component or subcomponent before you raise an incident with the IBM service team.

Using com.ibm.jvm.findroots.* in z/OS

FindRoots is a cross-platform tool for analyzing memory leaks in Java applications. Although the package is still called FindRoots for historical reasons, it is now divided into a number of smaller tools that do the same thing; this is mainly to separate the extraction of the heap from the subsequent analysis:

1. Run the convert tool to extract the heap dump and create a Portable Heap Dump (a .phd file).
2. Run PrintDomTree to analyze the .phd file before you try other tools. The advantage of this approach is that you can pass the smaller .phd files rather than large SVC or Transaction dumps.

The quintessential memory leak in Java is created by an application or middleware bug that is causing a reference to an object to be retained. Because this object is still reachable through all of its referents, it can render a large portion of an object graph still reachable and, therefore, not eligible for garbage collection.

For example, consider the case where some state data for a transaction is kept as an XML data object. The design is one where the data is only relevant for the life of the transaction, and the intent is that this object should become eligible for garbage collection after the

transaction has ended. There is a bug in this code that is causing some other global object to maintain a reference to the XML data object after the end of the transaction. Although the object itself is small, it contains a reference to non-trivial numbers of objects that is created by XML parsing.

A successful diagnosis of the problem might be as follows:

1. A still reachable XML data object exists after each transaction runs.
2. The remaining reachable data gradually increases over time after each garbage collection cycle.
3. You can observe this from the verbose:gc output (or from the incore verbosegc data).
4. Eventually, the Java heap is exhausted and an OutOfMemoryError is thrown.
5. You obtain a console dump of the server at the time when the heap usage is high.
6. You run the PrintDomTree tool and establish from the reports that there is an unexpectedly high number of these XML data objects.
7. You find the unexpected reference in the reports from the global object.
8. A review of the logic reveals that this reference is not nulled out after the transaction as the design anticipated.

More information about the different FindRoots utilities and the reports they produce is available in *WebSphere for z/OS V5 JVM Dump and Heap Analysis Tools*, REDP-3950.

21.2.2 HeapRoots

The HeapRoots utility is shipped in the HR204.jar file and is derived from the same requirement as that for being able to map the heap object graphs. It was originally developed for the JVM shipped with IBM AIX. It is now possible to use this code seamlessly with binary SVC or transaction dumps, providing a range of additional functions that are available with the FindRoots utility in svcdump.jar.

HeapRoots is available through the alphaWorks Web site at:

<http://www.alphaworks.ibm.com/tech/heaproots>

To use HeapRoots with .phd files, use the following command:

```
java -classpath svcdump.jar;HR204.jar HR.main.Launcher
```

Examples and details about HeapRoots are available in *WebSphere for z/OS V5 JVM Dump and Heap Analysis Tools*, REDP-3950.

21.2.3 Dumpviewer GUI and jformat

The Dumpviewer GUI and jformat are cross-platform utilities that allow you to use graphs to review native and Java stacks. They also provide facilities to help diagnose lock-related and heap-related problems. On platforms other than z/OS, the tools work with platform independent dump files that are extracted from the native dump. In z/OS, the tools work directly with an SVC or transaction dump, using the code from the svcdump.jar file.

The tools are shipped with the IBM JDKs for all platforms. The GUI is described in some detail in *IBM Developer Kit and Runtime Environment, Java 2 Technology Edition, Version 1.3.1, Diagnostics Guide*, SC34-6200, and *IBM Developer Kit and Runtime Environment, Java 2 Technology Edition, Version 1.4.1, Diagnostics Guide*, SC34-6309, which is shipped with the IBM JDKs 1.3.1 and 1.4.1.

You can invoke Dumpviewer with the following command:

```
java -Xmx512m -cp svcdump.jar com.ibm.jvm.dump.format.DvConsole -g
```

When the GUI initializes, select **File** from the menu to locate the dump for initialization.

Restriction: To use this tool with z/OS, you must export the DISPLAY environment variable to a valid X Server display on a Win32® or Linux system.

With the Win32 JDK shipped with WebSphere for Windows, the formatter can be invoked with the jformat command, which is found in C:\Program Files\IBM\Java142\bin\jformat.

The **-g** switch starts the GUI:

```
"C:\Program Files\IBM\Java142\bin\jformat" -J-Xmx512m -g
```

For more information, see *IBM Developer Kit and Runtime Environment, Java 2 Technology Edition, Version 1.3.1, Diagnostics Guide*, SC34-6200, (used with WebSphere V5.0), and *IBM Developer Kit and Runtime Environment, Java 2 Technology Edition, Version 1.4.1, Diagnostics Guide*, SC34-6309, (used with WebSphere V5.1). These manuals and detailed documentation for the garbage collector used by the IBM JVM, can be downloaded from:

<http://www.ibm.com/developerworks/java/jdk/diagnosis/>

21.3 Memory Dump Diagnostic Tool for Java

The Memory Dump Diagnostic for Tool for Java analyzes common memory dump formats (heap dumps) from the JVM that is running the WebSphere Application Server.

In z/OS, when out of memory errors (OutOfMemoryErrors) occur, a Java transaction dump is produced. This dump can be viewed with IPCS, but the traditional diagnostic information that is available is not helpful for finding the OutOfMemoryError. When a Java transaction dump is produced, you see message IEA822I in the servant region job log.

The analysis of memory dumps is targeted toward identifying regions in the Java heap that might be root causes of memory leaks. The tool is capable of analyzing very large memory dumps that are obtained from production environment application servers that have encountered OutOfMemoryErrors.

The tool is available for download from the developerworks Web site under WebSphere downloads at:

http://www.ibm.com/developerworks/websphere/downloads/memory_dump.html

Documentation that comes with the tool has several screen captures of the tool functions.

Two types of analysis mechanisms are available: a single dump analysis and a comparative analysis of two dumps. The tool displays the contents of the memory dump in a graphical format while highlighting regions that are identified as memory leak suspects. The GUI provides browsing capabilities for verifying suspected memory leaking regions and for understanding the data structures that comprise the leaking regions.

The following formats of memory dumps are supported by this tool:

- ▶ IBM Portable Heap Dump (.phd) format
- ▶ IBM text heap dump format
- ▶ HPROF heap dump format
- ▶ SVC Dumps

Figure 21-1 shows an example of the output that the tool produces.

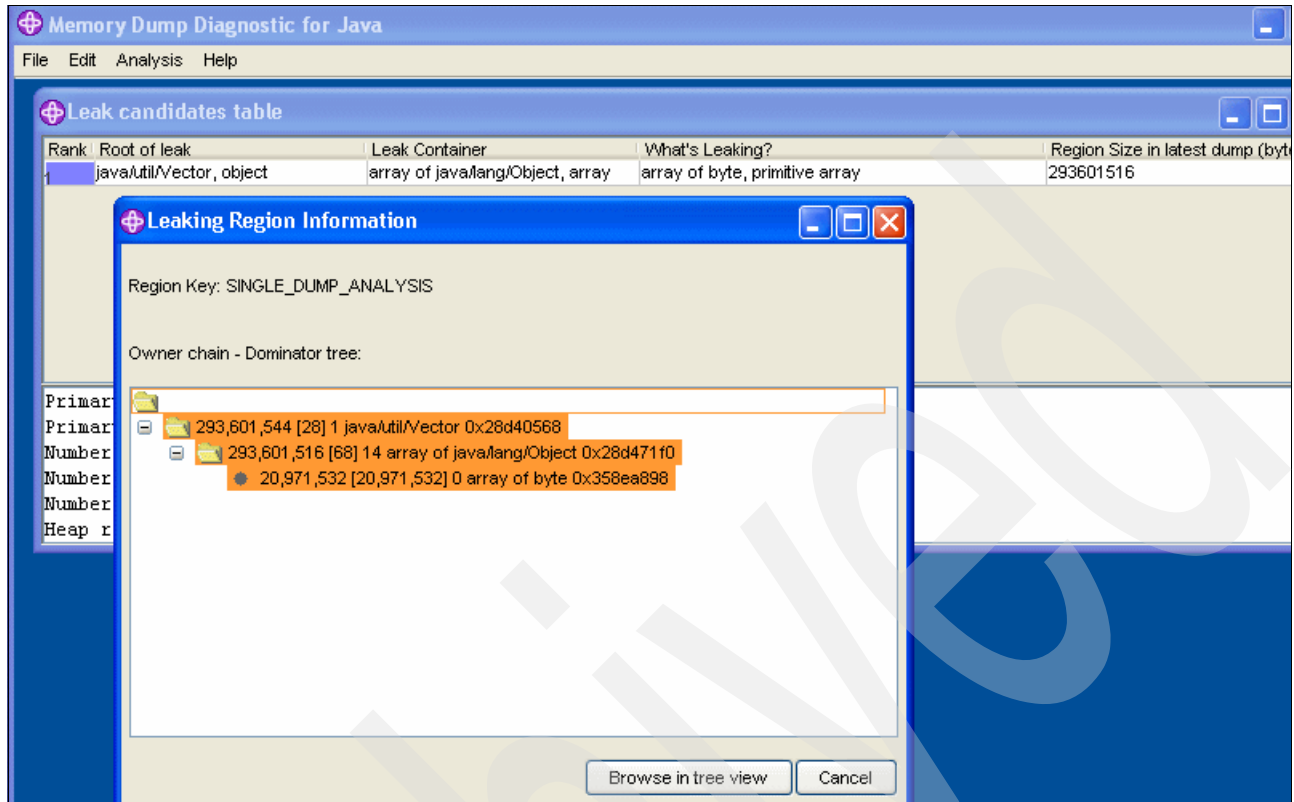


Figure 21-1 Memory Dump Diagnostic tool for Java screens

21.4 Trace Analyzer for WebSphere Application Server

WebSphere for z/OS produces primary diagnostic information in the form of text-based trace logs. The trace logs are used throughout the product to diagnose failures and confirm correct code execution. However, reading a trace log in raw format can be a tedious task. Moreover, WebSphere for z/OS is being deployed in increasingly complex environments and problems are becoming more difficult to diagnose and resolve.

Trace Analyzer for WebSphere Application Server eases the process of reading diagnostic information from the WebSphere for z/OS trace logs by showing sequential, easy to read event oriented traces. Trace Analyzer for WebSphere provides:

- ▶ Visual trace presentation
- ▶ Search and filter capabilities
- ▶ Trace highlighting and mark-up
- ▶ Entry and exit record pairing

Trace Analyzer is written in Java and requires JVM 1.3.1 or higher. To use it, follow this procedure:

1. Download the Trace Analyzer tool from the IBM alphaWorks Web site at:
<http://www.alphaworks.ibm.com/tech/ta4was>
2. Download the JAR file to the selected directory on your workstation.
3. To run the application, issue: **java -jar traceanalyzer.jar traceFile**

traceFile is the name of the trace file that you want to open for analysis. This is an optional feature. You can also open files from the GUI by selecting **File Open**. Figure 21-2 shows a the Trace Analyzer for WebSphere z/OS window.

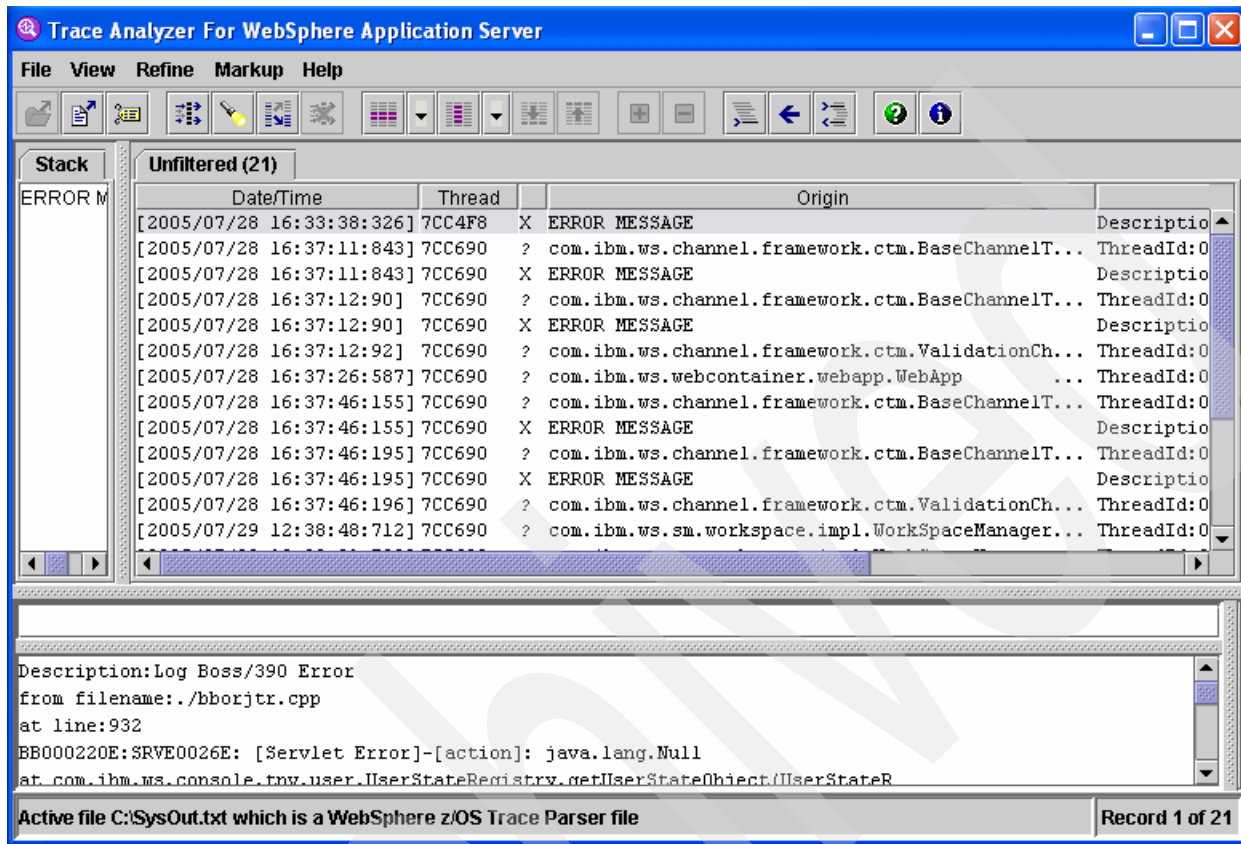


Figure 21-2 Trace Analyzer for WebSphere for z/OS

4. From the Refine menu, you can select **Filter and Search**. By selecting any entry from the trace pane, you can see its full contents in the bottom console.
5. For help using the program, see the integrated help system by selecting **Help** → **Using Trace Analyzer**.

This utility makes it relatively easy to read the diagnostic information, even when you are not very familiar with the component that is being debugged or tested.

21.5 Java Garbage Collection Formatter

WebSphere runs applications in JVM. The JVM heap stores all objects that are created by a running Java application. When the JVM fails to allocate memory because of a Java heap shortage, it starts garbage collection. Garbage collection is the process of automatically freeing objects that are no longer referenced by the Java program. The Formatter tool displays the Java garbage collection statistics in a tabular format. This helps identify whether the JVM heap size is large enough or if there is a memory leak.

There is a Java garbage collection formatter script by John Rankin, called VGC131v7.awk, in the Techdocs TD101216, that you can download from the IBM Web site:

<http://www.ibm.com/support/techdocs>

You first need to capture the Java garbage collection statistics. To use the Administrative Console to turn on verbose garbage collection, follow these steps:

1. Expand the Servers node in the left-hand menu and select **Application Servers**.
2. From the list of servers, select the application server for verbose garbage collection.
3. From the Java and Process Management menu, select **Process Definition**.
4. From the processType list, select the appropriate servant.
5. From the Additional Properties menu, select **Java Virtual Machine**.
6. Find **Verbose garbage collection** in the list of General Properties and select it.

Figure 21-3 shows the Advanced Java Virtual Machine settings for enabling Verbose mode.

Advanced Java virtual machine settings.

Configuration

General Properties

Classpath

Boot Classpath

☐ Verbose class loading

☒ Verbose garbage collection

☐ Verbose JNI

Initial Heap Size

256

Maximum Heap Size

256

☐ Run HProf

HProf Arguments

Additional Properties

■ Custom Properties

Figure 21-3 Enable GC Verbose in Advanced Java virtual machine settings

7. Click **Apply** to apply your changes. Click **Save** to save your configuration.
8. Run transactions through your server for a specific period. The results are a log file that is similar to that in Example 21-4.

Example 21-4 Java garbage collection trace sample

```
<AF[1]: Allocation Failure. need 528 bytes, 122081 ms since last AF>
<AF[1]: managing allocation failure, action=1 (0/255012224) (13421696/13421696)
<GC(1): GC cycle started Mon Aug 1 17:44:10 2005
<GC(1): freed 183448600 bytes, 73% free (196870296/268433920), in 400 ms>
<GC(1): mark: 332 ms, sweep: 68 ms, compact: 0 ms>
<GC(1): refs: soft 0 (age >= 32), weak 275, final 1348, phantom 0>
```

<AF[1]: completed in 403 ms>

The number in the brackets in AF[x] at the start of a line indicates how many times the memory allocation failed. The number in the parentheses in GC(y) indicates how many times garbage collection has occurred since the servant region started:

<GC(1): freed 183448600 bytes, 73% free (196870296/268433920), in 400 ms>

This line indicates how much free memory is available in the JVM after the GC, 73% in our example. If this number decreases over a period of time, there is a problem in the JVM memory heap. Ultimately, the JVM keeps trying to allocate memory and keeps failing because garbage collection cannot recall any free memory. This occurs when all objects in the JVM have held references that cannot be released.

9. Using a REXX or AWK script, format the output of the Java garbage collection trace to get a semicolon-delimited condensed file (Example 21-5).

Example 21-5 Semicolon-delimited Java garbage collection trace

```
afnum ; timeSinceLastAF ; aftime ; afsize ; gcnum ; conGCnumb; timeSinceLastConGC; conGCtime; ConRes ;
ConTarget ; ConTraced ; ConFree ;gcstart ; gcfreed ; freespace ; heapsize ; gctime ;threadStopTime ;
threadStartTime ; marktime ; sweeptime ; compacttime ; mmin ; mmax ; msavg ; moved ; bytes ; reason ;
1;122081;403;528;1;;;;;17:44:10;183448600;196870296;268433920;400;;;332;68;0;;;;;
2;102844;514;528;2;;;;;17:45:53;139549048;152714712;268433920;513;;;462;51;0;;;;;
3;53149;526;5184;3;;;;;17:46:47;134965264;147782608;268433920;526;;;464;62;0;;;;;
4;431101;525;528;4;;;;;17:53:58;140140296;150365624;268433920;524;;;474;50;0;;;;;
5;346417;446;528;5;;;;;17:59:45;156671872;164212856;268433920;446;;;392;54;0;;;;;
6;9495859;525;528;6;;;;;20:38:01;140242632;145579056;268433920;525;;;462;63;0;;;;;
7;12308;539;528;7;;;;;20:38:14;138183336;140835416;268433920;539;;;482;57;0;;;;;
```

10. FTP the output file (in ASCII) to your workstation and import into a spreadsheet tool such as Microsoft Excel® or Lotus® 1-2-3®.
11. From the spreadsheet, create a diagram. Figure 21-4 shows an example.

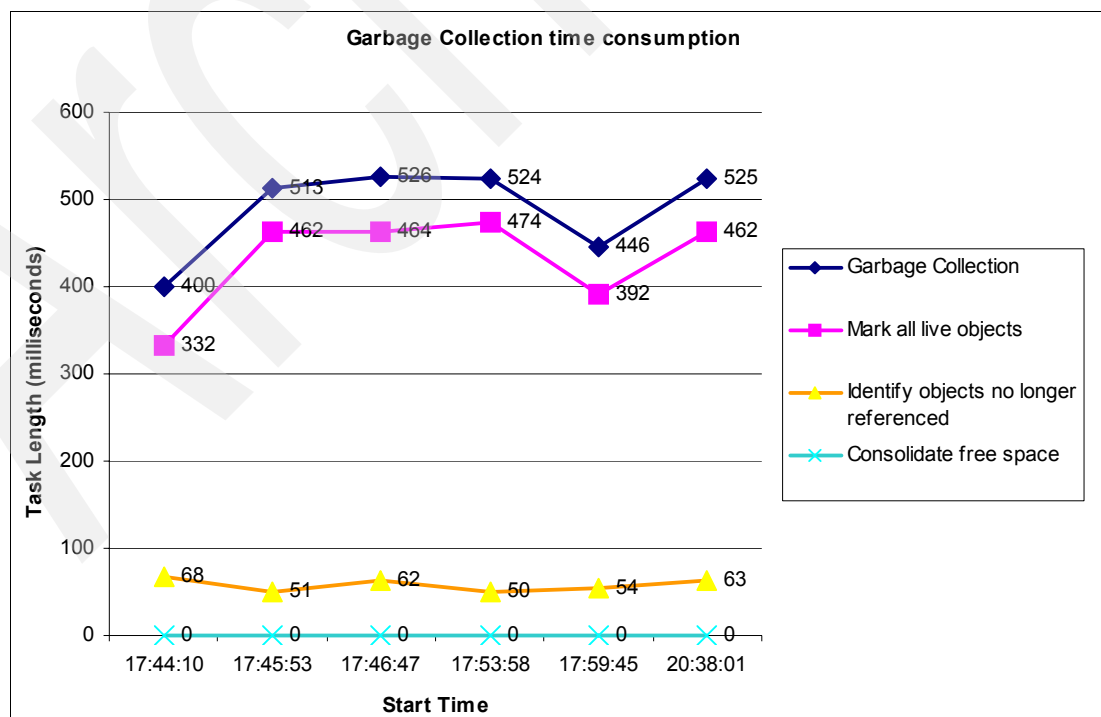


Figure 21-4 Diagram of garbage collection records

For more information, search for “Java memory tuning tip” at the WebSphere for z/OS Information Center:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

21.6 dumpNameSpace tool

Problems can surface when you are accessing resources through the namespace. The namespace is a collection of references to resources, such as connection pools, EJBs, and message listeners. In the WebSphere for z/OS environment, the namespace is federated among all servers in the cell, and each server process has its own name server.

The dumpNameSpace tool obtains the contents of a namespace in a name server. The tool can be invoked with a UNIX shell script or from an interface in the WebSphere Application Server API and you can specify a number of parameters. It can only dump namespaces from remote name servers that are not local to the server process.

To invoke the tool from the OMVS command line, follow these steps:

1. Change the present directory to `<Installation_directory>/<Profiles>/<Profiles_Name>/bin`, where:
 - `<Installation_directory>` is the directory into which WebSphere was installed.
 - `<Profiles>` is the directory under the WebSphere for z/OS installation directory.
 - `<Profiles_Name>` is the name of the profile for the namespace dump.

2. Run this command:

```
dumpNameSpace.sh -port 22809 > nameSpaceDumpFile.txt
```

The port parameter indicates the bootstrap port which, if not specified, defaults to 2809.

3. Log in to the Administrative Console. Select **Servers** → **Application servers** → **server** → **Ports**. You can see the bootstrap port used in your WebSphere for z/OS.
4. Set the final parameter to output to the appropriate file and directory.

For more detailed information about all available parameters and their uses, search for “dumpNameSpace tool” at the WebSphere for z/OS Information Center:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

The dumpNameSpace tool generates a list of all resources and their types. This can be useful for diagnosing problems when resources are referenced and not found in an application. This information can also help resolve ClassCastException messages.

Example 21-6 shows a sample of output from the dumpNameSpace tool. The resource is displayed in the left column, the type in the right column.

Example 21-6 dumpNameSpace output

```
Getting the initial context
Getting the starting context
=====
Name Space Dump
Provider URL: corbaloc:iiop:localhost:22809
Context factory: com.ibm.websphere.naming.WsnInitialContextFactory
Requested root context: cell
Starting context: (top)=cl6552
Formatting rules: jndi
Time of dump: Mon Aug 01 12:09:10 EDT 2005
=====
```

```

=====
Beginning of Name Space Dump
=====
1 (top)
2 (top)/legacyRoot                                javax.naming.Context
2   Linked to context: c16552/persistent
3 (top)/domain                                    javax.naming.Context
3   Linked to context: c16552
4 (top)/persistent                                javax.naming.Context
5 (top)/persistent/cell                            javax.naming.Context
5   Linked to context: c16552
6 (top)/cellname                                  java.lang.String
7 (top)/cell                                       javax.naming.Context
7   Linked to context: c16552
8 (top)/nodes                                    javax.naming.Context
9 (top)/nodes/nd6552                              javax.naming.Context
10 (top)/nodes/nd6552/domain                      javax.naming.Context
10   Linked to context: c16552
11 (top)/nodes/nd6552/servers                     javax.naming.Context
12 (top)/nodes/nd6552/servers/ws6552              javax.naming.Context
=====
End of Name Space Dump
=====

```

21.7 Rational Application Developer V6 Debug Perspective

Rational Application Developer V6 is a very comprehensive development environment with several perspectives. The Debug Perspective is a very useful tool for looking for problems in application code.

The Debug Perspective is used for testing and debugging Java applications, XSL transforms, and other components that are developed within Rational Application Developer. You can debug applications locally on a test server or remotely on a server such as WebSphere for z/OS.

21.7.1 When to use the Rational Application Developer Debug Perspective

With the debugger, you can control how your program runs by setting breakpoints, suspending launches, stepping through your code, and examining the contents of variables.

Breakpoints are temporary markers that you put in your program to tell the debugger to stop your program at a given point. When the workbench is running a program and encounters a breakpoint, it suspends execution. The corresponding thread is suspended (that is, temporarily stops running) so that you can see the stack for the thread.

The suspension occurs at the breakpoint before the statement is executed. You can check the contents of variables and the stack. You can then step over statements, step into other methods or classes, continue running until the next breakpoint is reached, or continue running until you reach the end of the program.

Application developers can use the Rational Application Developer remote debugger to ensure that their application is working as designed on the WebSphere for z/OS platform. It is used to identify any bugs specific to WebSphere for z/OS that cannot be tested during function testing in the WebSphere test environment on Rational Application Developer.

The local system runs the debugger, and the remote system runs both the debugging engine and your program. The person debugging the program on the workstation interacts with the program as usual (except where breakpoints or step commands introduce delays) and can control the program and observe the internal behavior of the remote program from the local system.

21.7.2 Setting up the Rational Application Developer Debug Perspective

To use the Rational Application Developer Debug Perspective, you must first set up the tool and then interact with the tool while it is connected to the running application.

Configuring Rational Application Developer for remote debugging requires changes in the WebSphere for z/OS environment and a connection to the remote application server from Rational Application Developer. Follow these steps to enable remote debugging:

1. Log in to the administrative console for your WebSphere for z/OS.
2. Expand **Servers** in the menu. Select **Application Servers** and the server for debugging.
3. Under Additional Properties, select **Debugging Service**. Figure 21-5 on page 267 shows a sample of the Debugging Service page.

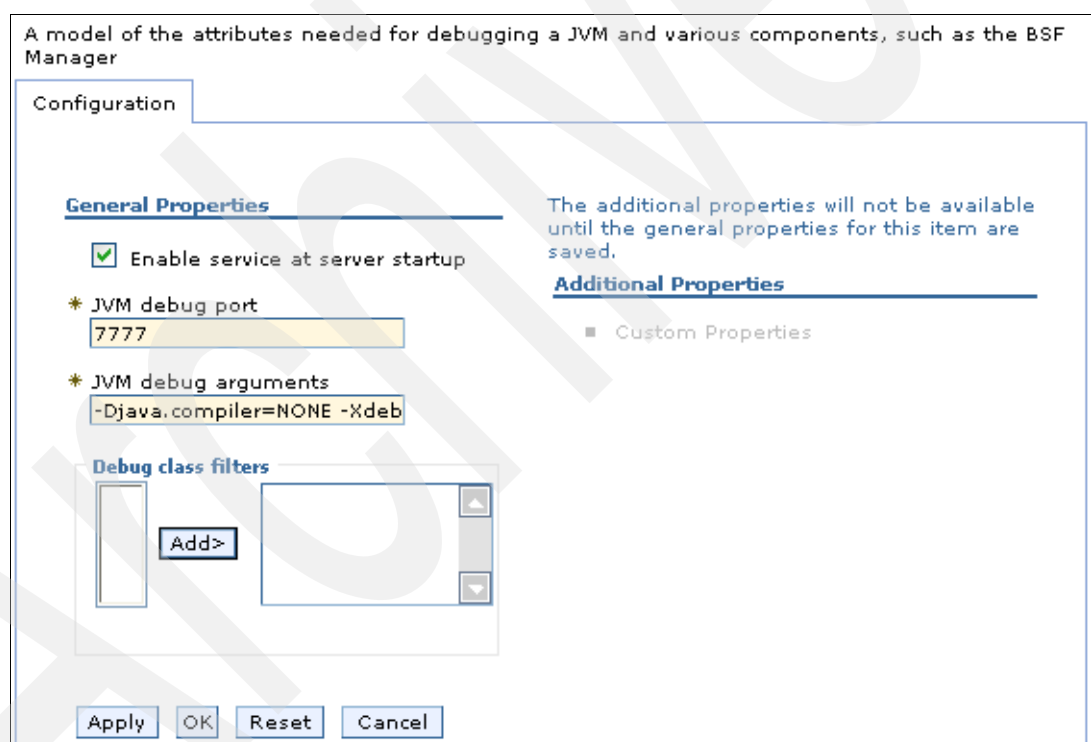


Figure 21-5 Enable Debugging Service in Administrative Console

4. Specify the **JVM debug port**. Port 7777 is the default.
5. Verify the **JVM debug arguments**. The default settings are:
 - Djava.compiler=NONE
 - Xdebug
 - Xnoagent
 - Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=7777

Debug filters do not need to be set initially. You might find them useful as you gain experience with the debugger.

6. Click **Apply** and **Save** to apply your modifications and save the configuration.
7. Stop and start the server for debugging.
8. In Rational Application Developer, open the Debug Perspective by selecting **Window** → **Open Perspective** → **Debug**.
9. Open the Debug configurations by clicking the debug icon and choosing **Debug** from the menu (Figure 21-6).

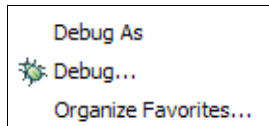


Figure 21-6 Debug menu from the Debug icon

10. Select **Remote Java Application** from the list of configurations. Click **New** to create a new configuration. You should see a panel similar to the one in Figure 21-7.

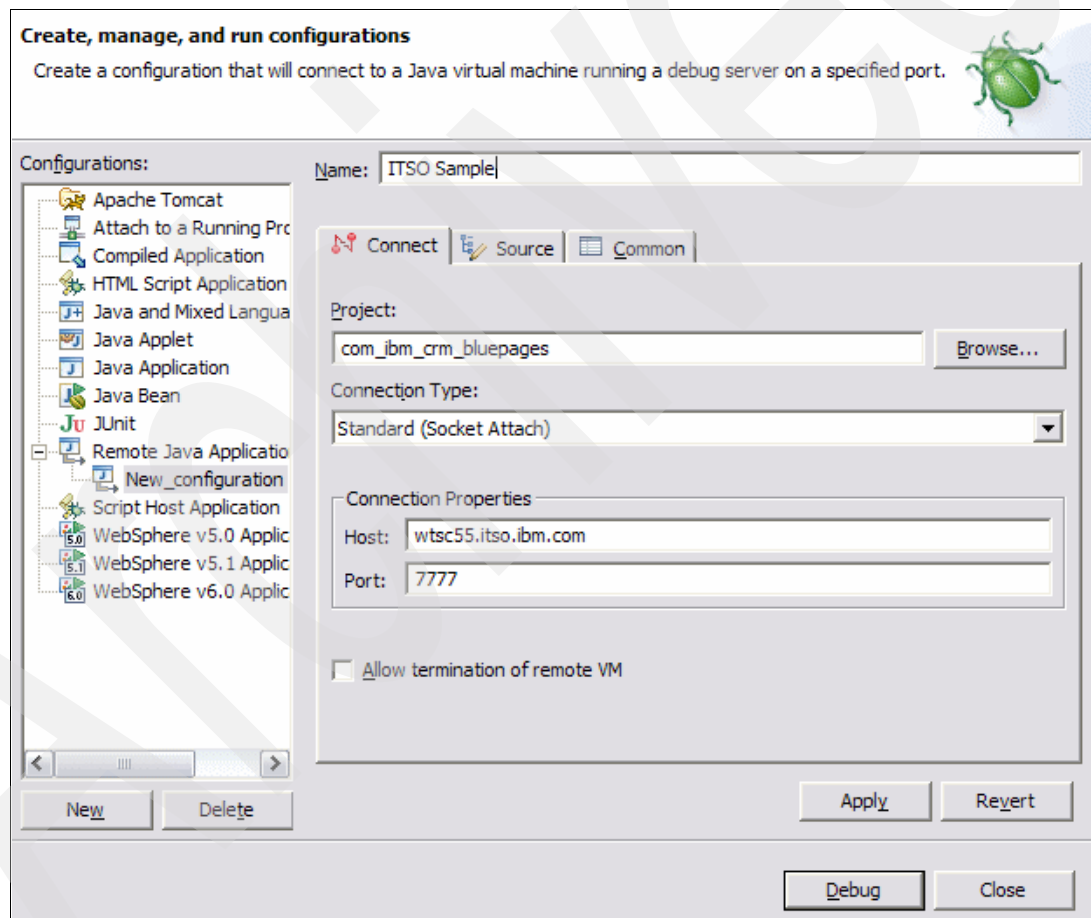


Figure 21-7 Remote Java Application Debug configuration

11. Enter a Name for your configuration. Select the project you want to debug by clicking **Browse** and choosing it from the list of projects in the work space.
12. Enter the host address of your WebSphere for z/OS application server and enter the JVM debug port that you specified in the administrative console.
13. Click **Apply** to save your changes.

14. Click **Debug**. Now, you can start debugging the application on the remote application server. A debug engine daemon is listening for a connection.

To debug Java source code, you set breakpoints in the source code. You can set a breakpoint on a line of code to be triggered when a certain exception occurs. Add breakpoints in your code by double-clicking the gray area next to the line of code that you want to break. Right-click the breakpoint and select **Breakpoint Properties** to specify more detailed properties.

To start a program in debug mode, click the Debug icon, and the Debug Perspective opens. If you have multiple debug configurations, you can choose which to debug by clicking the arrow next to the Debug icon and selecting it from the menu.

In the Debug Perspective, you use icons to step into a line of code, to step over a line of code, or to run to the end of a method (step return). There are multiple views to aid you in debugging your application, including the Variables, Inspector, Debug, and Outline views.

21.7.3 Rational Application Developer debugger output and interpretation

The debugger provides information interactively. As the application runs, you can view data as it is loaded, as it is manipulated, and as it flows through the application.

The Rational Application Developer Debug Perspective features different views that provide different information. Figure 21-8 shows some of these views. You can access views by selecting **Window** → **Show View**. Change between the currently open views by selecting a tab from the bottom of each area in the Rational Application Developer window.

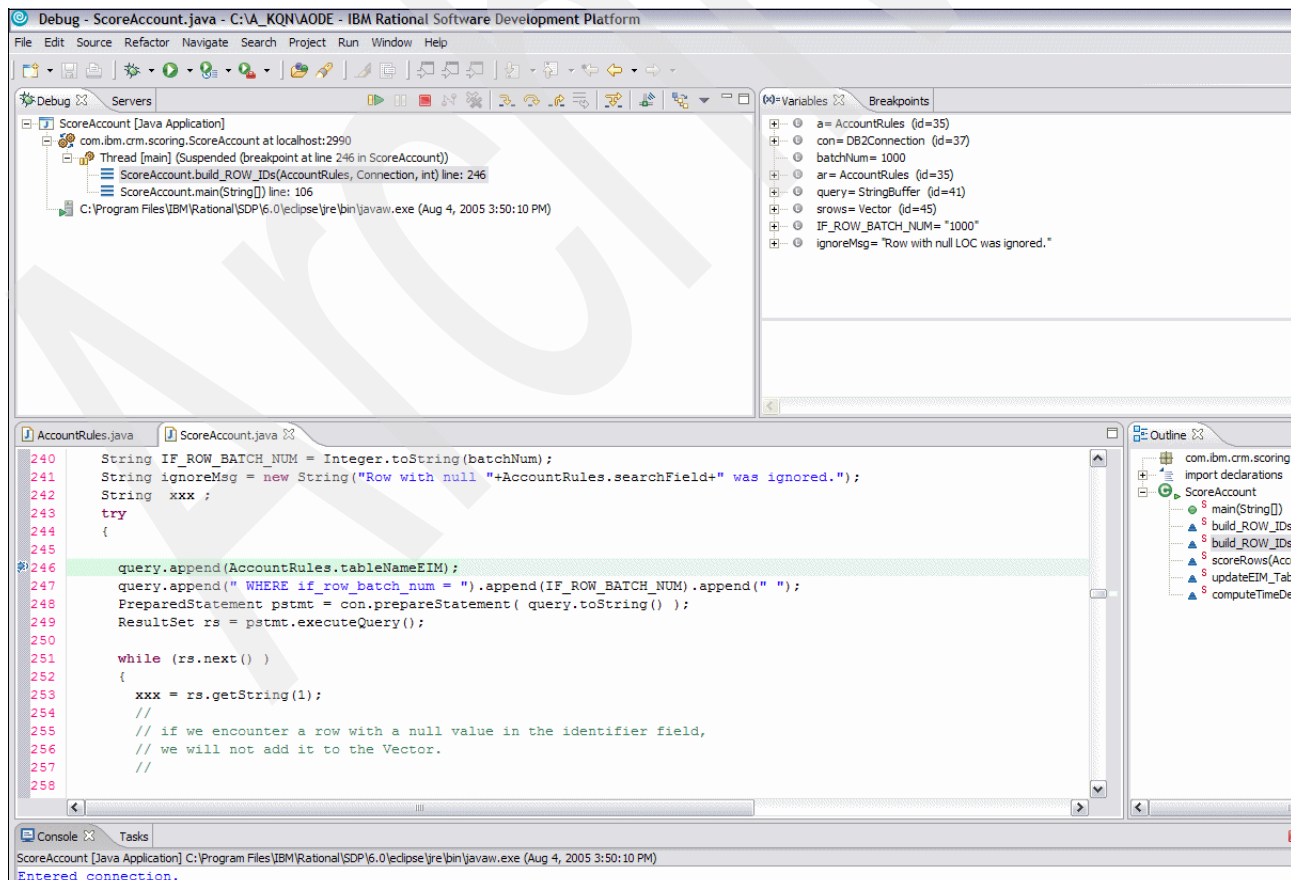


Figure 21-8 The Debug Perspective in Rational Application Developer

Another Rational Application Developer tool is the XML editor and perspective that can be used to properly create and maintain valid XML files. An XML validator ensures that the XML is in a valid format and can be useful in fixing files that have become incorrectly formatted.

The Rational Application Developer can import application client JAR, EJB JAR, EAR, and WAR files, which can be helpful in problem determination. Sometimes, it is important to be able to view the code of the application that you are deploying. These tools can take the place of a decompiler because they can import an archive file into an editable project.

21.8 Tivoli Performance Viewer

Tivoli Performance Viewer is a tool that is integrated in the Administrative Console. It is used to look for bottlenecks in WebSphere for z/OS configuration.

Tivoli Performance Viewer can be used to fine-tune the performance of an enterprise system by optimizing resources. You can view the current performance activity of a server using the Tivoli Performance Viewer.

With Tivoli Performance Viewer, administrators and programmers can monitor the current status of WebSphere Application Server. Because the collection and viewing of data occurs in the application server, performance is affected. To minimize performance impacts, monitor only those servers with activity that needs investigation.

21.8.1 Setting up Tivoli Performance Viewer

To use Tivoli Performance Viewer with WebSphere for z/OS V6, you must set it up by selecting **Monitoring and Tuning** → **Performance Viewer** → **Current Activity** → **server** in the Administrative Console and:

1. Click the server name to view the current activity for that specific server.
2. Select one or more servers from the list.
3. Click **Start Monitoring** to start the Tivoli Performance Monitor for the selected servers.

Figure 21-9 shows the panel for selecting servers for monitoring.

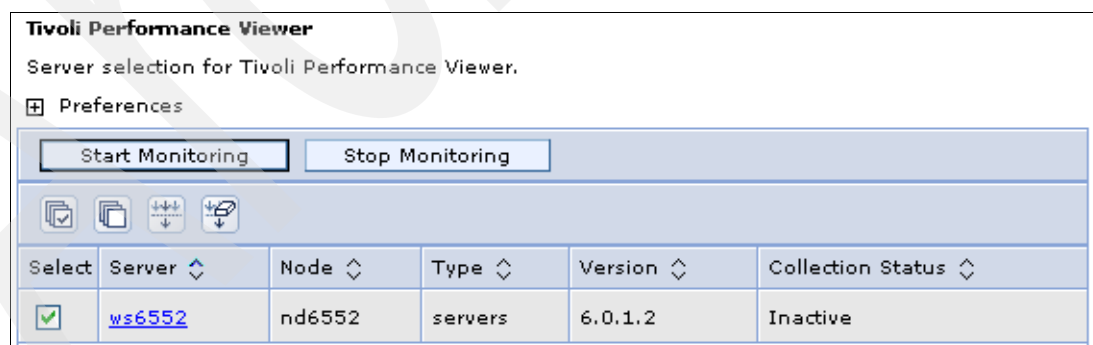


Figure 21-9 Start Tivoli Performance Viewer in Administrative Console

Tivoli Performance Viewer consists of two panels: the Resource Selection panel and the Data Monitoring panel (see Figure 21-10 on page 271). The Resource Selection panel provides a view of resources for which performance data can be displayed. The Data Monitoring panel displays numeric and statistical data for the resources in the Resource Selection panel.

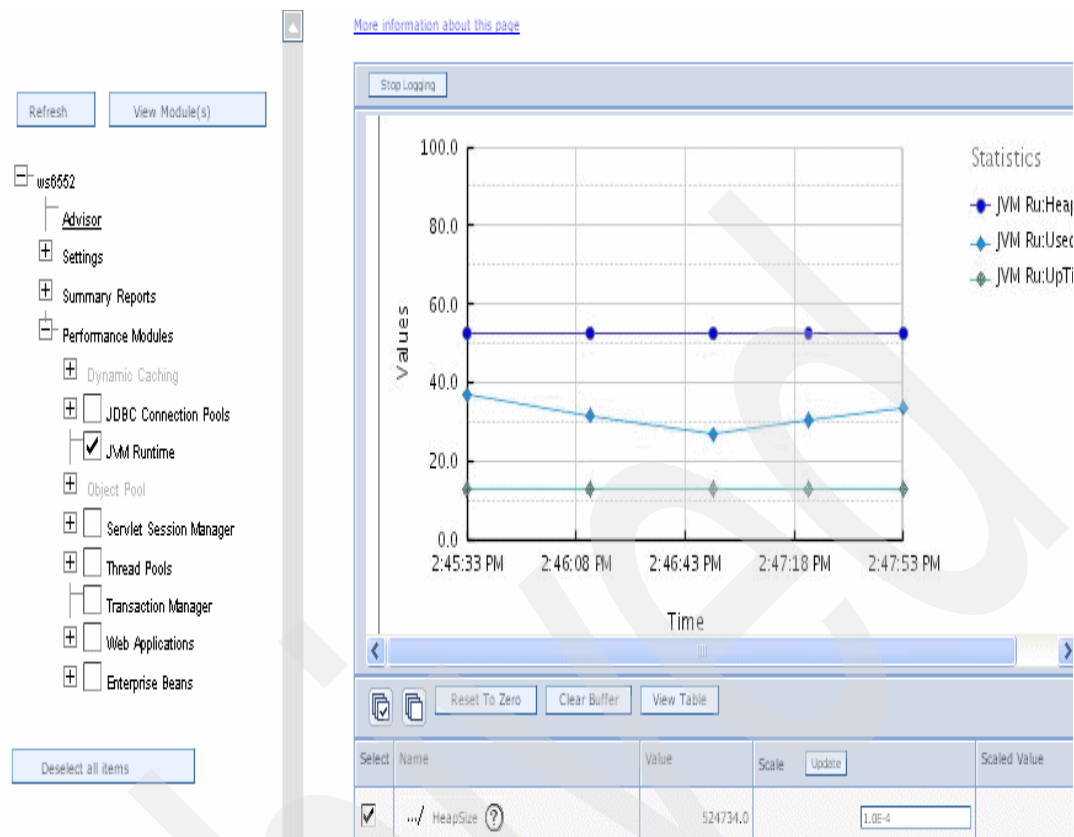


Figure 21-10 Tivoli Performance Viewer

21.8.2 Tivoli Performance Viewer output and its interpretation

In Tivoli Performance Viewer, you can see what is happening with the WebSphere for z/OS servers in real time. You can see, for example, if the number of concurrent threads in the Web container pool is high enough or if you must add more.

You must analyze the modules shown in Table 21-2 to identify problems in your WebSphere for z/OS configuration.

Table 21-2 Modules and description to verify in Tivoli Performance Viewer

Modules	Description
Average response time	Includes statistics such as servlet or Enterprise Beans response time
Number of request	Enables understanding of how much traffic is processed by WebSphere for z/OS, thus helping determine the capacity to manage
Web and EJB Thread Pools Database and connection pool size	Interpret together because these thread pools might constrain performance because of their size
JVM Memory	Use to understand the JVM heap dynamics, including the frequency of garbage collection

Request metrics is a tool (embedded in Tivoli Performance Viewer) that you can use to track individual transactions, recording the processing time in each of the major WebSphere Application Server components.

To enable request metrics from the Administrative Console:

1. Open the Administrative Console.
2. Select **Monitoring and Tuning** → **Request metrics** in the console navigation tree.
3. Select **Enable** in the Request metrics field under the Configuration tab.
4. Specify the components that are instrumented by request metrics.
5. Specify how much data to collect.
6. Enable and disable logging.
7. Enable Application Response Measurement (Application RM) Agent.
8. Specify which Application RM type to use.
9. Specify the name of the Application RM transaction factory implementation class.
10. Isolate performance for specific types of requests.
11. Add and remove request metrics filters.
12. Click **Apply** and **Save**.
13. Regenerate the Web server plug-in configuration file so that it recognizes the changes that you made for the request metrics configuration.

Another parameter in Tivoli Performance Viewer is thread pools. With Thread Pools, components of the server can reuse the threads and avoid the creation of new threads at run time. Creating new threads expends time and resources. Figure 21-11 shows a configuration panel for thread pool properties.

The screenshot shows a web browser window titled "Application servers". The breadcrumb navigation is "Application servers > ws6552 > Thread Pools > Default". Below the breadcrumb, there is a descriptive text: "A thread pool allows components of the server to reuse threads to eliminate the need to create new threads at runtime. Creating new threads is typically a time and resource intensive operation." The main configuration area is titled "Configuration" and is divided into two tabs: "General Properties" and "Additional Properties". Under "General Properties", there are several fields: "Name" (set to "Default"), "Description" (empty), "Minimum Size" (set to "5" threads), "Maximum Size" (set to "20" threads), and "Thread inactivity timeout" (empty milliseconds). There is also a checkbox labeled "Allow thread allocation beyond maximum thread size" which is currently unchecked. At the bottom of the "General Properties" tab, there are four buttons: "Apply", "OK", "Reset", and "Cancel". The "Additional Properties" tab is currently inactive, showing a "Custom Properties" link.

Figure 21-11 Configuration panel for thread pool properties

The maximum number of threads that you can create is constrained only by the limits of the JVM and the operating system. When a thread pool that can grow expands beyond the maximum size, the additional threads are not reused. They are discarded from the pool after the processing of the work items for which they were created is completed. When additional threads are created, a message is logged in the SYSOUT file to let you know that you went beyond the maximum size that was set for the thread pool.

Attention: The size of the thread pools constrain the performance, and setting the size of the thread pools too high impacts the amount of memory that is needed by the system.

For more detailed information about all available parameters, refer to “Monitoring performance with Tivoli Performance Viewer” at the WebSphere for z/OS Information Center:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

21.9 OMEGAMON XE for WebSphere

IBM Tivoli OMEGAMON XE for WebSphere for z/OS is a performance and availability monitoring tool. It provides real-time and historical management of WebSphere resources. It lets you monitor the performance of specific servlets, JSPs, and EJBs and take direct action to tune performance.

For further details about OMEGAMON XE for WebSphere Application Server for z/OS see:

<http://www.ibm.com/software/tivoli/products/omegamon-xe-was-zos/>

See also *IBM Tivoli OMEGAMON XE V3.1.0 Deep Dive on z/OS*, SG24-7155, which introduces the product and discusses problem determination, tracing concepts, and performance considerations with OMEGAMON facilities.

Archived

Other handy tools

In this chapter, we describe other tools that we have found very useful so that interested system administrators and application programmers for z/OS can have a quick reference guide to these tools. Although they are not directly related to problem determination for WebSphere Application Server for z/OS, the tools can be very powerful for performing day-to-day administrative tasks and problem determination for WebSphere for z/OS.

We describe TCP/IP-related tools such as TCP/IP checkout program (InetInfo.java) and TCP/IP network packet tracing with Ethereal. We also introduce MXI.

To gather performance data and create reports that help system programmers and administrators tune their systems optimally, react quickly to system delays, and diagnose and remediate performance problems, we introduce SMF records, RMF reports, and the SMF Browser.

To avoid problems caused by bottlenecks or programming issues, we recommend that you run load and stress tests with various tools for your applications and for your WebSphere for z/OS environment before you go into production. Examples of these tools include:

- ▶ WebSphere Studio Workload Simulator for z/OS and OS/390,
- ▶ Microsoft Web Application Stress tool

For the day-to-day administration tasks for WebSphere for z/OS, you also need powerful and efficient tools for managing Java, XML, or HTML files. Therefore, we finish the chapter by introducing FTP, Telnet, and editor tools such as:

- ▶ TeraTerm Pro
- ▶ Microsoft Windows FTP client
- ▶ WS_FTP Professional
- ▶ Directing SYSPRINT output to an HFS file
- ▶ UltraEdit

To build and service applications effectively in z/OS, developers need robust, easy-to-use tools to compile, test, and debug their applications. There are many more tools available to help them to face these challenges. For examples of specific IBM tools, see *Installing WebSphere Studio Application Monitor V3.1*, SG24-6491, which describes the Application Monitor, the Debug Tool, the Fault Analyzer, the File Manager, and the Workload Simulator in more detail.

22.1 TCP/IP related tools

Some of the TCP/IP-based tools can be very powerful for performing day-to-day administrative tasks and problem determination for WebSphere for z/OS. We describe the TCP/IP checkout program (InetInfo.java) and TCP/IP network packet tracing in this section.

It is always useful to know if the client actually sent what is expected and received what was sent from the server. The same is true for the server side. Usually, using tracing at the application server and container level is sufficient to obtain this message content. However, there are times when it is not possible to be certain (for example, if the server does not receive the request that was sent from the client). In these scenarios, the network packet tracer is an effective tool to apply.

TCP/IP packet tracing and analysis is a common approach for problem diagnosis at a network transport level. Various tools and techniques are available for handling all kinds of protocols and formats. These tools are usually provided as an operating system utility such as the TCP/IP for z/OS packet trace facility. Windows workstations do not have such a utility included in the OS, but various freeware utility packages are available, such as Ethereal.

22.1.1 TCP/IP checkout program (InetInfo.java)

InetInfo.java is a Java program that performs the following functions to display the Internet addressability for the host system:

getLocalHost	Obtain host IP address
getHostName	Using previously obtained IP address
getHostAddress	Using previously obtained host name

To use this tool, download the InetInfo.java program to your working directory in z/OS USS. Compile the Java program into the class file as follows:

```
>/usr/lpp/java/J1.4/bin/javac InetInfo.java
```

When you run the Java class, you get the following results, as shown in Figure 22-1:

1. The function getLocalHost returns an IP address.
2. The function getHostName by address returns a host name.
3. The function getHostAddress by name returns the IP address correctly.

```
>/Z16RA1/usr/lpp/java/J1.4/bin/javac InetInfo.java
>/Z16RA1/usr/lpp/java/J1.4/bin/java InetInfo

get Local Host
  IP Address: 9.12.4.28

get Host Name By Address using 9.12.4.28
  Host Name: wtsc55.itso.ibm.com

get Host Address By Name using wtsc55.itso.ibm.com
  Host Address: 9.12.4.28
```

Figure 22-1 InetInfo.java program output

If any of the functions fail, the following message is displayed:

Unknown Host, result: <returned message>

The <returned message> value is the reason for the exception.

You can copy or download this Java program from the Techdoc *Java program to test TCP/IP setup - InetInfo.java*, TD100609, available at:

<http://www.ibm.com/support/techdocs/>

22.1.2 TCP/IP network packet tracing with Ethereal

Ethereal is a network packet analyzer that is used for displaying and analyzing very detailed capture packet data with a GUI or command-line interface. Ethereal is open source software under the General Public License (GPL). Ethereal is available for UNIX, Linux, Windows, and Apple workstation environments.

Ethereal has the following attractive features:

- ▶ Analyzed data is captured in real time “off the wire” or read from a capture file.
- ▶ Real-time data can be from various network hardware devices for Ethernet, FDDI, PPP, Token-Ring, IEEE 802.11, Classical IP over ATM, and loopback interfaces.
- ▶ Data capture files can be from other tools such as UNIX tcpdump, Microsoft Network Monitor, and other hardware sniffers.
- ▶ Currently, it is able to dissect up to 706 network protocols.

To use Ethereal in a Microsoft Windows environment, you must download the Ethereal and WinPcap software packages from the Ethereal Web site and install them on your target workstation. Ethereal is a network packet analyzer and WinPcap is a network packet captor for Windows. WinPcap is only required if you also want to capture data on your workstation.

To use the Ethereal GUI program to start capturing packet data and analyzing what is captured, follow these steps:

1. Start your Ethereal program. Select **Start** → **Programs** → **Ethereal** → **Ethereal**.
2. Select **Capture** → **Start**.
3. This opens a Capture Options dialog box. Make sure that you select the correct interface card if you have more than one. Click **OK**. Another dialog box opens that shows you some capturing statistics. There is also a Stop button for stopping the capture.
4. Recreate the problem scenario, using the Web browser to access a page.
5. To stop capturing, click the Stop button on the Capture Info dialog box.

After the data is captured, you receive a list of network data packets. Ethereal provides many ways to analyze the captured traces. For example, you can:

- ▶ Control the amount of data that is displayed by selecting the display that interests you:
 - Attribute columns (hardware address, IP address, port, and so on)
 - Format of times (absolute, relative, delta, and so on)
 - Sort the list by any column
 - Apply your own display filter for selecting packet data
- ▶ Use the statistics to gain many kinds of higher perspective views, for example:
 - Overall summary of transmission
 - Summary by different endpoints (by hardware, IP, and so on)
 - Summary by protocol hierarchy
- ▶ Look into the formatted display of each data packet. Ethereal formats all kinds of protocol headers according to standard specifications.
- ▶ One of the most powerful tools is a feature called “Follow TCP Stream.” When looking at a TCP data packet, you can instruct Ethereal to analyze the user data portion of the packet

according to a higher-level application protocol specification such as HTTP. The data is formatted according to the HTTP header into separate sections inside other windows.

- In the data windows, you can display in ASCII, EBCDIC, hexadecimal, or C-array.

Figure 22-2 shows an Ethereal window that is analyzing a TCP data packet. The section display shows the details of all the formatted network headers of the TCP data packet. The Follow TCP Stream window shows the formatted HTTP header and HTML data.

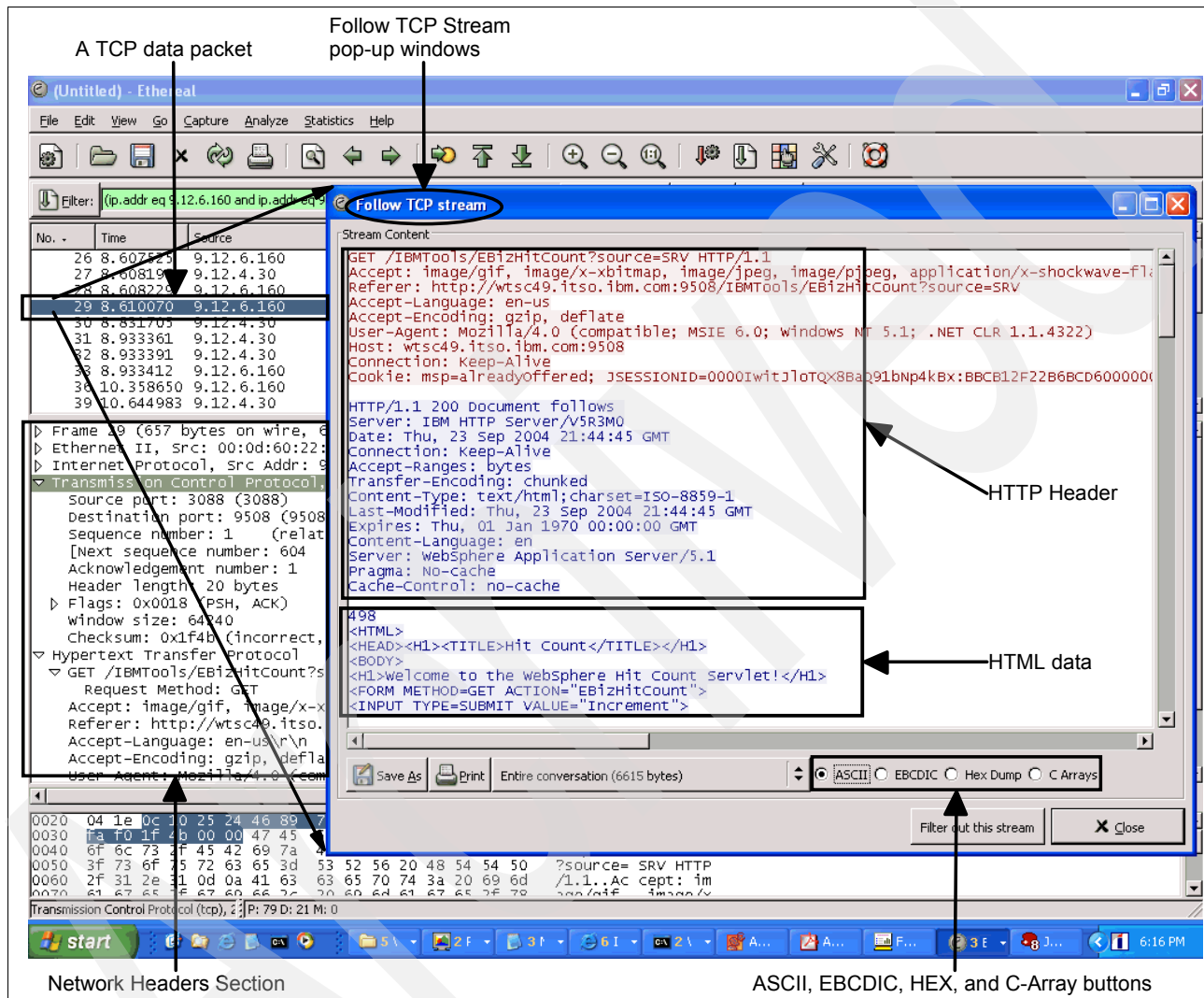


Figure 22-2 Ethereal data analysis with Follow TCP Stream windows

For more informal information about Ethereal, see:

<http://www.ethereal.com>

The source code and installers for Microsoft Windows, Red Hat Linux, Sun Solaris, IBM AIX, SUSE Linux, and more, can be found on this Web site:

<http://www.ethereal.com/download.html>

For more information about WinPcap, see:

<http://www.winpcap.org>

22.1.3 TCP/IP for z/OS packet trace

TCP/IP for z/OS packet trace is a z/OS facility for network packet capture and analysis. It uses a z/OS external writer to obtain component trace data for the TCP/IP stack, packet trace and other stack information. After the data is captured, you can use the z/OS IPCS tool to format the trace data for further analysis.

To capture trace data:

1. Set up an external writer JCL procedure (in SYS1.PROCLIB) to be used by the TCP/IP component trace (Example 22-1).

Example 22-1 External writer

```
//CTWTRPD  PROC
//IEFPROC  EXEC PGM=ITTTCWR,REGION=32M,TIME=1440
//TRCOUT01 DD DSN=SYS1.PROCLIB,DSN=WAS5PD.SC49.CTRACE,DISP=OLD
//SYSPRINT DD SYSOUT=*
```

2. Start the external writer using the following command:
/trace ct,wtrstart=CTWTRPD
3. Start the TCP/IP packet trace with filtering to pick up only one IP address:
/v tcpip,TCPIP,pkttrace,full,ip=9.12.6.160
4. When the system responds with a prompt, reply as follows:
/r xx,WTR=CTWTRPD,end
5. Use the DISPLAY command to check the external writer status (Example 22-2).

Example 22-2 Display trace status command

```
/d trace,comp=systcpda,sub=(TCPIP)
RESPONSE=SC49
IEE843I 15.02.26 TRACE DISPLAY 267
          SYSTEM STATUS INFORMATION
ST=(ON,0256K,00512K) AS=ON BR=OFF EX=ON MT=(ON,024K)
TRACENAME
=====
SYSTCPDA
                                MODE BUFFER HEAD SUBS
                                =====
                                OFF          HEAD    1
          NO HEAD OPTIONS
SUBTRACE          MODE BUFFER HEAD SUBS
-----
TCPIP             ON  0016M
ASIDS             *NONE*
JOBNAMES          *NONE*
OPTIONS           MINIMUM
WRITER            CTWTRPD
```

6. Recreate the problem scenario using the Web browser to access a page.
7. Stop the packet trace:
/trace ct,off,comp=systcpda,sub=(TCPIP)
8. Stop the external writer:
/trace ct,wtrstop=CTWTRPD

You use the IPCS utility to format the captured trace data into a user friendly format. During the format process, you have a choice of three levels of details: Summary, Short, and Full. Complete the following steps to format the trace data:

1. Access IPCS.
2. Select **2 (ANALYSIS)** from the option list.
3. Select **0 (DEFAULT)** from the option list and enter the trace data set name to be used as default source:
Source ==> DSNAME('WAS5PD.SC49.CTRACE')
Press **PF3** to return to the previous panel.
4. Select **7 (TRACES)** from the option list.
5. Select **1 (CTRACE)** from the option list.
6. Select **D (DISPLAY)** from the option list. Enter the component name, subsystem name, and trace detail level as shown in Figure 22-3. To start formatting, type **S** on the command line and press **Enter**.

```

----- CTRACE DISPLAY PARAMETERS
COMMAND ==>

System      ==>          (System name or blank)
Component   ==> SYSTCPDA (Component name (required))
Subnames    ==> TCPIP

GMT/LOCAL   ==> G          (G or L, GMT is default)
Start time  ==>          (mm/dd/yy, hh:mm:ss.dddddd or
Stop time   ==>          mm/dd/yy, hh.mm.ss.dddddd)
Limit       ==> 0          Exception ==>
Report type ==> FULL      (Short, Summary, Full, Tally)
User exit   ==>          (Exit program name)
Override source ==>
Options     ==>

To enter/verify required values, type any character
Entry IDs ==> Jobnames ==> ASIDs ==> OPTIONS ==> SUBS ==>

CTRACE COMP(SYSTCPDA) SUB(TCPIP) FULL

ENTER = update CTRACE definition.  END/PF3 = return to previous panel.
S = start CTRACE.  R = reset all fields.

```

Figure 22-3 IPCS CTRACE display parameters

Formatting the trace data with a **FULL** detail level results in information in the following sections:

- ▶ Interface device
- ▶ IP header
- ▶ TCP header
- ▶ Message data

Figure 22-4 on page 281 shows a generated report of one captured trace data packet. In the IP header, note the IP addresses (source and destination) and the date and time stamp. In the TCP header section, note the socket ports (source and destination).

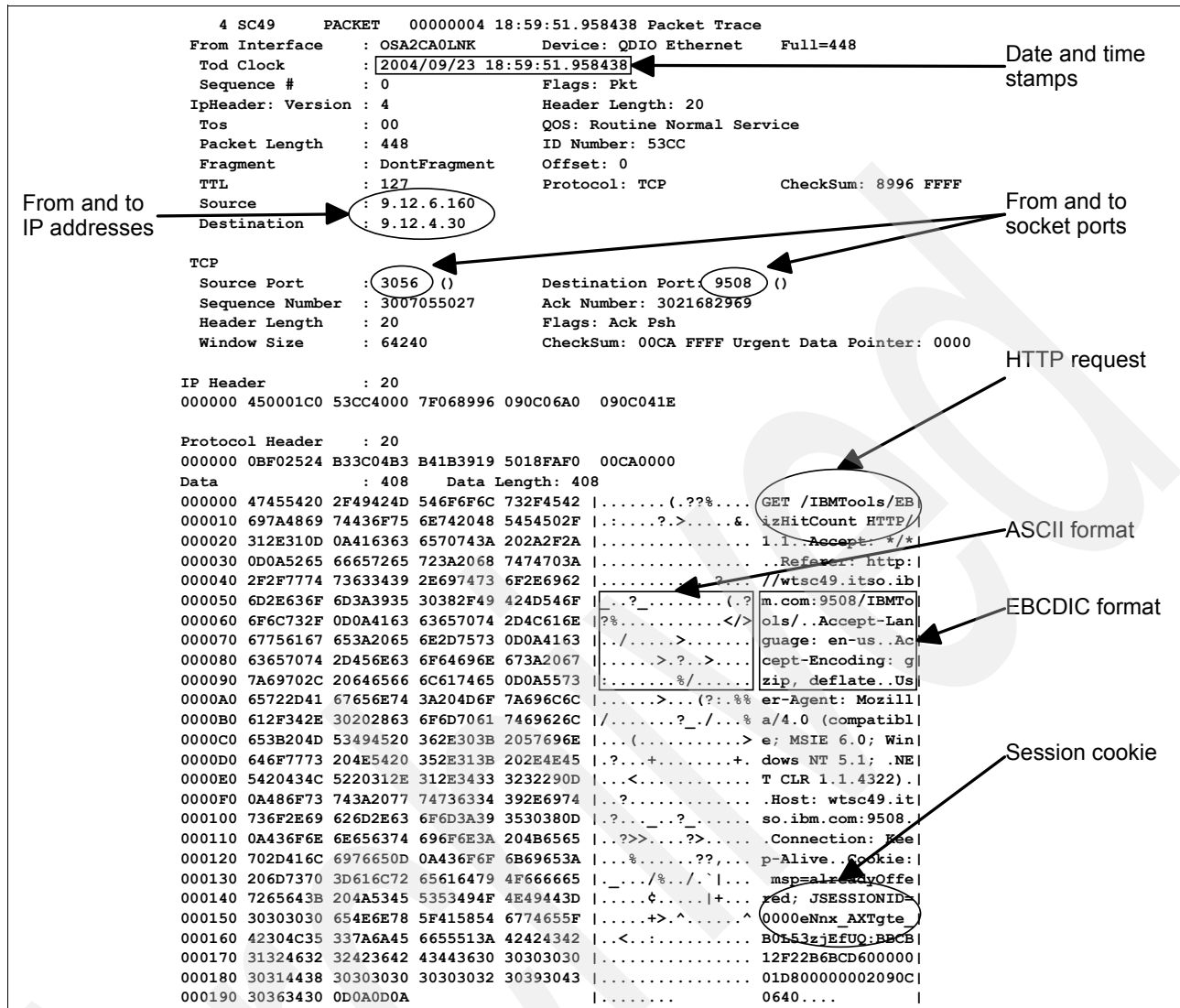


Figure 22-4 TCP/IP network packet trace report

For more information about the TCP/IP for z/OS packet trace, see *z/OS V1R5.0 Communication Server: IP Diagnosis Guide*, GC31-8782.

For more information about the IPCS tool, see *OS/390 V2R10.0 MVS Interactive Problem Control System (IPCS) User's Guide*, GC28-1756.

22.2 MVS Extended Information

MVS Extended Information (MXI) is an ISPF-based application that displays important configuration information about active OS/390 or z/OS systems. MXI is free software that is available from the following site:

<http://www.rocketsoftware.com/portfolio/mxi>

Although it is mainly used online, MXI also has a REXX interface and can be run in batch mode. It has a TCP/IP server application for issuing commands to remote systems and viewing the results locally.

MXI can display a wealth of information from your system, including:

- ▶ APF, LNKLST, and LPA data sets
- ▶ Active address spaces and ASVT slot usage
- ▶ Allocated data sets for any address space
- ▶ Master and user catalogs
- ▶ Common storage usage by address space or subpool
- ▶ Orphaned common storage
- ▶ Cross-memory connections
- ▶ CPU and LPAR information
- ▶ Online DASD and tape units
- ▶ Enqueue requests and contention
- ▶ HSM request queues
- ▶ ISPF screen images of any user
- ▶ LLA module statistics
- ▶ Memory contents of any address space
- ▶ Memory delete queue
- ▶ Real and auxiliary storage usage
- ▶ SMS classes
- ▶ SMS storage groups
- ▶ Subsystems
- ▶ SVCs and PC routines
- ▶ Sysplex information
- ▶ WLM information

Figure 22-5 shows an excerpt of the MXI Primary Option menu.

File	Datasets	Modules	Units	System	Sysplex	SMS	Storage	Tools	RACF

MXI - MENU - IG01 - HOME ----- CPU 15 UIC 254 PAG 0 ----- Row 1 of 49									
Command ==> Scroll ==>									
<u>AGRP</u>	SMS Aggregate Group Information				<u>NTOK</u>	System Name/Token Information			
<u>APF</u>	APF List Dataset Information				<u>NUC</u>	Display System Nucleus Modules			
<u>ASID</u>	Address Space Usage Information				<u>OMVS</u>	OpenEdition Configuration			
<u>CAT</u>	Catalog Information				<u>PAGE</u>	Page Dataset Information			
<u>CA1</u>	CA-1 Configuration Information				<u>PARM</u>	Active PARMLIB Information			
<u>CDE</u>	JPAQ and TCB loaded modules				<u>PEEK</u>	Show ISPF Screens			
<u>CF</u>	Coupling Facility Information				<u>PID</u>	OpenEdition Processes			
<u>CHP</u>	Online Channel Paths				<u>PLEX</u>	Display Sysplex Information			
<u>CPF</u>	Command Prefix Table				<u>PPT</u>	Program Properties Information			
<u>CPU</u>	CPU and LPAR Information				<u>RACF</u>	RACF Information			
<u>CS</u>	Common Storage Usage				<u>RCLS</u>	RACF Class Information			
<u>CSR</u>	Common Storage Remaining				<u>RL</u>	RACF Profile Information			
<u>DA</u>	Active Address Space Information				<u>RS</u>	Real Storage Usage Information			
<u>DAE</u>	Dump Elimination Information				<u>RSYS</u>	Remote System Selection			
<u>DASD</u>	Online DASD Information				<u>SCLS</u>	SMS Storage Class Information			
<u>DCLS</u>	SMS Data Class Information				<u>SGRP</u>	SMS Storage Group Information			
<u>DDNS</u>	Allocated Dataset Information				<u>SMF</u>	SMF General Information			
<u>DEU</u>	DASD Activity				<u>SMFD</u>	SMF Dataset Information			
<u>DS</u>	Personal Dataset List				<u>SMS</u>	SMS Configuration Information			
<u>DSP</u>	Dataspace Information				<u>SMSM</u>	SMS Module Map			
<u>DYNX</u>	Dynamic Exit Information				<u>SOFT</u>	System Software Levels			
<u>EDT</u>	Display EDT Information				<u>SP</u>	Common Storage Subpool Usage			
<u>EMCS</u>	E-MCS Information				<u>SPD</u>	Subpool Definitions			
<u>ENQ</u>	Display ENQ Information				<u>SSI</u>	Subsystem Information			
<u>ENQC</u>	Display ENQ Contention				<u>SRUC</u>	WLM Service Class Information			
<u>EXC</u>	System Exceptions				<u>STOR</u>	System Storage Information			
<u>GRS</u>	GRS Resource Name Lists				<u>SUC</u>	SUC Information			
<u>HFS</u>	OpenEdition File Systems				<u>SYH</u>	System Symbols			
<u>HSM</u>	HSM Configuration				<u>SYSX</u>	System Exit Information			
<u>HSMQ</u>	HSM Request Queues				<u>TAPE</u>	Online TAPE Information			
<u>INIT</u>	JES2 Initiators				<u>TCB</u>	TCB and RB Information			

Figure 22-5 MXI Primary Option menu

Most of the displays can be filtered using ISPF-like masking characters, and many display fields have “point-and-shoot” functionality that drills down to a more detailed display.

22.3 Resource Measurement Facility reports

IBM Resource Measurement Facility window (RMF) is designed to simplify management of single and multiple system workloads. RMF gathers data and creates reports that help system programmers and administrators to tune their system optimally, react quickly to system delays, and diagnose and remediate performance problems.

This section describes how to use standard RMF reports and simple arithmetic to produce performance information. RMF reports do not give application information, but they can be used to obtain system and workload characteristics.

RMF stores z/OS data in System Management Facility (SMF) records if you define the appropriate SMF recording options. For performance analysis, and depending on your software environment, the following SMF records might be especially helpful:

- ▶ Record type 70 to 79: RMF records (for example, record type 70 for processor activity, record type 72 for workload activity)
- ▶ Record type 88: System logger activity
- ▶ Record type 92: USS HFS information
- ▶ Record type 103: HTTP server information
- ▶ Record type 100 to 102: DB2 statistics, accounting, performance
- ▶ Record type 110: CICS TS Statistics
- ▶ Record type 115, 116: WebSphere MQ Statistics
- ▶ Record type 118, 119: TCP/IP Statistics
- ▶ Record type 120: WebSphere Application Server information

Important: Some of the records might be resource intensive to collect. Refer to the documentation for each subsystem for more information about using SMF records for DB2, CICS, WebSphere MQ, or TCP/IP.

There are various RMF interfaces that allow customized views, such as the RMF PM Java edition GUI. We chose the traditional RMF post processor with its predefined formats to document our approach in a general way.

For more details about the RMF functionality and instructions for setting up and customizing the components, see *Effective zSeries Performance Monitoring Using Resource Measurement Facility*, SG24-6645. This redbook puts special emphasis on the newest features such as Spreadsheet Reporter, Distributed Data Server, Linux data gatherer, and Performance Monitor.

Because we are focusing on WebSphere for z/OS performance, we simplified monitoring by grouping the WebSphere activities into predefined report classes:

WAS	For WebSphere infrastructure (controller region).
WASS	For Servant Regions.
WASE	For On Demand Business workloads running in the servant regions in enclaves.

WASC	For CICS regions called upon by WebSphere transactions.
WASD	For DB2.
OTHER	For other activity not directly related to our WebSphere environment. Because our exercise was to illustrate a production environment as opposed to a lab “controlled” environment, this was done to isolate started tasks, systems management tasks, TSO users, and so forth that were concurrently active in the sysplex.

22.3.1 Running the RMF post processor

The RMF post processor writes reports from the data that is collected and saved to the SMF data sets. This data is written, typically, at 15-minute or 30-minute intervals. The data contains very detailed information about the sysplex, systems, CPCs, and workloads.

Example 22-3 shows the JCL that we used to produce the RMF reports.

Example 22-3 JCL for running the post processor

```
//RMFRPT52 JOB (999,POK),'FRANCK',CLASS=A,REGION=4096K,
//          MSGCLASS=T,TIME=90,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//RMFSORT  EXEC PGM=SORT,REGION=0M
//***** SORTIN DATA SETS FOLLOWING HERE *****
//SORTIN   DD  DISP=SHR,
//          DSN=FRANCK.SMF.D06T1700
//SORTOUT  DD  DISP=(NEW,PASS),DSN=&&SORTOUT,UNIT=SYSALLDA,
//          SPACE=(CYL,(50,50)),DCB=*.RMFSORT.SORTIN
//SORTWK01 DD  DISP=(NEW,DELETE),
//          DSN=&&WK1,UNIT=SYSALLDA,SPACE=(CYL,(50,50))
//SORTWK02 DD  DISP=(NEW,DELETE),
//          DSN=&&WK2,UNIT=SYSALLDA,SPACE=(CYL,(50,50))
//SORTWK03 DD  DISP=(NEW,DELETE),DSN=&&WK3,
//          UNIT=SYSALLDA,SPACE=(CYL,(50,50))
//SYSPRINT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//SYSIN    DD  *
          SORT FIELDS=(11,4,CH,A,7,4,CH,A),EQUALS
          MODS E15=(ERBPPSRT,500),E35=(ERBPPSRT,500)
//POST1    EXEC PGM=ERBRMFPP
//MFPINPUT DD  DSN=*.RMFSORT.SORTOUT,DISP=(OLD,PASS)
//* REPORTS (CHAN)
//* REPORTS (ENQ)
//* REPORTS (IOQ)
//* REPORTS (PAGING)
//* REPORTS (DEVICE(DASD))
//* REPORTS (OMVS,HFS)
//*
//SYSIN    DD  *
          RTOD(0000,2400)
          STOD(0000,2400)
          REPORTS (CPU)
          SUMMARY (INT)
          SYSOUT(T)
//POST2    EXEC PGM=ERBRMFPP
//MFPINPUT DD  DSN=*.RMFSORT.SORTOUT,DISP=(OLD,PASS)
//* SYSRPTS (WLMGL(POLICY(FRANCK.LSM301_1)))
```

```

/**
//SYSIN DD *
RTOD(0000,2400)
STOD(0000,2400)
SUMMARY (INT)
SYSRPTS (WLMGL(RCLASS(WAS*,OTHER,SYS*)))
SYSRPTS (WLMGL(POLICY,SCPER(WAS*)))
SYSOUT(T)

```

22.3.2 Analyzing RMF reports

RMF reports provide resource utilization information such as:

CPU	The partition data report gives the logical partition view. The summary report and CPU report show the z/OS system view, while the workload report provides a breakdown by workload type.
Storage	The summary and CPU reports show the z/OS system view. The workload report provides storage allocation information by workload type.
I/O activity	The summary report, CPU report, and IOQ report shows system level indicators. The workload report provides information by workload type.

Additional resource reports for channels, paging, and virtual storage, can be further investigated.

CPU activity report

Figure 22-6 shows a sample CPU activity report. To determine the utilization of the processors, see the LPAR BUSY TIME columns for LPAR modes and MVS BUSY TIME for basic mode. The values report the percentage of time that all processors were busy during the RMF measurement interval.

C P U A C T I V I T Y														
z/OS V1R3				SYSTEM ID SC48				DATE 11/17/2002				INTERVAL 05.00.509		
				RPT VERSION V1R2 RMF				TIME 16.54.59				CYCLE 1.000 SECONDS		
CPU 2064	MODEL 2C7													
CPU	ONLINE TIME	LPAR BUSY	MVS BUSY	CPU SERIAL	I/O TOTAL	% I/O INTERRUPTS								
NUMBER	PERCENTAGE	TIME PERC	TIME PERC	NUMBER	INTERRUPT RATE	HANDLED VIA TPI								
0	100.00	90.17	95.15	0B0ECB	431.3	0.75								
1	100.00	90.16	95.13	1B0ECB	429.6	0.78								
TOTAL/AVERAGE		90.16	95.14	861.0		0.76								
SYSTEM ADDRESS SPACE ANALYSIS				SAMPLES = 301										
NUMBER OF ASIDS				DISTRIBUTION OF QUEUE LENGTHS (%)										
TYPE	MIN	MAX	AVG	0	1	2	3	4	5	6	7-8	9-10	11-12	13-14
----	-----	-----	-----	----	----	----	----	----	----	----	----	----	----	----
IN														
READY	1	12	4.0	0.0	6.9	3.3	12.2	49.1	21.5	4.6	1.6	0.0	0.3	0.0

Figure 22-6 CPU Activity Report (partial view)

We run in LPAR mode, so our report told us that the partition SC48 is running at 90.16% CPU busy. Although it is common to talk about a “CPU being p% busy,” this is an abbreviated statement that has no physical reality. At any time, a CP only has two operational states:

- Busy (that is, 100 percent busy)
- Idle (that is, 0 percent busy)

All CPU percentages in the RMF reports are relative to the RMF measurement time interval. The CPU percentages that are reported express the amount of time that the CPU was busy over the measurement interval. Hence, the correct way to understand the report really reads, “the CPU is 100% busy p% of the time.”

Also check the IN READY line. This is the dispatching queue, that is, the work that is in the system and ready to be dispatched.

In the example in Figure 22-6 on page 285, no immediate CPU contention is visible. Although CPU is 90% busy (something not unusual in a z/OS environment), the IN READY queue length remains below three times the number of CPs for more than 90% of the time. This might not be a problem if there are non-time-critical batch jobs running in the background.

Partition data report

The RMF partition data report is embedded in the CPU activity report when the server is running in LPAR mode. To access the report information for all partitions, you must be authorized. To do this, enable the Global Performance Management Control setting in the partition activation profile from the zSeries Hardware Management Console.

The partition data report section contains header information, partition data, logical partition processor data, and average processor utilization percentages.

Figure 22-7 shows a partial view of a partition data report.

P A R T I T I O N D A T A R E P O R T									
z/OS V1R3			SYSTEM ID SC48			DATE 11/17/2002		INTERVAL 05.00.509	
			RPT VERSION V1R2 RMF			TIME 16.54.59			
MVS PARTITION NAME			A11						
IMAGE CAPACITY			171						
NUMBER OF CONFIGURED PARTITIONS			15						
NUMBER OF PHYSICAL PROCESSORS			13						
CP			7						
ICF			6						
WAIT COMPLETION			NO						
DISPATCH INTERVAL			DYNAMIC						
----- PARTITION DATA -----									
....									
-- AVERAGE PROCESSOR UTILIZATION PERCENTAGES --									
-----MSU-----				LOGICAL PROCESSORS		--- PHYSICAL PROCESSORS ---			
NAME	S	WGT	DEF	EFFECTIVE	TOTAL	LPAR	MGMT	EFFECTIVE	TOTAL
A1	A	180	45	4.65	5.05	0.11		1.33	1.44
A2	A	10	30	4.52	4.91	0.11		1.29	1.40
A3	A	180	0	4.52	4.92	0.11		1.29	1.41
A4	A	10	0	0.78	0.92	0.04		0.22	0.26
A5	A	10	45	4.12	4.52	0.11		1.18	1.29
A6	A	10	0	10.19	10.59	0.12		2.91	3.03
A7	A	10	45	4.52	4.97	0.13		1.29	1.42
A8	A	10	0	3.99	4.38	0.11		1.14	1.25
A9	A	10	0	3.60	3.99	0.11		1.03	1.14
A10	A	10	0	3.31	3.71	0.11		0.95	1.06
A11	A	180	0	89.93	90.16	0.07		25.69	25.76
A12	A	10	50	0.91	1.04	0.04		0.26	0.30
PHYSICAL						4.19		4.19	
TOTAL						5.36		38.58 43.95	

Figure 22-7 Partition Data Report (partial view)

In the example report notice, that:

- ▶ The running partition is A11.
- ▶ It is using 90.16% of its logical CPs or 25.76% of the server CP capacity.

The zSeries CPs are only used 43.95% of the time.

The *Physical Management Time* that is reported by RMF in the PHYSICAL* line indicates the amount of processor time that is required to manage all active LPARs. The partition that is named PHYSICAL does not exist; the line is created by RMF for reporting purposes.

The logical partition *Dispatch Time Effective* that is indicated for each configured partition (Figure 22-8) is the sum of the z/OS captured time and the z/OS uncaptured time. The Partition LPAR Management Time is *not* a collected value, but is calculated by:

DISPATCH TIME DATA EFFECTIVE - DISPATCH TIME DATA TOTAL.

z/OS V1R3		SYSTEM ID SC48		DATE 11/17/2002					
				RPT VERSION V1R2 RMF	TIME 16.54.59				
MVS PARTITION NAME		A11							
IMAGE CAPACITY		171							
NUMBER OF CONFIGURED PARTITIONS		15							
NUMBER OF PHYSICAL PROCESSORS		13							
CP		7							
ICF		6							
WAIT COMPLETION		NO							
DISPATCH INTERVAL		DYNAMIC							
----- PARTITION DATA -----									
-- LOGICAL PARTITION PROCESSOR									
-----DISPATCH TIME									
NAME	S	WGT	DEF	ACT	DEF	WLM%	NUM	TYPE	EFFECTIVE
A1	A	180	45	4	NO	0.0	2	CP	00.00.27.930
A2	A	10	30	4	NO	0.0	2	CP	00.00.27.137
A3	A	180	0	4	NO	0.0	2	CP	00.00.27.192
A4	A	10	30	1	NO	0.0	2	CP	00.00.04.705
A5	A	10	45	4	NO	0.0	2	CP	00.00.24.785
A6	A	10	0	9	NO	0.0	2	CP	00.01.01.222
A7	A	10	45	4	NO	0.0	2	CP	00.00.27.174
A8	A	10	0	4	NO	0.0	2	CP	00.00.23.989
A9	A	10	0	3	NO	0.0	2	CP	00.00.21.648
A10	A	10	0	3	NO	0.0	2	CP	00.00.19.890
A11	A	180	0	77	NO	0.0	2	CP	00.09.00.486
A12	A	10	50	1	NO	0.0	2	CP	00.00.05.489
PHYSICAL									

TOTAL									
									00.13.31.653
C1	A	DED	0	86		0.0	2	ICF	00.10.00.977
C2	A	DED	0	86		0.0	2	ICF	00.10.00.774
C3	A	DED	0	86		0.0	2	ICF	00.10.00.644
PHYSICAL									

Figure 22-8 Partition Data report and processing weights (partial view)

Note that the flexibility that is brought by logical partitioning adds an additional level of complexity to the performance analysis; unless the LPAR is “capped”, the amount of CPU processing power that the partition can use can vary:

- ▶ The minimum CPU that the logical partition (LP) is entitled to is determined by the processing weights set as part of the partitioning definition:

Min LP CP share = Your LP weights / sum of all LP weights

This occurs when other partitions require their full share of CP resources. In Figure 22-8 on page 287, the sum of all WGT is 630, while our logical partition (A11) has a processing weight of 180. That means that the guaranteed CP share is $180/630 = 28.57\%$ of the shared CPs.

- The maximum CPU that the logical partition can use is fixed by the ratio of the number of CPs that is defined in the partition to the total number of available CPs in the shared pool:

Max LP CP share = number of CPs / sum of shared CPs

This occurs when other partitions do not need their full share of CPU resources. In Figure 22-8 on page 287, there are seven shared CPs available while our LP (A11) has 2 CPs defined. This means that the maximum usable CPU share is $2/7 = 28.57\%$ of the shared CPs.

In this example, we were able to align both the minimum and maximum values to simplify our tests, but in a real production environment this might not always be possible, nor desirable. Additionally, if the partition is part of an LPAR cluster (the set of LPARs in a single server that belongs to the same parallel sysplex), WLM can dynamically adjust the number of logical processors and the weight of an LPAR. This allows the system to distribute the CPU resources in an LPAR cluster to partitions where the CPU demand is high.

Because the processing weights can be dynamically adjusted, either by operations personnel or by LPAR cluster management, remember to check their settings before you start a time consuming workload analysis.

Note: All percentages indicated in the partition data report are relative to the RMF time interval. As such, they accurately show the amount of time that physical CPs were dispatched on behalf of a LPAR. However, these time-based figures do not take into account all processor costs of operating in LPAR mode and do not reflect the resulting processor power that is expressed in the Large System Performance Reference (LSPR) ITRs or MIPS.

The LPAR Capacity Estimator (LPARCE) tool should be run to estimate the impact of the LPAR configuration on the processing power. Consult your IBM support representative to obtain an LPARCE review for your configuration.

Summary report

This report (Figure 22-9) provides a summary view of the entire systems activity over multiple measurement intervals.

R M F S U M M A R Y R E P O R T																								
PAGE 001		z/OS V1R3				SYSTEM ID SC48				START 11/17/2002-16.24.59				INTERVAL 00.04.59										
RPT VERSION V1R2 RMF										END 11/17/2002-17.00.00				CYCLE 1.000 SECONDS										
NUMBER OF INTERVALS 7																								
DATE	TIME	INT	CPU	DASD	DASD	JOB	JOB	TSO	TSO	STC	STC	ASCH	ASCH	OMVS	OMVS	SWAP	DEMAND							
MM/DD	HH.MM.SS	MM.SS	BUSY	RESP	RATE	MAX	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX	AVE	RATE	PAGING							
11/17	16.24.59	05.00	64.2	2	264.2	0	0	2	2	110	109	0	0	6	5	0.00	0.20							
11/17	16.30.00	05.00	6.6	8	19.8	0	0	2	2	109	109	0	0	5	5	0.00	0.07							
11/17	16.35.00	04.59	20.0	4	69.0	0	0	2	2	110	109	0	0	5	5	0.00	0.63							
11/17	16.39.59	05.00	9.5	4	50.7	0	0	2	2	109	109	0	0	5	5	0.00	0.11							
11/17	16.45.00	04.59	10.6	4	46.6	0	0	2	2	108	108	0	0	5	5	0.00	0.02							
11/17	16.50.00	04.59	74.8	2	277.9	0	0	2	2	109	108	0	0	5	5	0.00	0.09							
11/17	16.54.59	05.00	90.2	2	328.3	0	0	2	2	109	109	0	0	5	5	0.00	0.23							

Figure 22-9 Summary Report

When you know your average system statistics, it is a very useful report for quickly spotting unusual behavior regarding:

- ▶ CPU busy
- ▶ DASD rate, that is, disk I/O activity per second
- ▶ Swap rate and paging demand

Workload reports

The RMF workload activity report contains information about your workload. The interpretation of the numbers depends on whether you are reporting a workload, a service class, or a reporting class. We strongly recommend using reporting classes.

Enclave report

An enclave report is a workload report for a reporting class that is associated with a WebSphere workload that is running in enclaves. It corresponds to the WLM definitions in the CB subsystem. Figure 22-10 and Figure 22-11 on page 290 show parts of a sample report.

REPORT BY: POLICY=LSA510			REPORT CLASS=WASE		
			DESCRIPTION =LSA510 WAS EBUSINESS WORKLOAD		
TRANSACTIONS		TRANS.-TIME	HHH.MM.SS.TTT	--DASD I/O--	
AVG	2.28	ACTUAL	147	SSCHRT	1.5
MPL	2.28	EXECUTION	101	RESP	1.8
ENDED	6777	QUEUED	45	CONN	1.2
END/S	22.59	R/S AFFINITY	0	DISC	0.3
#SWAPS	0	INELIGIBLE	0	Q+PEND	0.3
EXCTD	0	CONVERSION	0	IOSQ	0.0
AVG ENC	2.28	STD DEV	201		
REM ENC	0.00				
MS ENC	0.00				

Figure 22-10 Workload activity (part 1)

In these examples:

- ▶ AVG is the average number of active transactions during the interval.
- ▶ MPL is the average number of transactions in storage during the measurement interval.
- ▶ ENDED is the number of transactions that ended during the interval, and END/S is the number of transactions that ended per second. If the reporting class is set up correctly, this is a direct measure of the application throughput as seen by WebSphere.
- ▶ AVG ENC is the average number of enclaves concurrently active at any time. This information can be useful for sizing storage requirements or system recovery aspects.
- ▶ The DASD I/O section indicates the profile of the disk activity in your workload. High values for DISC, Q+PEND, or IOSQ might indicate an elongated response time.

The SSCHRT field indicates the disk start subchannel rate, in numbers per second. From this section, you can detect a possible delay caused by I/O activity to the disk subsystem.

By comparing this value with the DASD I/O column in the Summary Report, it is possible to quantify to what extent the WebSphere application participates in the I/O activity and possibly determine whether some system tuning actions are required.

- ▶ TRANS.-TIME contains the transaction time in HHH.MM.SS.TTT units that is seen by WLM. This is from the time that the transaction is put in the Servant Region WLM queue until the time the transaction is completed:

- ACTUAL is the actual amount of time required to complete the work submitted under the service class. This is the total response time.
- QUEUED is the average time that the WebSphere transaction was delayed in the WLM queue. The time can increase under full load conditions if the number of servers in MAX_SRS is too low.
- STD DEV is the standard deviation of ACTUAL. It is a measure of variability of the data in the sample. The higher the standard deviation, the more spread-out it looks on a graph (Figure 22-11).

REPORT BY: POLICY=LSA510				REPORT CLASS=WASE			
				DESCRIPTION =LSA510 WAS EBUSINESS WORKLOAD			
--DASD I/O--		--SERVICE RATES--		PAGE-IN RATES		---STORAGE----	
SSCHRT	1.5	ABSRPTN	38580	SINGLE	0.0	AVG	0.00
RESP	1.8	TRX SERV	38580	BLOCK	0.0	TOTAL	0.00
CONN	1.2	TCB	229.7	SHARED	0.0	CENTRAL	0.00
DISC	0.3	SRB	0.0	HSP	0.0	EXPAND	0.00
Q+PEND	0.3	RCT	0.0	HSP MISS	0.0		
IOSQ	0.0	IIT	0.0	EXP SNGL	0.0	SHARED	0.00
		HST	0.0	EXP BLK	0.0		
		APPL %	76.6	EXP SHR	0.0		

Figure 22-11 Workload activity (part 2)

- Note that the STORAGE field is always 0 for an enclave type report. Since enclaves are not associated with a specific address space, no storage values are reported.
- The APPL% field indicates the CPU activity incurred on behalf of all activities that are part of the enclave. It is expressed as a percentage of CP time used over the interval. Note that this represents *all* the CPU activity across all address spaces that are spanned by the transaction, including DB2 and CICS if the transaction contains JDBC or JCA connectors.

No activity (or response time) information is reported by WLM in the CICS assigned service class or report class.

- From the above fields, it is possible to calculate the average CP cost per transaction. Using APPL%, the measurement interval length expressed in milliseconds and the number of ended transactions over the interval are multiplied:

$$CP_millisecPerTran = Interval_length \text{ in milliseconds} * APPL\% / 100 / ENDED$$

Note that there are now multiple APPL% values to show zAAP activity as well.

Using the RMF fields for the WASE report class in Figure 22-10 on page 289, you can determine the following values for the specific measurement interval:

- 2.28 transactions were concurrently active, all of them running in enclaves.
- A total of 6777 transactions ended, which translates into an average throughput of 22.59 transactions per second.
- The average response time was 147 ms, with a standard deviation of 201 ms.
- For the measurement interval, APPL% shows that one CP was busy 76.6% of the time to service WASE. Because the measurement interval is 5 minutes, this translates as:

$$\text{Used CP time} = 300 \text{ sec} \times .766 = 229.8 \text{ sec}$$

Over the same interval, 6777 transactions were processed. The average CP cost is:

$$CP_MillisecPerTran = 229.8 \times 1000 / 6777$$

$$CP_MillisecPerTran = 33.90 \text{ ms}$$

Address space report

An address space report (Figure 22-12) is a workload report for a servant region (not running in enclaves) if you defined a report class. Server address space activity should be assigned to a service class in the STC group. This processing time is associated with garbage collector, or memory leak.

REPORT BY: POLICY=LSA510				REPORT CLASS=WASS			
TRANSACTIONS		--SERVICE RATES--		DESCRIPTION =LSA510 WAS SERVER AS ACTIVITY		PAGE-IN RATES	
AVG	2.00	ABSRPTN	181961	SINGLE	0.0	AVG	56146.9
MPL	2.00	TRX SERV	181961	BLOCK	0.0	TOTAL	112293
ENDED	0	TCB	8.0	SHARED	0.0	CENTRAL	112293
END/S	0.00	SRB	0.3	HSP	0.0	EXPAND	0.00
#SWAPS	0	RCT	0.0	HSP MISS	0.0		
EXCTD	0	IIT	0.0	EXP SNGL	0.0	SHARED	3216.83
AVG ENC	0.00	HST	0.0	EXP BLK	0.0		
REM ENC	0.00	APPL %	2.8	EXP SHR	0.0		
MS ENC	0.00						

Figure 22-12 Workload report for WebSphere server address space (partial)

There are three major differences in the interpretation of the data, because the reported activity is address-space based:

- ▶ The TRANSACTION AVG indicates the number of Servant Region address spaces that are active over the interval. Using this field, you can monitor the evolution of the number of servers between the MIN_SRS and MAX_SRS settings.
- ▶ STORAGE values are now provided.
- ▶ Under normal conditions, the APPL% is typically very low. However, a gradual increase in APPL% might be an indication of excessive garbage collector activity caused by a heap size that is too small, or a memory leak.

Using workload definitions, it is possible to calculate the system uncaptured percentage value. This is the part of CP resources that is used by system-related services on behalf of the workloads but not directly accounted for in the enclave or address space activity:

1. For each member in the sysplex, multiply the CPU_Busy% obtained from the CPU report by the number of CPs available to the z/OS LPAR. This brings the percentage value to a unit consistent with the APPL% reported in the workload report. Then, the sum for all systems participating in the sysplex is:

$$All_CP_Busy\% = \text{Sum of } [CPU_Busy\% * \text{Number of CPs}]$$

2. From the RMF Workload Activity report, obtain the total CP utilization that has been reported for all workloads. This is indicated by the APPL% value for the policy. The report is obtained when option WLMGL(POLICY) is specified. The APPL% value for the policy represents the percentage of time that any CP in the sysplex configuration was busy processing a workload that was defined in the WLM policy:

$$ALL_Wk1\% = \text{APPL\% from RMF Policy report}$$

3. The uncaptured CP value, expressed in percentage of CP activity over the measurement interval, is calculated by subtracting ALL_Wk1% obtained in step 2 from All_CP_Busy% calculated in step 1:

$$\text{uncaptured_CP\%} = All_CP_Busy\% - ALL_Wk1\%$$

Typically, the uncaptured CP% represents 10% to 20% of the total CP utilization.

The workload report provides response times for all service class periods and response time distribution information. The response time distribution (Figure 22-13) is provided per service class, for each service where a response time objective is defined. This is much more meaningful to the performance analyst than the average response time value.

Figure 22-13 Response time distribution (partial view)

From this information, it is also possible to set an achievable percentile response time, a value commonly used in establishing service level agreements.

For more information about RMF reports, see the following manuals:

- ▶ *z/OS Resource Measurement Facility Report Analysis*, SC33-7991.
- ▶ *z/OS Resource Measurement Facility User's Guide*, SC33-7990
- ▶ *z/OS Resource Measurement Facility Performance Management Guide*, SC33-7992

These are available in the Elements and Features list for your specific z/OS version at:

<http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/>

See the following Web sites:

<http://www.ibm.com/servers/eserver/zseries/zos/rmf/>
<http://www.ibm.com/servers/eserver/zseries/zos/wlm/>

22.4 System Management Facility records and browser

This section shows you how to enable and use System Management Facility (SMF) to collect and record system and job-related information. 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 might require.

You can enable SMF recording for:

- ▶ Capacity planning:
 - To determine the number of transactions that have run
 - To determine the average and maximum completion time for methods running on each server
 - To determine the number of clients that are attached to each server instance and the number that are active
- ▶ Application profiling:
 - To show an application broken down into its component parts
 - To provide timing information about the component parts of the application
- ▶ Error reporting:
 - To detect and record soft failures (those that are generated through an exception or those that are performance related)
 - To trigger an event that will cause an action to occur after a threshold has been reached

22.4.1 Setting up SMF recording

Follow these steps to enable SMF recording for WebSphere Application Server and select SMF type 120 records for output to the SMF data sets:

1. Use the WebSphere Administrative Console to enable properties for specific record types:
 - a. Select **Server** → **Application Servers**. The Application Servers page opens.
 - b. Click the application server name in the Name column of the Application Server collection table. The configuration panel of the application server selected appears.
 - c. In the configuration panel, under the Additional Properties section, click **Custom Properties**.
 - d. To enable SMF type 120 records, click **New**. Specify one or more of these properties:
 - name = server_SMF_server_activity_enabled = 1 (or server_SMF_server_activity_enabled = true)
 - name = server_SMF_server_interval_enabled = 1 (or true)
 - name = server_SMF_container_activity_enabled = 1 (or true)
 - name = server_SMF_container_interval_enabled = 1 (or true)
 - name = server_SMF_interval_length, value=n (where n is the interval, in seconds, that the system will use to write records for a server instance. Set this value to 0 to use the default SMF recording interval)
 - e. Click **OK** or **Apply**.
 - f. Save the changes and make sure a file synchronization is performed before you restart the servers.

2. Edit the SMFPRMxx parmlib member and update the SYS or SUBSYS(STC,...) statement to include the type 120 record. Example 22-4 shows a sample SMFPRMxx member that creates interval records every 2 minutes and records the following SMF record types:
 - 30: Address space
 - 70 to 79: RMF
 - 82: Crypto
 - 88 to 90: System logger, usage, and system data
 - 101: DB2
 - 110: CICS
 - 120: WebSphere

Example 22-4 Sample SMFPRMxx member

```

ACTIVE                                /*ACTIVE SMF RECORDING*/
DSNAME(&SYSNAME..MAN1, &SYSNAME..MAN2) /*TWO MAN DATASETS */
LISTDSN                              /* LIST DATA SET STATUS AT IPL*/
NOPROMPT                             /* DON'T PROMPT THE OPERATOR */
INTVAL(02)                            /* SMF GLOBAL RECORDING INTERVAL */
SYNCVAL(00)                           /* GLOBAL SYNC VALUE */
MAXDORM(3000)                          /* WRITE AN IDLE BUFFER AFTER 30 MIN*/
STATUS(010000)                         /* WRITE SMF STATS AFTER 1 HOUR*/
SID(&SYSNAME(1:4))                      /* USE SYSNAME AS SID */
SUBSYS(STC,EXITS(IEFU29,IEFACTRT),INTERVAL(SMF,SYNC),
TYPE(0,30,70:79,88:90,101,110,120,245))

```

To avoid collecting more SMF data than you need, review SMFPRMxx to ensure that only the minimum number of records are being collected. Use SMF 92 or 120 only for diagnostics. SMF 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 92 records are created. Unless you must have this information, turn off SMF 92 records.

You might find that running SMF 120 records in production is appropriate, because these records provide information that is specific to WebSphere applications, such as response time for J2EE artifacts and bytes transferred. If you do choose to run with SMF 120 records enabled, the authors recommend that you use the server interval SMF records and container interval SMF records rather than the server activity records and container activity records.

3. Use SET=xx to activate the SMFPRMxx member from SYSx.PARMLIB. Use the **D SMF,0** to display the parameters in effect. You must issue the SET command before you start WebSphere Application Server. If you issue the command after the application server has started, SMF 120 records will not be collected.
4. For the changes to take effect, restart the application server.
5. Use a tool such as WebSphere Studio Workload Simulator (see 22.5.1, “WebSphere Studio Workload Simulator for z/OS and OS/390” on page 300) to simulate an application stress load. While the transactions are running, switch to SDSF and RMF to observe the transactions.
6. Format the SMF recording output data set for printing to the screen or other output device:
 - a. Switch the SMF data sets by entering **i smf** from the MVS console.
 - b. Run the SMF Dump program (IFASMFDP) to create a sequential data set. A sample is shown in *z/OS MVS System Management Facilities (SMF)*, SA22-7630.
 - c. You have successfully formatted the output data set when SMFDUMP ends with return code 0.
7. To interpret the output data set see 22.4.2, “WebSphere for z/OS SMF browser”.

For an overview of SMF recording, see Chapter 1 of *z/OS MVS System Management Facilities*, SA22-7630.

After WebSphere performance data is collected, it can be monitored and analyzed with a variety of tools:

- ▶ Monitor performance with Tivoli Performance Viewer (formerly Resource Analyzer) as described in 21.8, “Tivoli Performance Viewer” on page 270. This tool is included with WebSphere.
- ▶ Use third-party vendor tools or write your own applications to exploit the Performance Monitoring Infrastructure (PMI). Search for “Developing your own monitoring applications” at the WebSphere Information Center.
- ▶ Use RMF as discussed in 22.3.2, “Analyzing RMF reports” on page 285. See “RMF Workload Activity reports and RMF Monitor III” at the WebSphere Information Center.
- ▶ Refer to “WLM Delay Monitoring” at the WebSphere Information Center.

22.4.2 WebSphere for z/OS SMF browser

The WebSphere for z/OS SMF browser is a tool for interpreting complete SMF output data sets from IFASMFDP. It writes a header line for all SMF record types and a detailed dump for SMF record 120.

The tool is a Java utility. It is executed by a JVM in the z/OS USS environment. The base tool can be downloaded from this Web site (the download requires registration):

<http://www6.software.ibm.com/dl/websphere20/zosos390-p>

Download the browser package and read the associated documentation.

You can also download *Performance Summary Report for SMF 120 records from WAS V.5 for z/OS*, PRS752, from the TechDoc Web site:

<http://www-03.ibm.com/support/techdocs/atmastr.nsf/WebIndex/PRS752>

This extended version adds a summary report that shows activity for each J2EE server instance, servlet, JSP, EJB, and method from the SMF 120 records.

This Java SMF browser is provided in the form of a JAR file named WSCSMFperfV510.jar. You need at least Java 1.3, which contains the necessary record I/O support at the SDK 142 SR2 or SDK 131 SR25 levels. To access the browser:

1. Copy the file WSCSMFperfV510.jar to your local tools directory in your z/OS system.
2. Verify that the PATH environment variable includes the correct Java library:

```
export PATH=/usr/bin/java/J1.3:$PATH
```
3. Use IFASMFDP to copy SMF records to a cataloged sequential file for offload processing. Because the tool does not interpret any SMF records other than record 120, it is recommended that you filter out all other records. Example 22-5 shows a dump of SMF records 120 from system data sets SYS1.SC48.MAN1 and SYS1.SC48.MAN2 into a sequential file named FRANCK.SC48T.SMF.

Example 22-5 Using IFASMFDP to copy SMF records into a sequential file

```
//LSA5101 JOB 999,'ITS0',  
//          MSGCLASS=T,NOTIFY=&SYSUID,CLASS=A  
//DUMP1    EXEC PGM=IFASMFDP  
//INSMF1   DD DSN=SYS1.SC48.MAN1,DISP=SHR
```

```
//INSMF2 DD DSN=SYS1.SC48.MAN2,DISP=SHR
//SMFDATA DD DSN=FRANCK.SC48T.SMF,
//          DCB=(RECFM=VBS,LRECL=32760),
//          SPACE=(CYL,(25,50)),
//          UNIT=SYSALLDA,
//          DISP=(NEW,CATLG)
//*
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
          OUTDD(SMFDATA,TYPE(120))
          INDD(INSMF1,OPTIONS(DUMP))
          INDD(INSMF2,OPTIONS(DUMP))
```

4. To interpret SMF data from our file named FRANCK.SC48T.SMF and produce a detailed report the WTSCplexSMFout.txt file, run this command (in TSO OMVS, all on one line):

```
java -cp WSCSMFperfV510.jar com.ibm.ws390.sm.smfview.Interpreter
      "FRANCK.SC48T.SMF" 1>WTSCplexSMFout.txt
```

5. To add the summary report showing the performance data, specify a second parameter (all on one line):

```
java -cp WSCSMFperfV510.jar com.ibm.ws390.sm.smfview.Interpreter
      "FRANCK.SC48T.SMF" "./WTSCplexSMFsummary.txt"
      1>WTSCplexSMFout.txt
```

The summary report of the z/OS SMF Browser will be saved in the WTSCplexSMFsummary.txt file and is available for browsing or editing through ISPF.

Note: It is implicit in the Java command parameters that your current working directory is the tools directory. If this is not the case, you receive a `NoClassDefFoundError` on `com.ibm.ws390.sm.smfview.Interpreter`. Java does not generate a diagnostic when it does not find `WSCSMFPerfV510.jar` in the current directory.

6. Enable SMF recording as described in 22.4.1, “Setting up SMF recording” on page 293, and “The SMF Dump Program” in *z/OS MVS System Management Facilities (SMF)*, SA22-7630.

Figure 22-14 on page 297 shows a sample summary report.

WSC SMF 120 Performance Summary V510 Date: Fri Dec 19 09:44:08 EST 2003 SysID: SYSB										
- record subtypes: 1:Svr_Act. 3:Svr_Int. 5:EJB_Act. 6:EJB_Int. 7:Web_Act. 8:Web_Int.										
SMF Numbr	-Record Type	Time hh:mm:ss	Server Instance	Bean/WebAppName Method/Servlet	Bytes toSvr	Bytes fromSvr	# of Calls	El.Time(mSec) Avg	WLM Max	Enclave_CPU Time(uSec) Tot/Av Max Min
1	1	1	2	3	4	5	6	7	8	9
563	120.5	9:50:03	X5SR01B	My_IVT_Application_PMI#MyIVTStatelessSession.jarMy_IVT_EJB_Name						
				remove:	1	1	1	629	629	629
				getContents:	1	0	0	129	129	129
				create:	1	144	144	11770	11770	11770
				removeItem:java.lang.String	1	0	0	143	143	143
				addItem:java.lang.String	3	1	1	170	261	120
564	120.7	9:50:03	X5SR01B	ivtejb			1707		220545	
565	120.1	9:50:03	X5SR01B	My_IVT_Application_PMI#MyIVTWebApp.war					2457	
566	120.7	9:50:03	X5SR01B		773	212				
				SimpleFileServlet			18		766	
567	120.3	9:50:03	X5SR01B	My_IVT_Application_PMI#MyIVTWebApp.war	1659	970			304718	
568	120.6	9:50:03	X5SR01B	My_IVT_Application_PMI#MyIVTStatelessSession.jarMy_IVT_EJB_Name						
				remove:	1	1	1	629	629	629
				getContents:	1	0	0	129	129	129
				create:	1	144	144	11770	11770	11770
				removeItem:java.lang.String	1	0	0	143	143	143
				addItem:java.lang.String	3	1	1	170	261	120
569	120.8	9:50:03	X5SR01B	My_IVT_Application_PMI#MyIVTWebApp.war						
				ivtejb	1	1707	1707	220545	220545	220545
				SimpleFileServlet	1	18	18	766	766	766
572	120.7	9:50:03	X5SR01B	SimpleFileServlet			18		891	
				My_IVT_Application_PMI#MyIVTWebApp.war						
==WSC=SMF=120=V510===== End of Report ===== End of Report ==WSC=SMF=V510=JMH=16May,2005										
Run Time: 3 seconds = 3067 milliseconds										

Figure 22-14 Sample summary report from SMF Browser

The detailed report file lists each activity that occurs during the collection interval for the server, Web container, and J2EE container.

The summary report file sample from an application called Trade2A is shown in Example 22-6.

Example 22-6 SMF Browser (1)

WSC SMF 120 Performance Summary2 -Date: Sun Nov 10 13:37:00 EST 2002 , SysID: SC52										
SMF Numbr	-Record Type	Time hh:mm:ss	Server Instance	Bean/WebAppName Method/Servlet	Bytes Sent	Bytes Rec'd	# of Calls	El.Time(mSec) Ave.	Max.	
1	1	1	2	3	4	5	6	7	8	9
30	120.1	13:37:00	FMISRVC		579	4191				
31	120.5	13:37:00	FMISRVC	Trade_WebApp dispatch()			1	1	1	
32	120.7	13:37:00	FMISRVC	/welcome.jsp				1		
				JSP 1.1 Processor				1		
				trade Web Application_0						
33	120.1	13:37:00	FMISRVC		863	4559				
34	120.5	13:37:00	FMISRVC	TradeRegistryBean						
				findByPrimaryKey(trade.Registr			1	1	1	
				>ejbLoad			1	0	0	
				>ejbActivate			1	0	0	
				login(java.lang.String)			1	0	0	

	>ejbStore	1	0	0
	>ejbPassivate	1	0	0
	TradeAccountBean			
	findByPrimaryKey(trade.Account	1	3	3
	>ejbLoad	1	0	0
	>ejbActivate	1	0	0
	getBalance()	1	0	0
	>ejbPassivate	1	0	0
	Trade_WebApp			
	dispatch()	1	16	16
	TradeSession			
	create()	2	0	0
	login(java.lang.String,java.la	1	1	1
	getBalance(java.lang.String)	1	3	3
35 120.7 13:37:00 FMISRVC	/tradehome.jsp		1	
	TradeAppServlet		15	
	JSP 1.1 Processor		1	
	trade Web Application_0			

This trace shows the activities in the server, J2EE container, and Web container that are caused by a login transaction in the Trade2 sample at a specific time. It includes invocation of welcome.jsp, tradehome.jsp, and TradeAppServlet by the Web container, and EJB activities such as each method invocation of TradeRegistryBean entity EJB, TradeAccountBean entity EJB, and TradeSession session EJB. Response time for each method call and the number of bytes downstream and upstream served by the server are also collected.

The summary report displays statistics, such as average and maximum elapsed time for the server, container, Web container, and J2EE container, for each type of activity during the collection interval. A type of activity can be the same JSP invocation, or the same method call on the same EJB. The following is a sample of a summary report file from an application called eLITSO, for an interval of 5 minutes.

Example 22-7 SMF Browser (2)

WSC SMF 120 Performance Summary2 -Date: Mon Nov 18 18:15:03 EST 2002 , SysID: SC50

SMF Numbr	-Record Type	Time hh:mm:ss	Server Instance	Bean/WebAppName Method/Servlet	Bytes Sent	Bytes Rec'd	# of Calls	El.Time(mSec) Ave.	Max.								
1	---	1	---	2	---	3	---	4	---	5	---	6	---	7	---	8	---
44	120.3	19:30:01	FMESRVB		34004	226469											
45	120.6	19:30:01	FMESRVB	ItemEntity													
				findByPrimaryKey(itemEntityPac			5	1553	2129								
				WebERWWNO_WebApp													
				create()			3	1	1								
				driveLoadServlet(java.lang.Str			1	1345	1345								
				dispatch()			8	3587	19361								
				WarehouseEntity													
				findByPrimaryKey(warehouseEnti			17	3428	7107								
				WebERWWJustPC_WebApp													
				create()			4	0	0								
				driveLoadServlet(java.lang.Str			2	992	1002								
				dispatch()			8	11524	49825								
				PriceChangeSession													
				create()			5	17	45								
				priceChangeSession(priceChange			5	3639	6930								

				PaySession				
				create()	15	1	2	
				paySession(paySessionPackage.P	15	18521	46859	
				DeliverySession				
				deliverySession(deliverySessio	1	46129	46129	
				create()	2	2	2	
				RemoteWebContainer				
				create()	14	0	2	
				driveLoadServlet(java.lang.Str	14	526	1361	
46	120.6	19:30:01	FMESRVB	WebERWWD_WebApp				
				create()	4	1	1	
				driveLoadServlet(java.lang.Str	2	751	1344	
				dispatch()	2	29765	59506	
				NewOrderSession				
				create()	2	0	0	
				NewOrderEntity				
				findByWIdAndDId(short,short,bo	7	6667	32528	
				WebERWWPY1_WebApp				
				create()	12	0	1	
				driveLoadServlet(java.lang.Str	9	270	1346	
				dispatch()	26	12730	82043	
47	120.8	19:30:01	FMESRVB	WebERWWOS_15				
				WebERWWSL_17				
				WebERWWPQ_16				
				WebERWWjmsPRR_25				
				eRWPriceChangeHTTPSession_26				
				WebERWWPC_21				
				DEController	1	59328	59328	
				SimpleFileServlet	1	22	22	
				JSP 1.1 Processor	1	59326	59326	
				/DEAGResults.jsp	1	54922	54922	
				WebERWWDelivery_20				
				SimpleFileServlet	8	27	51	
				WebERWWNO_19				
				SimpleFileServlet	3	49	61	
				/error.jsp	5	132	168	
				JSP 1.1 Processor	5	132	168	
				WebERWWJustPC_14				
				PAYController	8	33396	62242	
				/PAYAGResults.jsp	8	31355	57871	
				SimpleFileServlet	18	25	67	
				/error.jsp	8	680	898	
				JSP 1.1 Processor	8	33361	62177	
				WebERWWPay_24				
49	120.3	19:35:01	FMESRVB			57124	147311	

For example, we can see that the `findByPrimaryKey` method on `ItemEntity EJB` was called five times with an average elapsed time of 1553 ms and maximum elapsed time of 2129 ms. Another example is `SimpleFileServlet`, which is responsible for serving static pages in the Web application. The report shows the number of `SimpleFileServlet` calls in each Web application and the average elapsed time in the Web container.

22.5 Stress test tools

We describe load and stress tests in this section because some problems only occur when you test the applications in WebSphere for z/OS under load (stress). To avoid problems caused by bottlenecks or programming issues, we recommend that you run load and stress tests with various tools for your applications and for your WebSphere for z/OS environment before you go into production. Consider the following tools for this process:

- ▶ WebSphere Studio Workload Simulator for z/OS and OS/390
- ▶ Microsoft Web Application Stress tool

22.5.1 WebSphere Studio Workload Simulator for z/OS and OS/390

WebSphere Studio Workload Simulator is an automated test tool that simulates the numbers of Web browser users or “virtual users” and generates Web traffic to test Web applications and Web servers such as WebSphere.

WebSphere Studio Workload Simulator consists of two components:

- ▶ Controller
- ▶ Engine

The WebSphere Studio Workload Simulator Controller is installed on Windows-based hardware. It provides the control function and monitor capability for WebSphere Studio Workload Simulator.

The WebSphere Studio Workload Simulator Engine is installed on a zSeries server. It is a UNIX daemon that acts as a load generator and runs as a started task. It receives instructions from the monitor, generates the HTTP requests for the simulation, sends them to the Web server, and then returns statistics to the monitor at the end of the run.

Setting up the Workload Simulator

To set up a workload simulation with WebSphere Studio Workload Simulator, follow these steps:

1. Create or record a test script for WebSphere Studio Workload Simulator. A user can record a script using a Windows GUI for WebSphere Studio Workload Simulator. The script is series of HTTP operations that the engine on z/OS uses to run the simulation. To create a record of a test script:
 - a. Select **File** → **New** → **Capture**. The capture session starts. A pop-up window and a browser window open (Figure 22-15).

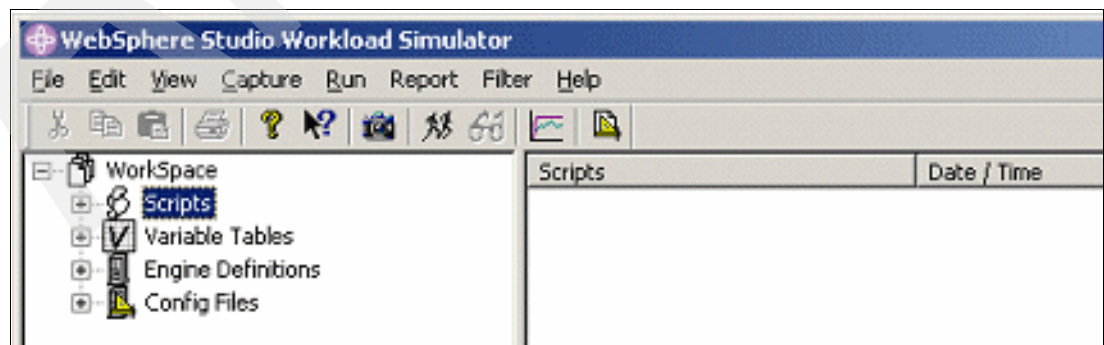


Figure 22-15 WebSphere Studio Workload Simulator window

- b. In the browser, type the URL of the Web site that is the source of the session data that you want to capture. Click **Start** in your Capture window to begin recording a script, as shown in Figure 22-16. The capture session starts and the data stream for the Web session is recorded.

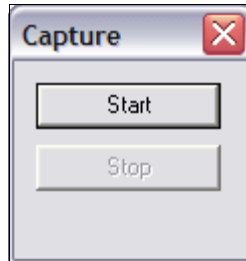


Figure 22-16 Pop-up window to start recording

- c. Click **Stop** to end the recording. When the capture session ends, WebSphere Studio Workload Simulator prompts you to enter a script name and description (Figure 22-17).

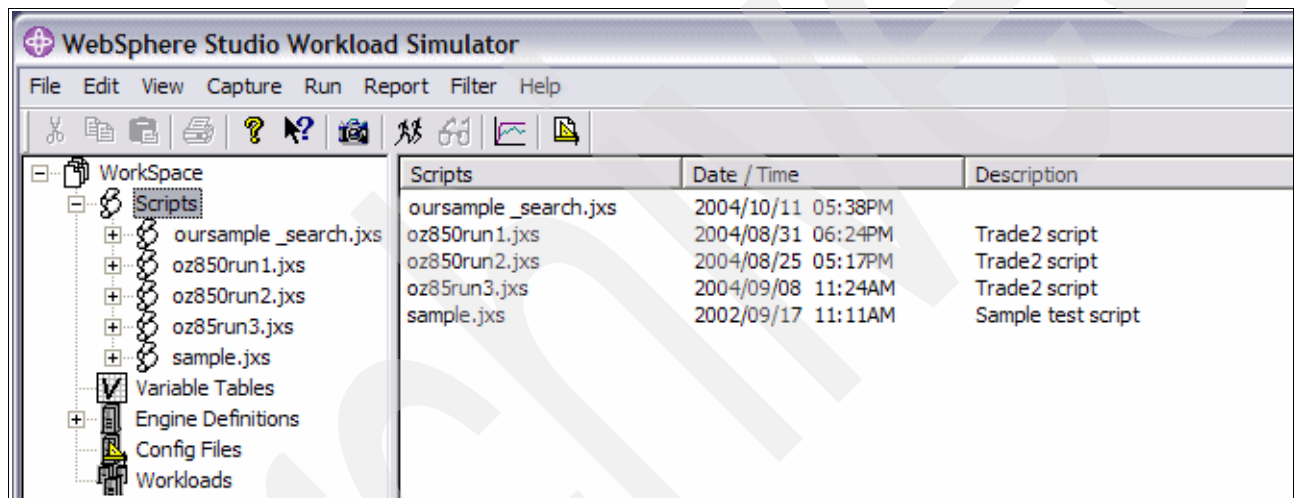


Figure 22-17 WebSphere Studio Workload Simulator with scripts of captured sessions

- d. The script shows a list of HTTP interactions (Web session elements). You can edit or change the value of these interactions (see Figure 22-18 on page 302).

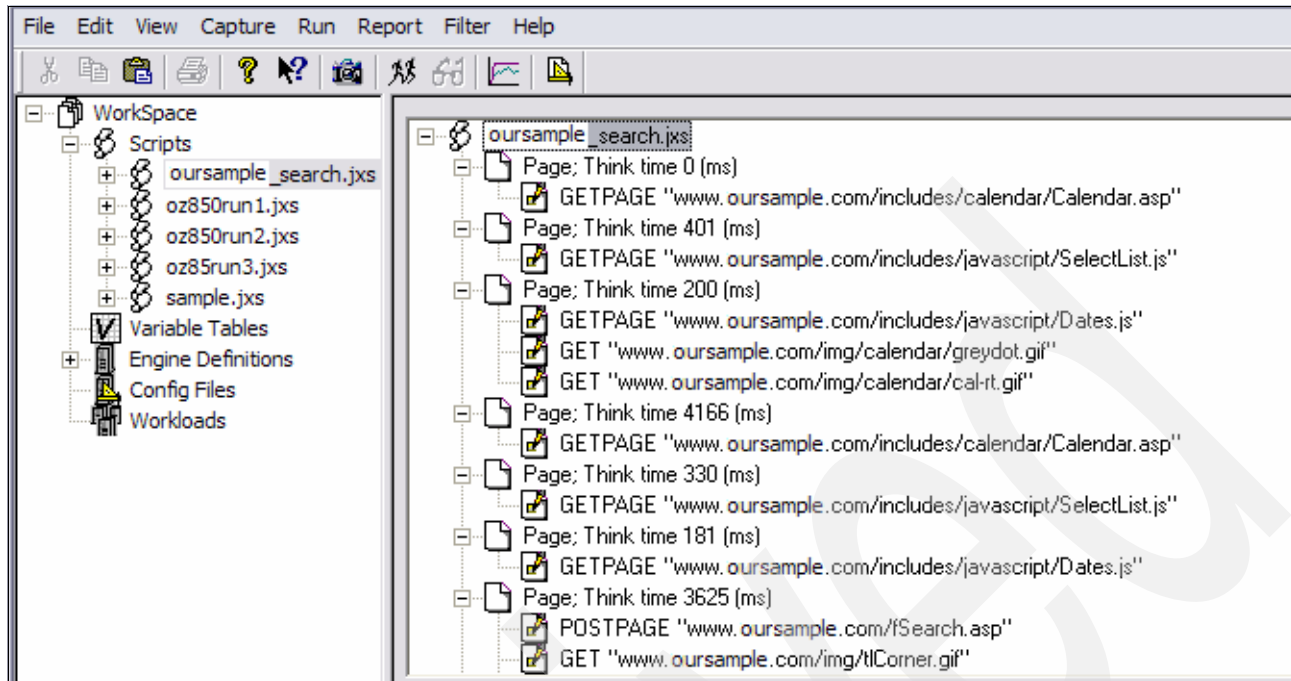


Figure 22-18 WebSphere Studio Workload Simulator window: Web session elements

- e. Variable elements in the script are revealed through a filter (Figure 22-19).

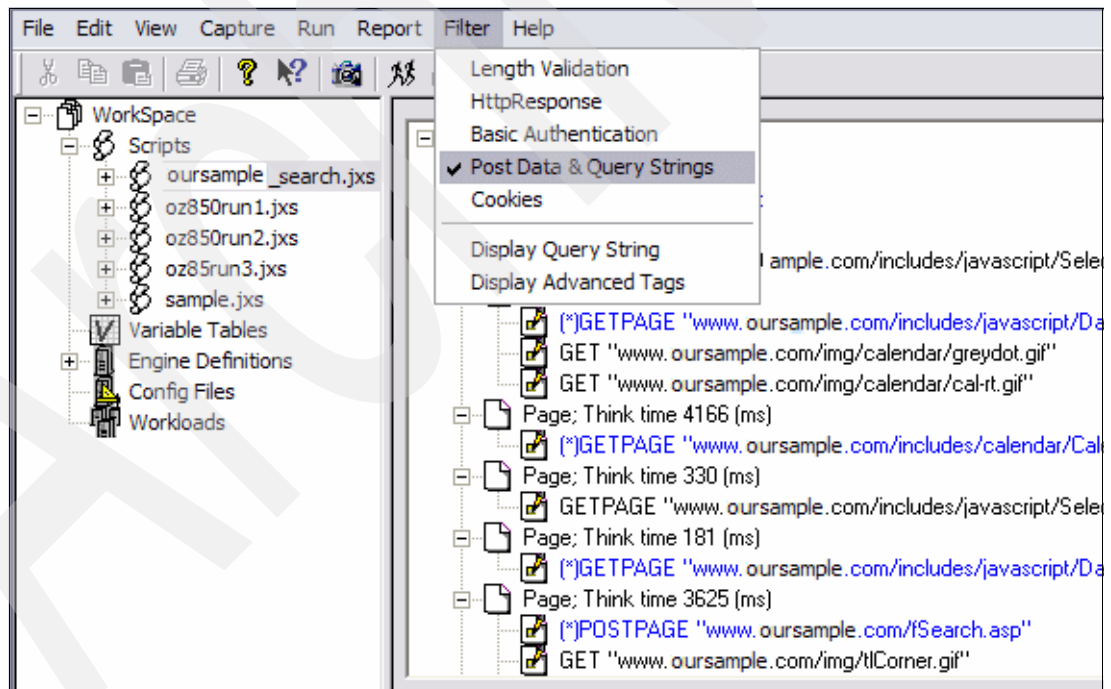


Figure 22-19 Variable elements through a filter

2. Set various runtime parameters (for example, number of clients, number of times to repeat the script, delay controls, turn dynamic cookies on or off, a time limit for the test, HTTP trace, and Socks support) before executing the script (Figure 22-20). The runtime parameters can be saved in a configuration file for reuse.

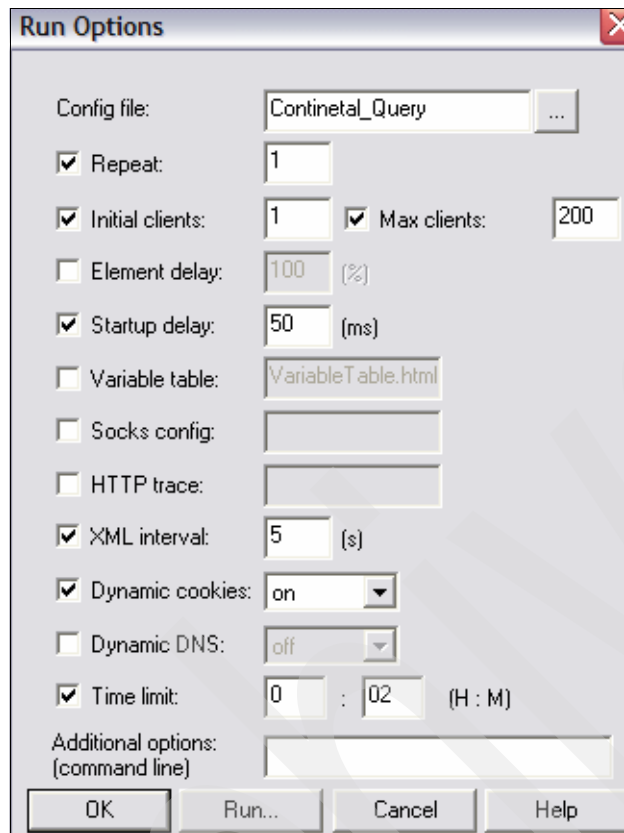


Figure 22-20 Various runtime parameters

3. Run the script and monitor the test. When the script runs, you can monitor the test engine in real time with a Windows GUI (see Figure 22-21 on page 304).

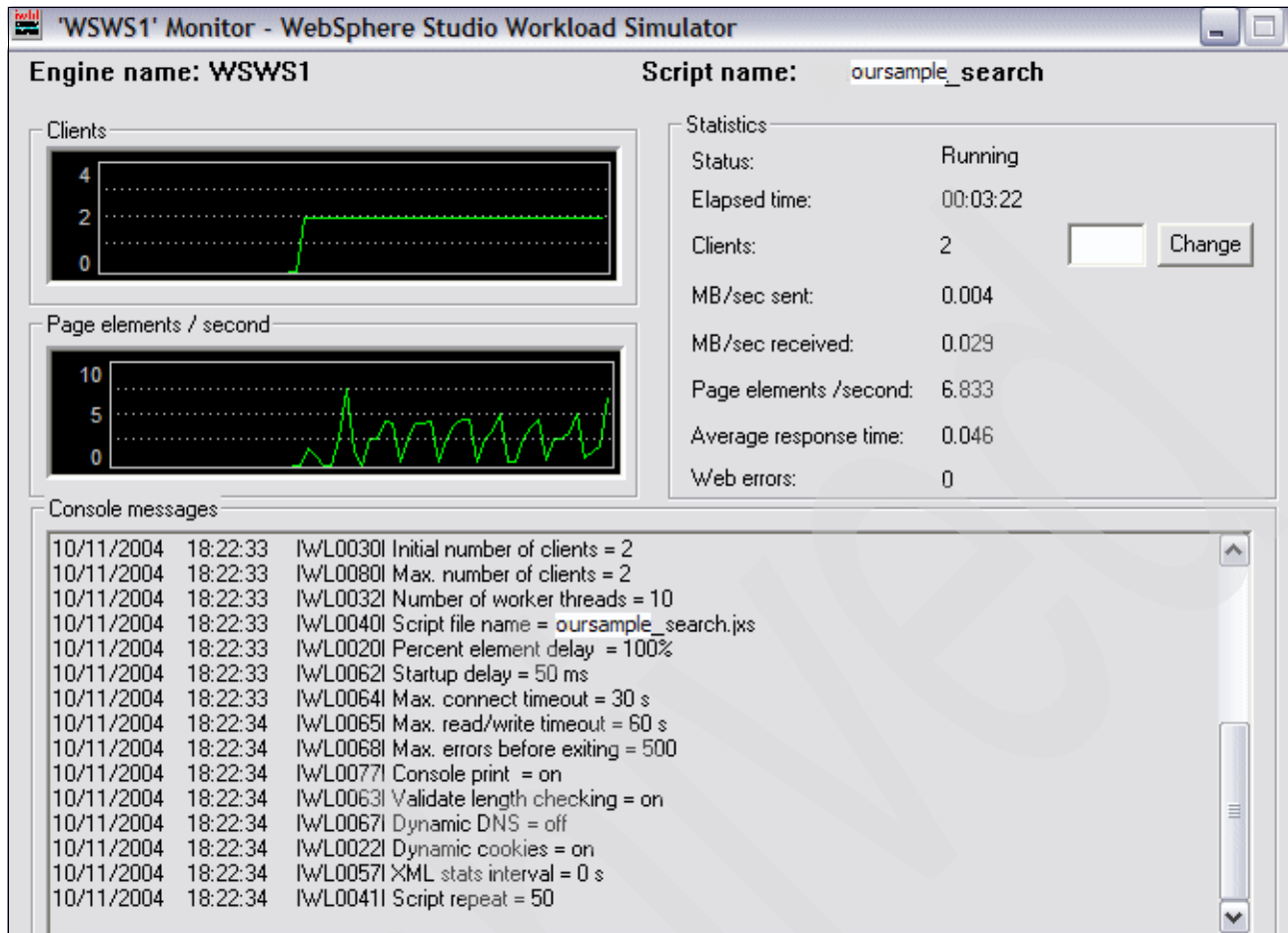


Figure 22-21 WebSphere Studio Workload Simulator Monitor GUI

Workload Simulator output and its interpretation

When the run finishes, reports are saved in the HFS on z/OS. Run statistics are recorded in the form of an XML record. These records can be used by the monitor to graph results.

There is also a log file that shows messages that are generated by the test engine during operation. This can be useful if you need to debug engine problems.

The results of the simulation are displayed as a graph for analysis (see Figure 22-22 on page 305). You can see various graphs for performance measurements while it is running, such as CPU or memory utilization, response time, data read, page elements, transactions, or written transfer (throughput).

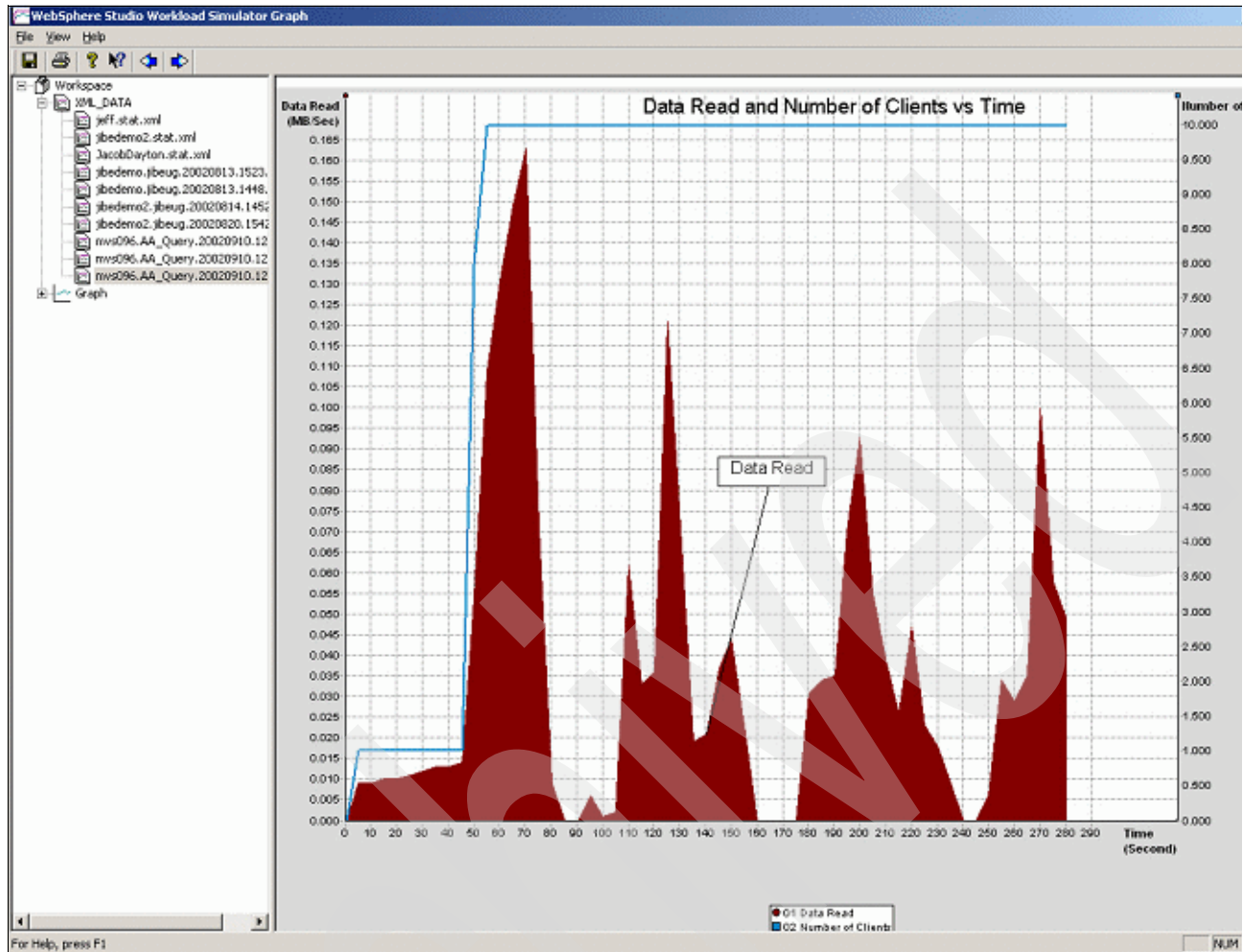


Figure 22-22 Sample simulation graph

See *WebSphere Studio Workload Simulator User's Guide*, SC31-6307, and *WebSphere Studio Workload Simulator Getting Started*, SC31-6383, on the WebSphere Studio Workload Simulator Library page for more information:

<http://www.ibm.com/software/awdtools/studioworkloadsimulator/library>

22.5.2 Microsoft Web Application Stress Tool

The Microsoft Web Application Stress Tool is designed to simulate multiple browsers requesting pages from a Web application. You can use this tool to gather performance and stability information about your Web application. It is extremely important to use this type of tool to test an application and eliminate problems prior to deploying the application in a production environment.

You can download the Microsoft Web Application Stress Tool, which is free (licensed pursuant to an End User License Agreement that is available during the setup process), from this Microsoft Web site:

<http://www.microsoft.com/technet/archive/itsolutions/intranet/downloads/webstres.mspx>

Figure 22-23 shows a screen capture¹ of the Microsoft Web Application Stress Tool in use.

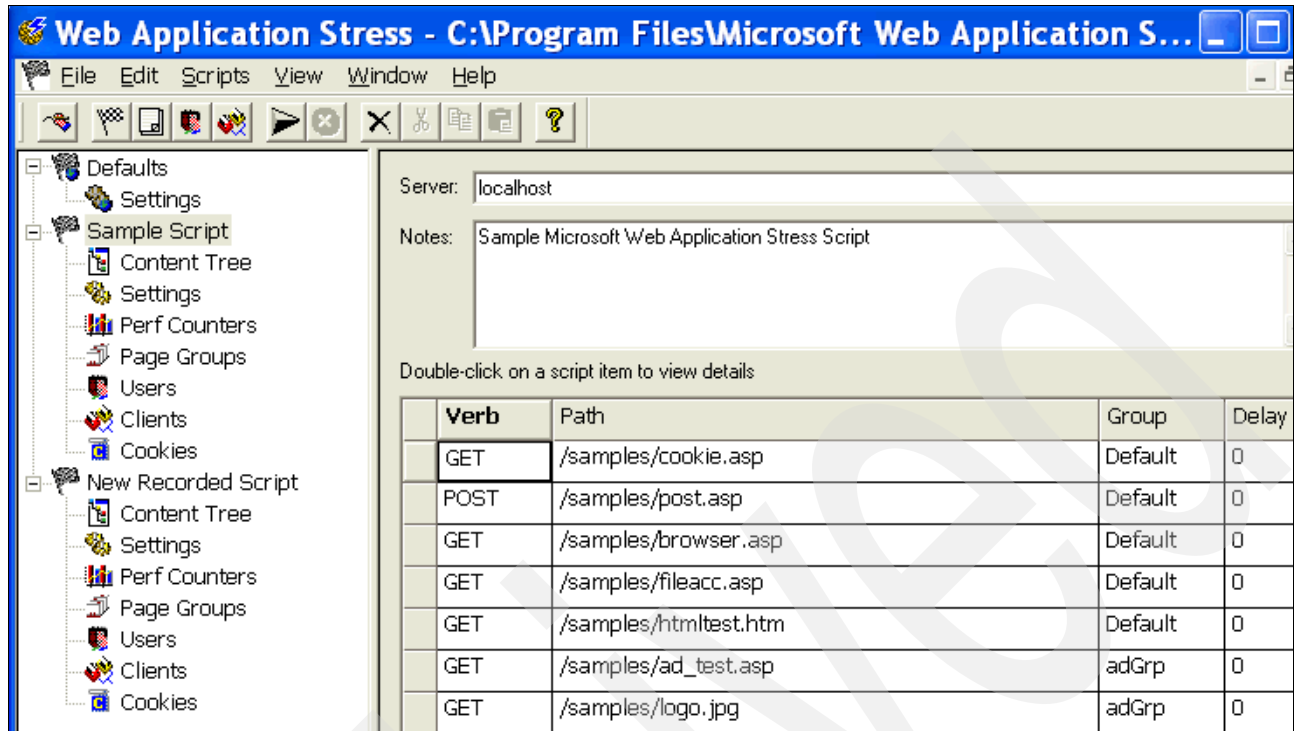


Figure 22-23 Microsoft Web Application Stress Tool

Microsoft Visual Studio® .NET Edition ships with a license for a tool called Application Center Test 1.0 that has similar functionality and that is easy to use. To learn more, visit:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/act/htm/actml_main.asp

22.6 FTP, Telnet, and editors

There are tools available on the Web that can help you in the day-to-day tasks of WebSphere for z/OS administration. Several of them are especially useful for managing Java, XML, or HTML files in the WebSphere for z/OS environment. Although not directly related to problem determination, for your convenience, we provide more details about the following tools:

- ▶ TeraTerm Pro
- ▶ WS_FTP Professional
- ▶ Directing SYSPRINT output to an HFS file
- ▶ UltraEdit

22.6.1 TeraTerm Pro

TeraTerm Pro is a free software terminal emulator (communication program) for Microsoft Windows. It provides:

- ▶ VT100 emulation
- ▶ Selected VT200/300 emulation
- ▶ TEK4010 emulation
- ▶ Kermit, XMODEM, ZMODEM, B-PLUS, and Quick-VAN file transfer protocols

¹ Microsoft product screen shot reprinted with permission from Microsoft Corporation.

You can download the application from:

<http://www.tucows.com/preview/195282.html>

Figure 22-24 shows the TeraTerm Pro emulator window.

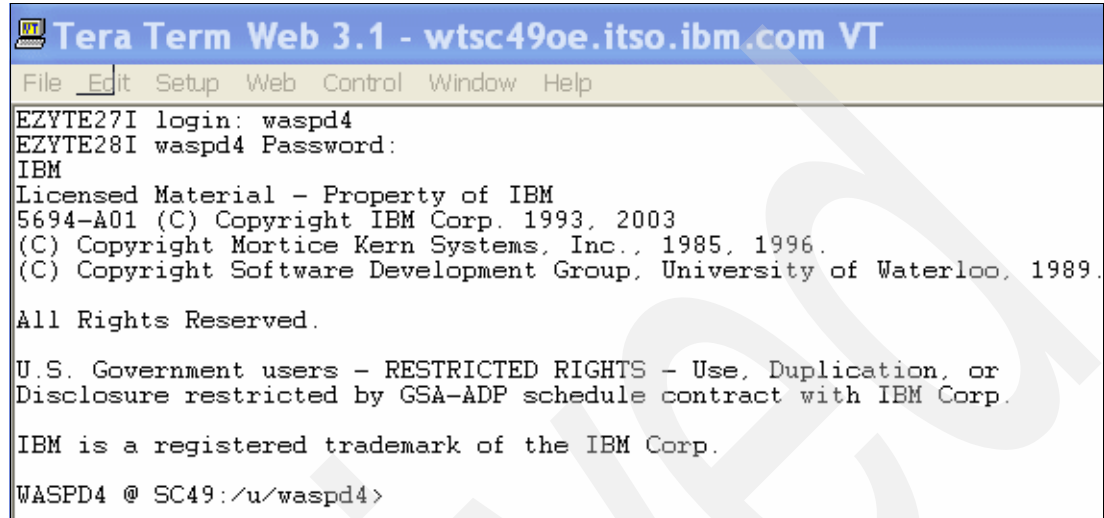


Figure 22-24 TeraTerm Pro

22.6.2 WS_FTP Professional

Ipswitch WS_FTP Professional is an FTP client program. It is a fully licensed product. You can use it to transfer files to and from z/OS easily (see Figure 22-25 on page 308).

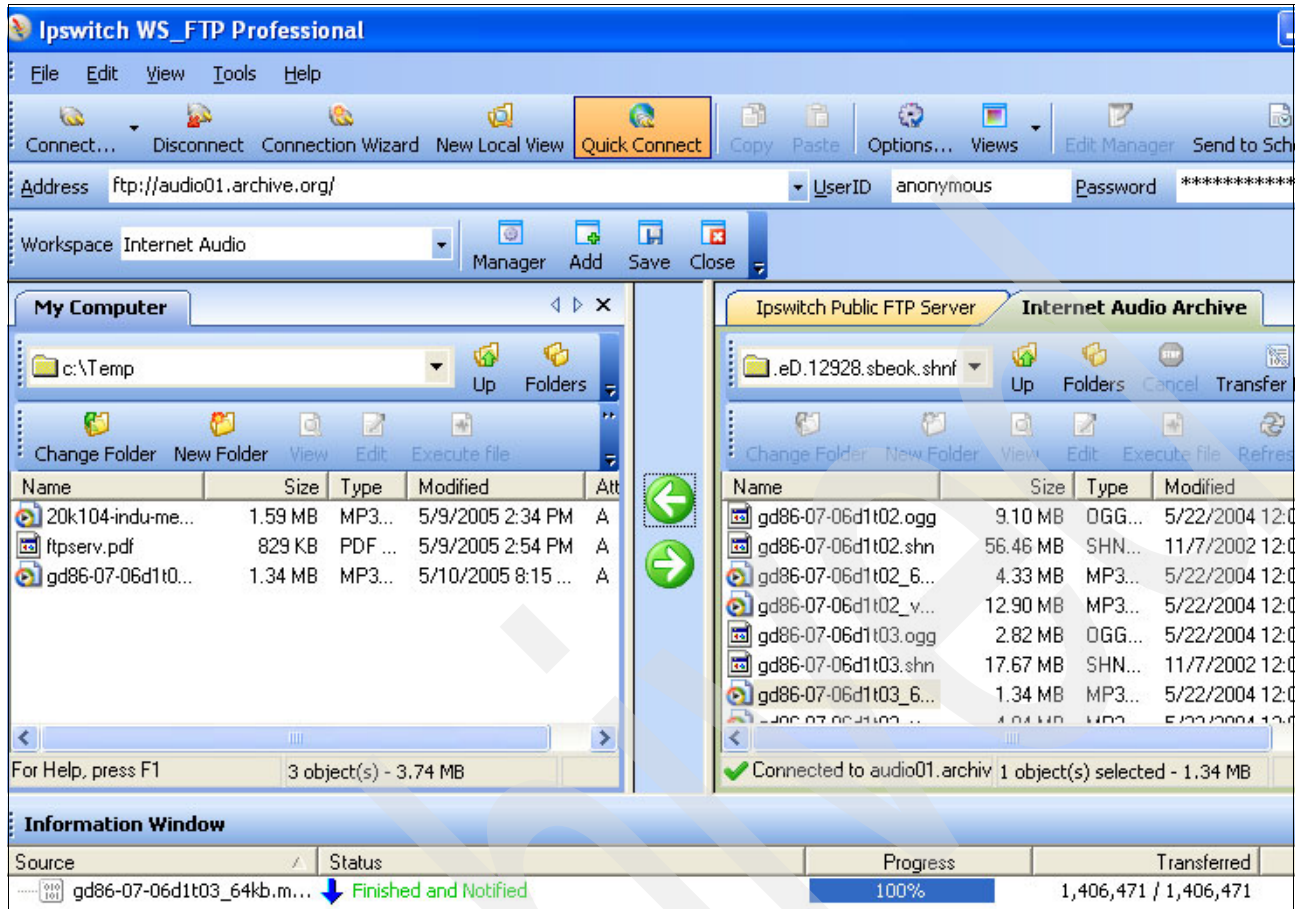


Figure 22-25 Example of WS_FTP Professional)

A free 30-day trial evaluation version is available from:

<http://www.ipswitch.com>

22.6.3 Directing SYSPRINT output to an HFS file

Many WebSphere for z/OS customers that are familiar with a UNIX or Microsoft Windows NT environment are somewhat reluctant to use SDSF to view SYSPRINT output from their application servant regions. They would much rather use a familiar editor (such as VI) in a Telnet session to view the STDOUT and STDERR information directed to SYSPRINT.

For information about how to redirect SYSPRINT to HFS files so that they can be viewed with common editors and related topics, refer to *Directing SYSPRINT Output to an HFS File in WebSphere for z/OS*, TD101087, on the IBM Techdocs Web site:

<http://www.ibm.com/support/techdocs/atsmastr.nsf/Web/TechDocs>

22.6.4 UltraEdit

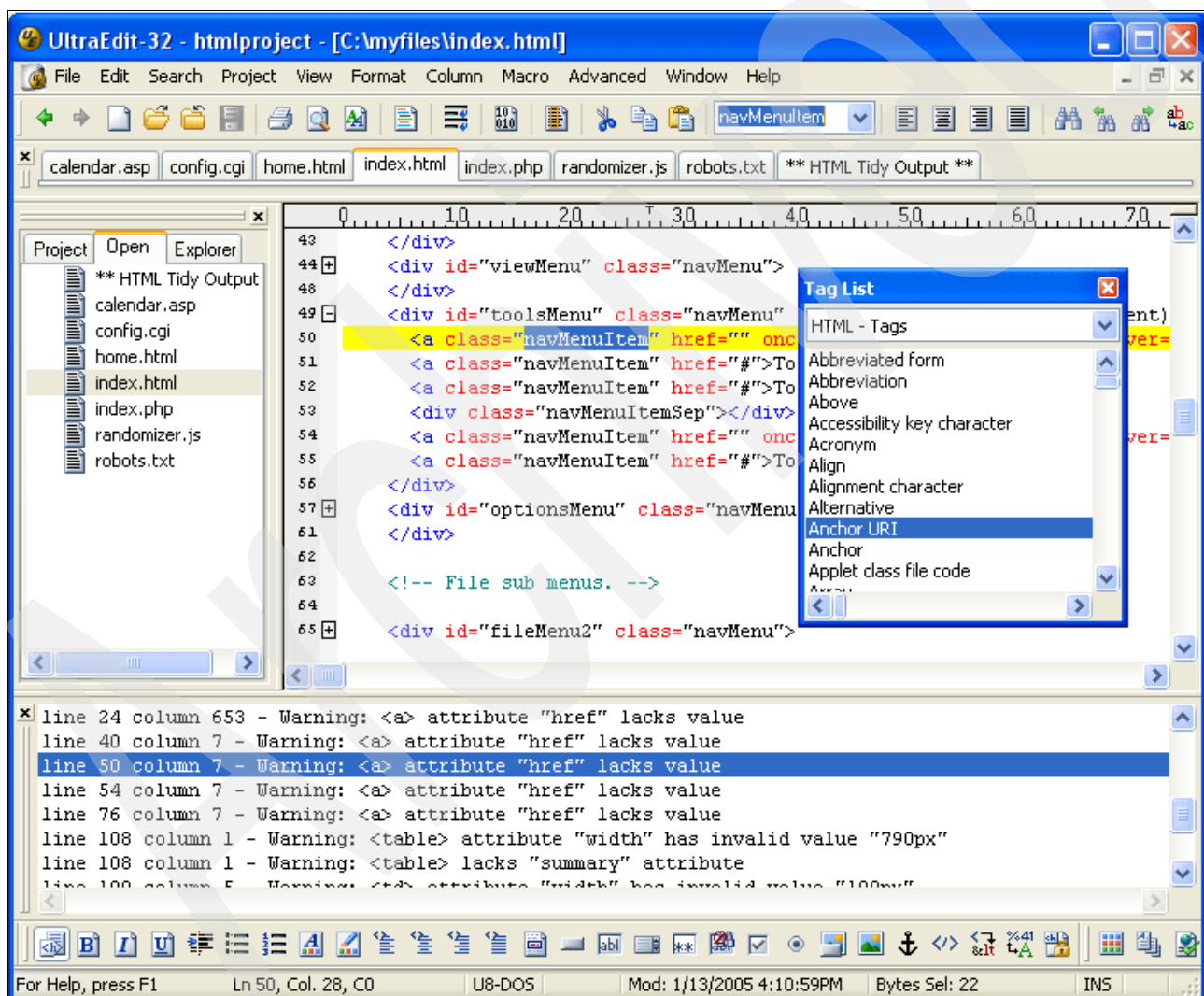
UltraEdit is a text editor, hexadecimal editor, HTML editor, and programmer editor. You can download it from the Web site for IDM Computer Solutions, Inc.:

<http://www.ultraedit.com/index.php>

Some of the more popular features of this editor are:

- ▶ You can edit files remotely using FTP. This is especially useful when working with WebSphere for z/OS. You can easily edit traces, logs, configuration files, and so on that are data sets or HFS files in z/OS.
- ▶ You can edit or compare files in binary, hexadecimal, ASCII, and so on with:
 - Easy management of the search utility (when you look for a string in a log, a window shows all the lines that contain the string)
 - User-configurable syntax highlighting specific to the language that is being edited (Java, HTML, XML, C/C++, and so on).
 - Column mode and useful macros

Figure 22-26 shows an HTML file in the UltraEdit editor.



Archived

Messages and codes

This appendix provides messages and codes for WebSphere Application Server components and subsystems of z/OS to help you analyze errors and problems.

We explain the format of WebSphere for z/OS message codes, list specific Java component messages, mention minor codes, provide WebSphere for z/OS related abend codes, and cite the most common non-WebSphere for z/OS-related message prefixes with details about where they come from.

The tables are summaries from the *WebSphere Application Server for z/OS V5.1: Messages and Codes*, GA22-7915, and the *WebSphere for z/OS Information Center* at:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

A.1 WebSphere for z/OS message codes

The prefix for WebSphere for z/OS messages is BBO. The format is BBOcnnnt. Table A-1 provides the WebSphere for z/OS message formats.

Table A-1 WebSphere for z/OS message formats

Message format	Description
BBO	Identifies it as a WebSphere for z/OS message
DYNA	Identifies it as a WebSphere for z/OS Dynamic Fragment Cache message
c	Indicates the component
nnnn	A unique identifier
t	Severity (Information, Warning, or Error)

Table A-2 gives an overview of where BBO messages come from and where they appear.

Table A-2 WebSphere for z/OS messages overview

Prefix	Come from	Appear on or in
BBOJnnnt	JVM	Operator's console error log, job log
BBOMnnnt	Runtime environment	Operator's console, error log, job log
BBOOnnnnt	Control process, servant process, daemon, CORBA. (These are general messages.)	Operator's console, error log, job log
BBOSnnnt	Security system	Operator's console, error log, job log
BBOTnnnt	Transaction service	Operator's console, error log, job log
DYNAnnnt	Dynamic Fragment Cache	Job log

To look up specific message codes, follow these steps:

1. In the Information Center navigation panel, click **WebSphere Application Server for z/OS V6** to see the table of contents.
2. Select **Reference** → **Troubleshooter** → **Messages**.
3. Choose the tab according to the first few letters in your message code.

You can also search for the specific message or code with the search function at the top of the window.

A.1.1 Specific Java component messages

Table A-3 on page 313 includes the Java component messages that are prefixed by the BBOO0222I message to provide a quick reference for your convenience. (*Mgmt* is an abbreviation for Management.)

Table A-3 BBOO0222I message components

Msg	Component	Msg	Component	Msg	Component
ACIN	Access Intent	CWSIY	Y SIBus Mediation Handlers	PMON	PMI, Tivoli Performance Viewer
ACWA	Work Area	CWSIZ	Z SIBus Mediation Framework	PMRM	Performance Monitoring Request Metrics
ADFS	Mgmt File Service Subsystem	CWSJA	A Admin	PMWC	PME Edition Support
ADMA	Application Deployment	CWSJB	B inter-bus messaging engine	PROC	Process Mgmt and Spawning Facility
ADMB	Subsystem	CWSJC	C SIBus Core SPI	PROX	Proxy
ADMC	Mgmt Connector Subsystem	CWSJD	D Admin	SCHD	Scheduler
ADMD	Mgmt Process Discovery				
ADME	Mgmt Event Subsystem	CWSJO	O SDO Repository Component	SECG	WEBUI SecurityCenter
ADMF	Mgmt Command Framework	CWSJQ	Q MFP MQ interoperability component	SECJ	Security
ADMG	Mgmt Connector Subsystem			SESN	Session and User Profiles Example
ADMK	Mgmt Utilities	CWSJR	R SIBus	SIEG	SOAP Support
ADML	Mgmt Process Launching Tool	CWSJU	U Jetstream Message Tracing	SRMC	Service Reference ManagerTransactions
ADMN	Activity Service	CWSJW	W WLM Classifier	SRVE	Transactions
ADMR	Mgmt Repository	CWSWS	S SIBus Web Services	SSLC	SSL Channel
ADMS	Mgmt Subsystem				
ADMU	Mgmt Utilities	CWUDD	Web Services UDDI Deployment & Removal	STFF	Staff Support Service
ADNT	Adaptive Entity	CWUDG	UDDI User Console	STUP	Startup Beans
APPR	Application Profile	CWUDM	UDDI Mgmt Interface	TCPC	TCP Channel
ASYN	Asynchronous Beans	CWUDN	UDDI Node Manager	TRAS	Trace Facility
BBOJ	EJB Container	CWUDQ	UDDI Migration	TUNE	Perform Auto-Tuning Support
BBOM	Naming	CWUDR	UDDI Logging and Tracing	UDAI	UDDI API
BBOO	Runtime, Web	CWUDS	UDDI SOAP Interface	UDCF	UDDI Configuration
BBOS	Security			UDDA	UDDI Data Types
BBOT	OTS and RRS			UDDM	UDDI DOM
BBZW	WBI SF Install	CWUDT	UDDI Registry Transaction Manager	UDEJ	UDDI EJB Interface
BCDS	Business Context Data Service for Event Infrastructure	CWUDU	UDDI Utility Tools	UDEX	UDDI Exceptions
BNDE	Binding EJB References	CWUDV	UDDI Value Set Tools	UDIN	UDDI Installation
CHFW	Channel Framework	CWUDX	Web Services JAXR	UDLC	UDDI Local API
CHKC	Event Infrastructure Validation	CWWCW	W Validation	UDPR	UDDI Persistence
CHKP	PME Validation	CWWDR	R Data Replication Service	UDRS	UDDI Logging
		CWWSG	G Web Service Gateway	UDSC	UDDI Security
		DCSV	DCS	UDSP	UDDI SOAP Interface
				UDUC	UDDI User Console
CHKS	SIB Validation	DSRA	Resource Adapters	UDUT	UDDI Utility Tools
CHKW	Validation	DWCT	Dynamic Workload Mgmt Client	UDUU	UDDI UUID
CHKX	XD Validation			UTLS	Utilities
CMPN	Compensation	DYNA	Dynacache	WACS	Activity Session Service
CNTR	EJB Container	EAAT	Placeholder	WACT	Activity Service
CONM	Connection Manager	ECNS	Entity Change Notification Service	WASX	Non WSCP Scripting
CSCP	CScope Service			WBIA	Support for Business Integration Adapters
CWRCB	B Core Group Bridge	ESOP	State Observer Plug-in for Event Infrastructure	WHFW	Handler Framework
CWSIA	A Service Integration Bus				

Msg	Component	Msg	Component	Msg	Component
CWSIB	B SIBus Common	GWIN	Web Services Gateway	WKSP	Work Space
CWSIC	C Communications	HMGR	HA Manager	WKSQ	Workspace Query Utilities
CWSID	D Admin	HTPC	HTTP Channel	WLTC	Transaction Monitor
CWSIE	E SIBus Externals	I18N	Internationalization Service	WMSG	Messaging Service
CWSIF	F SIBus MFP	ILMC	Instance Location Manager	WSBB	WsByteBuffer
CWSIH	H Jetstream MatchSpace	INST	Install	WSCL	WebSphere Client
CWSII	I Security	IVTL	Installation Verification Tool	WSCP	Non WSCP Scripting
CWSIJ	J Communications	J2CA	J2EE Connector	WSEC	Web Services Security
CWSIK	K SIBus Return Codes	JSAS	Security Association	WSGW	Web Services Gateway
CWSIL	L PSB	JSFG	jsf (bean class type)	WSIF	Web Services Invocation Framework
CWSIM	M SIBus Mediations	JSPG	Java Server Pages	WSSC	SOAP Channels
	SIMediationSession Interface	JSSL	ORB SSL Extensions	WSSK	Web Services Security
CWSIN	N SIBus Mediations Framework	LTXT	Localizable Text		Kerberos
	O SIBus Migration	MIGR	Release-to-Release	WSVM	Validation Manager
CWSIO	P Jetstream Message Processor	MSGs	Migration Tooling		Implementation
CWSIP		NMSV	JMS Server	WSVR	Server Runtime
		OBPL	Naming Service	WSWS	Web Services
		ODCF	ObjectPool		
CWSIQ	Q MQFap Channel	ORBx	On Demand ConFIGuration	WTRN	Transaction recovery
CWSIR	R SIBus Core	PLGC	ORB Extensions	WUDU	WebUI Deployment
CWSIS	S MessageStore	PLGN	Plug-in Configuration Generator		Descriptor Utilities
CWSIT	T TRM	PLPR	Transactions Plug-in Processor	WUPD	Update Installer
CWSIU	U Utilities		Persistence Manager	WVER	Product History Information
CWSIV	V SIBus Resource Adapter	PMGR		WWLM	WLM Client
CWSIW	W SIBus Mediations	PMI		XMEM	XMem Channel
CWSIX	X SIBus Mediations				

A.1.2 Minor codes

A minor code is shown in the WebSphere for z/OS error log and SYSPRINT. It is a hexadecimal value with the *C9C2nnnn* format, where:

C9C2 Identifies the code as WebSphere for z/OS
nnnn Uniquely identifies the code

A minor code is often associated with an exception, as shown in Example A-1.

Example: A-1 Exception with minor code

```
Trace: 2004/10/10 13:37:59.801 01 t=8BD0F0 c=UNK key=S2 (00000004)
Description: Throw CORBA system exception
exception id: CORBA::INTERNAL
minor code: c9c2110f
from filename: ./bboosyse.cpp
at line: 719
```

Some of the minor code meanings are described at the *WebSphere for z/OS Information Center*.

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp>

Important: Error (minor) codes that are not listed at the *WebSphere for z/OS Information Center* should be reported directly to IBM Support.

A.1.3 Abends

Table A-4 shows the WebSphere for z/OS related abend codes.

Table A-4 WebSphere-related abend codes

Abend code	Issuer
CC3	Daemon processing failure
DC3	Controller region processing failure
EC3	Servant region processing failure

Some reason codes are also passed along with these abend codes. They are described in detail at the Information Center; search for “abend (reason) codes.” Table A-5 shows an example with an explanation quoted directly from the Information Center.

Table A-5 Example abend code and related reason code

Abend code	Abend reason	Explanation	Suggested action
CC3	000C0009	An exception occurred on the main thread of execution, probably during initialization. The address space is abended with this code to cause the space to terminate.	Further information about the exception should be found in the job log for the space and also possibly in the error log.

If no explanation is given in the reason code, and no indication is found in any information source, the problem should be reported to IBM.

A.2 System and component message table

Table A-6 shows the most common non-WebSphere for z/OS-related message prefixes and where they come from.

Table A-6 System and component messages

Prefix	Product	Message structure
DSN	IBM DB2 UDB for z/OS	DSNcnnnt <i>c</i> : Subcomponent identifier <i>nnn</i> : Unique numeric identifier <i>t</i> : Type with I - information, A - immediate action, D - immediate decision, E - eventual action Example: DSNB209I Source: <i>DB2 UDB for z/OS Version 8 Messages and Codes</i> , GC18-7422
DB2 Information Center topic “Messages and Codes” is available at: http://publib.boulder.ibm.com/infocenter/dzichelp/index.jsp		
EZA EZB EZD EZY EZZ SNM TM	Communications Server (TCP/IP)	pppnnnt <i>ppp</i> : Prefix <i>nnn</i> : Unique identifier <i>t</i> : Type with A - immediate action, E - eventual action, D - immediate decision, I - information Example: EZZ0902I Source: <i>z/OS V1.6 Communications Server: IP Messages: Volume 1-4</i> , GC31-8783/4/5/6

Prefix	Product	Message structure
ICH	Security Server (RACF)	<p>ICHcntt</p> <p><i>c</i>: Identifies the RACF function, where:</p> <ul style="list-style-type: none"> 0: SAF initialization 3: RACROUTE REQUEST=VERIFY macro 4: RACF processing 5: RACF initialization 7: RACF status 8: RACROUTE REQUEST=AUTH macro 9: RACROUTE REQUEST=DEFINE macro <p><i>nn</i>: Message serial number</p> <p><i>t</i>: Type, where:</p> <ul style="list-style-type: none"> A - action; operator must perform a specific action. D - decision; operator must choose an alternative. E - eventual action required. I - information. W - Wait; processing stops. <p>Example: ICH500I</p> <p>Source: <i>z/OS V1R6.0 Security Server RACF Messages and Codes</i>, SA22-7686</p>
ISP ISR FLM	ISPF, PDF SCLM	<p>pppanna</p> <p><i>ppp</i>: Prefix</p> <p><i>a</i>: Alphabetic character</p> <p><i>nnn</i>: Unique identifier</p> <p>Example: ISPA001</p> <p>Source: <i>z/OS V1R6.0 ISPF Messages and Codes</i>, SC34-4815</p>
IKJ	TSO/E	<p>pppcnnnt</p> <p><i>ppp</i>: Prefix</p> <p><i>cc</i>: System module prefix (in decimal)</p> <p><i>nnn</i>: Message serial number identifying the program that issued the message</p> <p><i>t</i>:- Type, where:</p> <ul style="list-style-type: none"> A - action; the terminal user must perform the action specified in the message text. E - error; processing terminates. I - information; no action is required. <p>Example: IKJ55112E</p> <p>Source: <i>z/OS V1R6.0 TSO/E Messages</i>, SA22-7786</p>
IRX	TSO REXX processing	<p>pppcnnt</p> <p><i>ppp</i>: Prefix</p> <p><i>cc</i>: System module prefix (in decimal)</p> <p><i>nnn</i>: Message serial number identifying the program that issued the message</p> <p><i>t</i>: Type, where:</p> <ul style="list-style-type: none"> E - error; processing terminates. I - information; no action is required. <p>Example: IRX0042I</p> <p>Source: <i>z/OS V1R6.0 TSO/E Messages</i>, SA22-7786</p>

Prefix	Product	Message structure
IWM	Workload Manager (WLM)	<p>pppnnnt <i>ppp</i>: Prefix <i>nnn</i>: Message serial number <i>t</i>: Type, where: A - action by operator, D - decision by operator, E - eventual action by operator, I - information for operator/programmer, S - severe error, W - wait for operator action</p> <p>Example: IWM003I Source: <i>z/OS V1R6.0 MVS System Messages, Vol 9 (IGF-IWM)</i>, SA22-7639</p>
CEE EDC	z/OS Language Environment Runtime C/C++ Runtime	<p>pppnnnnnt <i>ppp</i>: Prefix <i>nnnn</i>: Message serial number <i>t</i>: Type, where: I - informational message, W - warning message, E - error message, S - severe error message, C - critical error message</p> <p>Example: CEE0252W Source: <i>z/OS V1R1 Language Environment Run-Time Messages</i>, SA22-7566</p>
GIM	SMP/E	<p>pppnnnnnt <i>ppp</i>: Prefix <i>nnnnn</i>: Message serial number <i>t</i>: Type, where: I - informational, W - warning, E - error, S - severe, T - terminating</p> <p>Example: GIM20101S Source: <i>z/OS V1R1 SMP/E Messages, Codes, and Diagnosis</i>, GA22-7770</p>
FDB FOM FSUM	UNIX System Services (USS) Debugger USS Shell & Utilities	<p>pppcnnnn <i>ppp</i>: Prefix <i>c</i>: Component identifier <i>nnnn</i>: Unique identifier Example: FOMC1013 Source: <i>z/OS V1R6.0 UNIX System Services Messages and Codes</i>, SA22-7807</p>
IXG	System Logger	<p>pppnnnt <i>ppp</i>: Prefix <i>nnnnn</i>: Message serial number <i>t</i>: Type, where: I - informational message, E - recoverable error, W - warning, S - serious error, T - terminating</p> <p>Example: IXG004I Source: <i>z/OS V1R6.0 MVS System Messages, Vol 10 (IXC-IZP)</i>, SA22-7640</p>
ATR	Resource Recovery Services (RRS)	<p>pppnnnt <i>ppp</i>: Prefix <i>nnnn</i>: Message serial number <i>t</i>: Type, where: I - informational message, E - recoverable error, W - warning, S - serious error, T - terminating</p> <p>Example: ATR120I Source: <i>z/OS V1R6.0 MVS System Messages, Vol 3 (ASB-BPX)</i>, SA22-7633</p>

Prefix	Product	Message structure
IMW	HTTP Server	<p>pppnnnt</p> <p><i>ppp</i>: Prefix</p> <p><i>nnnnn</i>: Message serial number</p> <p><i>t</i>: Type, where:</p> <p>I - informational message, E - recoverable error, W - warning, S - serious error</p> <p>Message ID ranges: Components:</p> <p>IMW0001-IMW2000 - IMWHTTPD IMW2000-IMW2500 - Proxy Server IMW3501-IMW3700 - CONSOLE IMW3701-IMW3999 - HTCCounter IMW4000-IMW5000 - HTIMAGE IMW5001-IMW6000 - HTADM IMW6100-IMW6900 - SSL Security</p> <p>Example: IMW0442E</p> <p>Source: <i>IBM HTTP Server Planning, Installing, and Using</i>, SC34-4826</p>
z/OS Internet Library, available at: http://www.ibm.com/servers/eserver/zseries/zos/bkserv/		

Additional material

This appendix refers to additional material that can be downloaded from the Web.

B.1 Locating the Web material

The Web material associated with this redbook is available in softcopy from the IBM Redbooks Web server at:

<ftp://www.redbooks.ibm.com/redbooks/SG246880>

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select **Additional materials** and open the directory that corresponds with the redbook form number, SG246880.

B.2 Using the Web material

The additional Web material that accompanies this redbook includes the following files:

<i>File name</i>	<i>Description</i>
SG24-6880-00.pdf	WebSphere for z/OS V4 Problem Determination
SG24-6880-01.pdf	WebSphere for z/OS V5 Problem Determination

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 323. Note that some of the documents referenced here might be available in softcopy only.

- ▶ *Monitoring WebSphere Application Performance on z/OS*, SG24-6825
- ▶ *Systems Programmer's Guide to Resource Recovery Services (RRS)*, SG24-6980
- ▶ *WebSphere Application Server for z/OS V5 and J2EE 1.3 Security Handbook*, SG24-6086
- ▶ *Installing WebSphere Studio Application Monitor V3.1*, SG24-6491
- ▶ *Effective zSeries Performance Monitoring Using Resource Measurement Facility*, SG24-6645
- ▶ *IBM Tivoli OMEGAMON XE V3.1.0 Deep Dive on z/OS*, SG24-7155
- ▶ *WebSphere for z/OS V5 JVM Dump and Heap Analysis Tools*, REDP-3950
- ▶ *Problem Determination Methodology for WebSphere on z/OS*, REDP-6001
- ▶ *Problem Symptoms in WebSphere for z/OS and Their Resolution*, REDP-6002
- ▶ *Problem Avoidance for WebSphere Application Server for z/OS*, REDP-6003
- ▶ *WebSphere for z/OS Problem Determination Means and Tools*, REDP-6880

Other publications

These publications are also relevant as further information sources:

- ▶ *WebSphere Application Server for z/OS Version 6.0.2*
 - *Program Directory*, GI11-2825
 - *Migrating, Coexisting, and Interoperating*, SA23-2207
 - *Installing Your Application Serving Environment*, GA22-7957
 - *Administering Applications and Their Environment*, GA22-7962
 - *Setting Up the Application Serving Environment*, GA22-7958
 - *Using the Administrative Clients*, SA23-2208
 - *Securing Applications and Their Environment*, SA22-7961
 - *Developing and Deploying Applications*, SA22-7959
 - *Troubleshooting and Support*, GA22-7964
- ▶ *Tuning Guide*, SA22-7963
- ▶ *z/Architecture Principles of Operation*, SA22-7832-03
- ▶ *EREP V3R5 Reference*, GC35-0152
- ▶ *EREP V3R5 User's Guide*, GC35-0151
- ▶ *z/OS V1R6.0 MVS Planning: Global Resource Serialization*, SA22-7600-03

- ▶ *z/OS V1R6.0 MVS System Codes*, SA22-7626-10
- ▶ *z/OS MVS System Commands*, SA22-7627-11
- ▶ *z/OS MVS Diagnosis: Procedures*, GA22-7587
- ▶ *z/OS V1R6.0 MVS Planning: Global Resource Serialization*, SA22-7600-03
- ▶ *z/OS V1R6.0 MVS IPCS Commands*, SA22-7594-05
- ▶ *z/OS V1R2.0 MVS IPCS User's Guide*, SA22-7596-01
- ▶ *Installing your application serving environment*, GA22-7957-03
- ▶ *z/OS V1R5 MVS System Commands*, GC28-1781
- ▶ *TCP/IP V3.2 for MVS: Users Guide*, SC13-7136
- ▶ *WebSphere Application Server for z/OS V6, Troubleshooting and support*, GA22-7964-03
- ▶ *z/OS V1R6.0 MVS Setting Up a Sysplex*, SA22-7625-10
- ▶ *z/OS V1R6.0 MVS Diagnosis: Tools and Service Aids*, GA22-7589
- ▶ *z/OS V1R6.0 MVS System Commands*, SA22-7627
- ▶ *Java 2 Technology Edition, Version 1.4.2, Diagnostics Guide*, SC34-6358-01
- ▶ *z/OS Resource Measurement Facility Report Analysis*, SC33-7991
- ▶ *z/OS Resource Measurement Facility User's Guide*, SC33-7990
- ▶ *z/OS Resource Measurement Facility Performance Management Guide*, SC33-7992
- ▶ *z/OS Planning: Workload Management*, SA22-7602
- ▶ *WebSphere Studio Workload Simulator User's Guide*, SC31-6307
- ▶ *WebSphere Studio Workload Simulator Getting Started*, SC31-6383
- ▶ *Directing SYSPRINT Output to an HFS File in WebSphere for z/OS*, TD101087
- ▶ *How can I put a local copy of the WebSphere Information Center on my workstation*, FQ102912
- ▶ *Performance Engineering & Tuning for WebSphere V5 and V6 on z/OS*, PRS804
- ▶ *Migrating from WebSphere for z/OS V5.x to V6 - An Example Migration*, WP100559
- ▶ *Disabling the Deployment Manager HTTP Timeout*, TD101703
- ▶ *Java program to test TCP/IP setup - InetInfo.java*, TD100609
- ▶ *z/OS MVS System Management Facilities*, SA22-7630
- ▶ *Performance Summary Report for SMF 120 records from WAS V.5 for z/OS*, PRS752
- ▶ *Directing SYSPRINT Output to an HFS File in WebSphere for z/OS*, TD101087

Online resources

Web sites with further information sources are mentioned in the relevant chapters.

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Archived

Archived

Index

A

- abend 5, 50, 248–249, 315
 - code 50, 52, 315
 - system abend 18
- ABEND EC3 147
- ABEND SCC3 147
- ABEND SEC3 149, 158
- abnormal end 50
- access
 - log 232, 234, 236
 - log sample 235
- AccessLog 234
- ad hoc utilities 255
- address space 282
 - active 282
- address space buffer 242
- agent log 232
- alphaWorks 33, 259
- APAR 16, 20, 142, 147
 - Authorized Program Analysis Report 16, 20
- APF 282
- API documentation 33
- APPL% 134, 137, 290
- application
 - environment 199
 - dynamic 142, 146
 - format trace data 242
- Application Center Test 306
- ARM 10
- ASCII 21, 210, 221, 225, 278, 309
- ASID 208–209
- ASTK 157
- ASVT slot usage 282
- Automatic Restart Management 143
- AVG ENC 289

B

- BBO 215, 312
- BBOC_HTTP_TRANSACTION_CLASS 123
- BBORBLOG 216–218
- binary 21, 255–256, 259, 309
- bootstrap 265
- BossLog 217
- bottlenecks 270, 300
- B-PLUS 306
- Breakpoint Properties 269
- breakpoints 266
- buffer
 - address space 242
 - core trace 255
 - exceeded 158
 - size and number 242

C

- C/C++ 309
- cache access log 232
- capacity planning 127
- CC3 52, 315
- CEEDUMP 18, 214–215, 249–250
 - parameter 250
 - view 250
- CERR 217
- CGI error log 232
- Change Log Detail levels 93, 229
- checklist
 - migration 145
- coexistence 146, 149
 - problems 147
- common storage
 - orphaned 282
 - usage 282
- communication
 - prevent 205
 - problems 202
 - program 306
 - services 202
 - with IBM 14
- Communications Server (TCP/IP) 315
- component
 - ID 19
 - trace 242, 279
- compress large files 21
- compressing data 21
- configuration
 - change root directory 148
 - error 237
 - information 281
 - message 216
 - problems 143, 147
- Configuration tab 93, 229
- connection
 - cross memory 282
 - external 203
 - ID 203
 - inbound 203
- console
 - dump 247, 255
 - log 236
- control region 214
 - abend 149
 - failure code 52, 315
 - migration 146
- controller region 214
- CPU 282
 - information 209
 - problem 16, 116
 - utilization 304
- CPU Activity Report 286

- CPU report 285
- crash 255
- CTRACE 18, 242
 - how to set it up 242
 - setup 146
 - view with IPCS 242
 - with IPCS 280
- Current Activity 270

D

- DAE 248
- daemon
 - failure code 52, 315
 - group 149
 - job log 18
 - port collision 146
 - regions 214
- DASD 282
- Data Monitoring 270
- data set
 - allocated 282
 - compress 21
 - format 21, 148
 - permission 142
 - tersed 21
 - utilization 215
- data source 10
- databases 244
- DB2 315
 - Administrator 10
 - hints 144
 - messages 315
- DC3 52, 315
- Debug 268
- debug
 - engine 269
 - hints and tips 31
 - levels 235
 - production environment 8
- Debug Perspective 266
- debugger
 - remote 254
- debugging 266
- Debugging Service 267
- Defining 15
- delay 205
- deployment
 - application 145
 - phase 156
- deployment manager 18, 163, 322
 - configuration 143
 - version 148
- developer information 32
- DeveloperWorks 16, 32
- df command output 206
- DFSMS ACS 148
- diagnose
 - lock and heap problems 259
 - system 4
- diagnostic information 261

- Diagnostic Trace Service 93, 228
- disk
 - full 207
 - space remaining 206
 - space usage 206–207
- Dispatch Time Data Effective 287
- DISPLAY
 - OMVS 209
 - sample 209
- DNS Server 204
- doc.jar 255
- du command output 207
- dump
 - analysis 258
 - CEEDUMP 249
 - CTRACE 242
 - data set 18, 143, 247
 - display report panel 243
 - Dump Analysis and Elimination 247–248
 - Dumpviewer 254, 259–260
 - information 215
 - IPCS 242
 - resize 143
 - transaction 255
 - transfer 21
 - unformatted 255
- dump utility 255, 258
 - parameters and options 256
 - shell script 257
- dumpNameSpace 265
- duplicate login configuration name 148
- DVIPA 143

E

- EAR file 157
- EBCDIC 210
- EC3 52, 315
- edit remotely by FTP 309
- edit traces, logs, configuration files 309
- editors 306
- education 36
- EJB
 - references 10
- Enable Log 93, 228
- enclave 135, 289
- enqueue request 282
- environment
 - variables 216
- error
 - log 10, 18, 146, 232, 236, 314
 - BBORBLOG 216
 - problem 215
 - sample 233
 - server 233
 - view 217
 - message 5, 17, 234, 317
 - message flow chart 42
 - native code 41
 - runtime server 216
 - state 5

- ErrorLog 233
- Ethereal 277–278
- Example 8-14 DISPLAY WLM,DYNAPPL=* 200
- exception
 - FFDC 224
 - flow chart 42
 - Java 41
 - log 220
- external writer 279

F

- FAQs 31
- Fast Response Cache Accelerator 232
- FFDC 219–220, 224–226
 - example 225
 - exception 224–226
 - ffdcRun.properties 220–222, 224–225
 - output and interpretation 221
 - set up 220
- file 93, 228
 - compare 309
- file system
 - display 206
 - full 207
 - size 206
- FindRoots 255, 258–259
- firewall 203
- First Failure Data Capture 219, 221
- flashes 31
- Foreign Socket 203
- Formatter 227
- FRCA 232
- FTP 306
 - data transfer 225
 - naming conventions 23
 - to IBM 21

G

- garbage collection 124, 258–260
- garbage collection (GC) 300
- gather background information 16
- Global Performance Management Control 286

H

- Handler 227
- hang 255, 258
- Hardware Management Console (HMC) 286
- hardware specifications 8
- heap
 - dump 258
 - HeapRoots tool 254, 259
 - information 258
 - occupancy 255
 - portable dump 258
 - usage 259
- hexadecimal 208–209, 278, 308–309
- HFS
 - directory 22

- display 206
- edit remotely 309
- environment 9
- large data set 255
- mount 144
- permission 142, 144
- plan 143
- shared 142
- Hierarchical File Structure (HFS) 142
- hints and tips 31–33
- hit rate 119
- host name 204, 276
- HR204.jar 259
- HSM request queues 282
- HTML 309
 - editor 306, 308
- HTTP 278
 - error 500 225
 - Plug-in 237
 - return code 235
 - Server 214, 236
 - logs and traces 232–233
 - message 318
- HTTPD 237
- httpd.conf 233–234, 237

I

- IBM
 - contact 23
 - Link 16
 - Support 15, 19, 226, 242
 - guide 35
- IBM HTTP Server 128, 232
- IFASMFDP 295
- Incident/Support Case 19
- InetInfo.java 276
- Information Center 15
- installation
 - plan 142
 - problems 143, 147
- interface device 280
- Internet
 - helpful pages 33
- IP address 205
 - client 234–235
 - dynamic virtual 143
 - filter 279
 - host 236
 - local 203, 276
 - name server 204
- IP header 280
- IPCS
 - for CEEDUMP 250
 - for CTRACE 242
 - format trace 279–280
 - reference 281
- ISPF 282
 - configure log 217
 - message 316
 - reference 143

J

J2EE

- migration 145
- server
 - create 10

Jakarta Commons-Logging 227

Java

- API 255
- edit 309
- heap 156, 255, 259
- messages 312
- stack 259
- turn on/off trace dynamically 201

Java garbage collection 262

Java garbage collection formatter 262

JCL 21, 144, 196, 215, 279

JDBC 244

JES

- EJES 197, 214
- JESJCL 215
- JESMSG LG 215
- JESYSMSG 215

JES2 spool 214

jformat 259

job log 18, 21, 197, 214–216, 235, 249

- information 215

JOBLIB 22

JRas 227, 242–243

JVM

- heap 254
- Monitoring Interface 254
- Profiling Interface 254
- properties 215
- JVM debug arguments 267
- JVM debug port 267
- JVM dump 255–256, 258–259
 - and heap analysis tools 254

K

Kermit 306

L

Language Environment 250, 317

large EAR file 158

LDAP 10

libsvcdump.so 255

LLA 282

LNKLST 22, 282

Local Socket 203

log 232, 237–238

- job log and system log 214–215
- LOG_STREAM_NAME 218
- LogLevel 238
- stream 216–218

Logger 227

logger 10, 216, 317

logging 227

Logging and tracing 93, 229

LookAt messages 34

loop 16, 116, 255, 258

LPA 146–147, 247, 282

LPAR 282

- cluster 288

LPARCE

- Capacity Estimator 288

M

markers 266

Master catalog 282

memory

- content 282
- delete queue 282
- leak 258
- utilization 304

Memory buffer 93, 228

memory leak 136, 262

message 34

- data 280
- prefix 311, 315
- returned 5

Microsoft Visual Studio .NET 306

Microsoft Web Application Stress Tool 306

migration 146

- checklist 145
- problems 144, 147
- Version 3.5 SE to 5.1 145
- Version 4.0.1 to 5.1 145
- Version 5.0 to 5.1 144

minor code 314

MODIFY command 196

monitor 270

monitoring 254

Monitoring and Tuning 270

mount-point name 206

MQ Administrator 10

MustGather 47

Mustgather 16

MVS

- eXtended Information 281

MXI 281–282

N

name server lookup 204

namespace 265

naming

- conventions 23, 142

native stack 259

netstat 202–203

- command 203
- sample output 203

network

- data packet 277
- hops 205
- packet
 - analyzer 277, 279
 - capturer 279
- topology 8
- transport 276

node agent
 job log 18
nslookup 202–204

O

object leak 255
OMVS 218
 command tools 206
Open Perspective 268
OutOfMemoryError 255, 259

P

packet trace 279
Packlib 21
page elements 304
Page View Rate 119
paging activity 132
Parallel Sysplex 10
partition data report 285
PC routines 282
PD/PSI 4, 8
 What PD/PSI is 4
PDF 36, 316
performance 270
 analysis 126
 configuration guidelines 128
 CTRACE 242
 expectations 126, 128, 130
 FFDC 220
 hints and tips 216, 236
 HTTP plug-in log 238
 monitoring 126
 problems 4, 20, 254
 report 295, 322
 vv trace 235
 WebSphere Studio Workload Simulator 304
Performance Viewer 270
perspective 266
Physical Management Time 287
ping 202, 205
PKZip 22
plug-in
 log 237
 trace 238
plugin-cfg.xml 237
PMR 15, 19, 23
 Investigating a PMR 19
port
 conflict 146–147
Ports 265
PrintDomTree 258–259
problem 177
 PMR 14–15, 19
 Problem Management Record 14, 19
 scenario 277
process
 information 206, 238
processing weights 288
product

 documentation 20
 information 15
 support resources 15
production environment 30, 151, 254, 305
profiling 254
programmers' editor 308
programming issues 300
proxy 232
ps command sample 208
PTF 16
PThread ID 238

Q

Quick-VAN 306

R

RACF 9, 17, 142, 316
Rational Application Developer 266
Redbooks 31
Redbooks Web site 323
 Contact us xxi
referer log 232
Release Notes 15
Remote Java Application 268
request
 header 237
Resolution Team 19–20
resource
 Recovery Services 317
 shared 225
Resource Measurement Facility (RMF) 283
Resource Selection 270
response time 5, 118, 304
 distribution 292
 expectations 136
 objective 124
RETAIN
 REmote Technical Assistance Information Network 19
REXX 281
RMF
 CPU information 285
 partition data report 286
 Post Processor 284
 reports 284
RRS 10, 196, 317
runtime
 state 50
Runtime tab 93, 229

S

SBBOSLIB 247
scenario 15, 225
SCLM 316
SDSF 21, 197, 208, 214
search 210, 309
security
 administrator 142

- servant
 - region 214
- server
 - access log 232, 234
 - error log 232–234, 236
 - log 235, 237
 - name 196
 - properties 216
 - region 18, 52, 206, 215, 315
 - trace 237
- Server Region address space classification 124
- Server Region enclave classification 122
- ServerRoot 234
- session
 - affinity 237
 - state 203
- severity level 18
- SimpleFileServlet 299
- skills 8–11, 14, 36
- SLIP command 242, 249
- SMF 294
 - Record Interpreter 295
- SMP/E
 - messages 317
 - problems 148
 - references 143
- SMS
 - classes 282
 - storage group 282
- Socket State 203
- Software
 - defect 20
 - Maintenance 20
 - Problem Report 20
 - terminal emulator 306
- spool space 242
- spreadsheet 264
- SSCHRT 289
- stack information 279
- stack trace 222, 224, 226
- Started Task 196
- state data 255
- STD DEV 290
- STEPLIB 146–147
- STORAGE 291
- storage usage 282
- stress tests 300
- subsystem 282
- summary report 285
- support
 - before contacting IBM 15
 - case 19
 - line 20
 - pages 26
- SVC 282
 - dump 250, 255, 258–259
 - in detail 247
 - interpretation 258
 - view 259
- svcdump.jar 254–255, 259

- symptoms 4–5, 8, 15
- synchronization 225–226
- SYSLOG 17, 214–216, 248
- SYSOUT 215, 217, 237
- sysplex 282
- Sysplex Distributor 143
- SYSPRINT 22, 201, 215, 242, 308, 314
 - to HFS File 306, 308
 - view 308
- system
 - administration 8
 - log 17, 197, 214–215
 - output 215
 - programming skills 9
 - programming staff 255
 - trace 255
- System.out.println 227

T

- tape unit 282
- tar command 22
- TCB 209
- TCP header 280
- TCP/IP
 - checkout program 276
 - commands 10, 202, 281
 - component trace 279
 - CTRACE 242
 - DVIPA 143
 - messages 315
 - packet trace 276–277, 279, 281
 - reference 281
 - references 143
 - skills 9–10
 - stack 279
 - test 277, 322
 - tools 202, 281
- Techdocs 31, 277, 308
- Technotes 16
- TEK4010 emulation 306
- telnet 306
- TeraTerm Pro 306–307
- test
 - environment 6
- testing 266
- text editor 308
- Think Time 119
- thread
 - analysis 258
 - display command 208–209
- threads
 - servant region 208
- throughput 118
- timeout 158
- Tivoli
 - Performance Viewer 202
- Tivoli Performance Viewer 202, 270
- token 43
- trace 235, 261
 - back 215, 249

- buffer 255
- commands 201, 205
- component 18
- data 279–280
- exchange 21
- format 279–280
- JVM 215
- MODIFY 201
- request 196
- skills 10
- start and stop 236, 242, 279
- TCP/IP 276
- to data set 242
- variables 215
- Trace Analyzer for WebSphere 261
- tracert 202, 205
- tracing 227
- transaction 118
 - class 123
 - diagnose production environment 255
 - dump 255, 258–259
 - throughput 254, 304
- transfer files 211
- Troubleshooting 93, 229
- TRSMAN 21, 23
- TSO Command panel 202, 204–205
- TSO REXX processing 316
- TSO/E 205, 316
- tuning 4, 31, 313

U

- UltraEdit 306, 308–309
- unexpected condition 227
- UNIX
 - System Services 9, 22, 206, 317
 - threads 208
- URI 239
- URL
 - error log 234
 - URI matching 237
- user
 - catalog 282

- USS
 - and OMVS command tools 206, 300
 - CTTRACE 242
 - messages 317
 - problems 207
 - references 143
 - skills 10

V

- Verbose garbage collection 263
- verbose GC trace 136
- version 148
- virtual host
 - match 237, 239
- VT100 emulation 306
- VT200/300 emulation 306
- vv trace 232, 235–237

W

- Wait 316–317
- warning message 317
- WASgrep.sh 206, 210
- Web
 - Application Stress tool 305
 - helpful pages 33
 - server plug-in 232
 - traffic 300
- WebSphere
 - commands 196
 - plug-in 237
 - Proof of Concept for z/OS 30, 151
 - Studio
 - Workload Simulator 300, 305
 - Studio Workload Simulator 127
 - support structure 14
- WebSphere Studio Workload Simulator 300, 304
- whitepapers 16, 31, 33
- WinPcap 278
- Winzip 22
- WLM 10, 282
 - commands 199
 - dynamic 142
 - information 209
 - maximum number of instances 121
 - messages 317
 - minimum number of instances 121
 - references 196
 - static 142
- WLM queues 122
- workload
 - Simulator for z/OS 300
- Workload Activity Report 137, 289
- worksheet 143
- wrong output 5
- WS_FTP 306–308
- WSAdmin 10
- wsdeploy tool 157
- wsjaas.con 148

X

- XML 306, 309
 - parser 259
 - problems 259
 - references 259
 - tools 206, 210, 304
- XMODEM 306
- XSL 266

Z

- z/OS external writer 279
- z/OS Internet library 36
- z/OS packet trace facility 276
- ZIP file 22
- ZMODEM 306

Archived



Problem Determination for WebSphere for z/OS

(0.5" spine)
0.475" <-> 0.873"
250 <-> 459 pages



Problem Determination for WebSphere for z/OS



Problem determination methodology

Problem symptoms and their solutions

Means and tools to support problem determination

IBM WebSphere Application Server for z/OS V6 is a complex product made up of many components. This IBM Redbook focuses on the problems that you can experience with WebSphere for z/OS. It is intended for system programmers and administrators who need to identify, analyze, and fix problems efficiently so that they can deliver good support for the WebSphere environment.

In Part 1, we provide an overview of problem determination methodology, what skills you need, where to find information about related topics, and how to communicate with IBM when a problem occurs.

In Part 2, we describe the most common problem symptoms. Flow charts guide you through the problem analysis process step by step. Individual tasks and questions help you filter out irrelevant facts and find the problem area, so that you can identify the type, source, cause, and possibly a solution.

In Part 3, we identify possible problem areas and arrange them into four phases that correspond with WebSphere for z/OS life cycle stages. We explain how to analyze the problems and provide valuable hints and tips for avoiding them.

In Part 4, we provide means and tools for problem determination such as commands, logs, dumps, traces, and diagnostic tools. We describe other tools that can ease the day-to-day tasks and prevent problems. We also explain where to get these tools, show you how to use them, and provide examples.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks