IBM

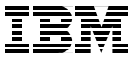
z/OS for e-business: An Introduction to System Tuning

- Set up the z/OS infrastructure for e-business and Java applications
- Tune the HTTP Server
- Tune the WebSphere Application Server

Franck Injey
Sebastien Llaurency
Patrick C. Ryan
Ronald Townsend
Tyrrell G. Ulerie

Redbooks

ibm.com/redbooks



International Technical Support Organization

z/OS for e-business: An Introduction to System Tuning

July 2002

SG24-6542-00

Take Note! Before using this information and the product it supports, be sure to read the general information in “Notices” on page ix.

First Edition (July 2002)

This edition applies to WebSphere Application Server V4.0.1 for z/OS and OS/390 at Service Level L00PTF11, program number 5655-F31 for use with z/OS Version 1 Release 1 or OS/390 Version 2 Release 10.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

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

Contents

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 illustrates 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. 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 IBM's application programming interfaces.

Trademarks

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

Redbooks(logo)™ 	MVS™	SecureWay®
AIX®	OS/390®	SP™
CICS®	PAL®	Tivoli®
CICS/ESA®	Parallel Sysplex®	VTAM®
DB2®	Perform™	WebSphere®
DFS™	RACF®	z/OS™
IBM®	Redbooks™	zSeries™
IMS™	RMF™	3090™
Language Environment®	S/390®	

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

Lotus®	Approach®	Notes®

The following terms are trademarks of other companies:

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

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

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

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

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

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

Preface

This IBM Redbook provides a high-level overview of what is needed to set up a z/OS configuration in an e-business environment. It consolidates some of the tuning information for the WebSphere Application Server and z/OS operating system environments in one place.

In addition, since system environments change over time and you need to proactively prevent performance problems in today's e-business world, it covers aspects of monitoring your system's performance.

This publication represents an introductory work on this subject, so to guide your further understanding we reference other IBM Redbooks, Web sites, and IBM documentation that contain more detailed pieces of the overall picture of what is needed to successfully set up and tune the WebSphere Application Server and supporting z/OS operating system.

The team that wrote this redbook

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

Franck Injey is a Project Leader at the International Technical Support Organization, Poughkeepsie. He has 25 years of experience working on S/390 hardware and system performance. Before joining the ITSO, Franck was a Consulting IT Architect in France.

Sebastien Llaurency is a Java developer and WebSphere specialist in France at the Design Center for e-Transaction Processing. He has two years of experience in prototype development for EMEA customers. His areas of expertise include Java development tools and WebSphere Application Server.

Patrick C. Ryan is a Senior OS/390 e-business Consultant working for the IBM e-business Center for Technical Competency (eCTC). He has over 28 years of experience in the IT industry. In his current position, he has worked with major IBM customers, installing and configuring WebSphere, both in the US and Europe. He has co-authored several Redbooks about WebSphere on the OS/390 and z/OS platforms. In the past he has worked as an MVS Programming Support

Representative, VTAM level 2/3 Specialist, VM designer, LAN Administrator and IT Specialist for the S/390 and z/OS division. He has also presented at Share and Guide meetings and demonstrated new technology at numerous CIO and technical conferences.

Ronald Townsend is an IBM IT Specialist working for the IBM zSeries e-business Services team. He has 20 years of experience in S/390 applications and systems development. He holds a Bachelors degree in Computer Science from Purdue University.

Tyrrell G. Ulerie is a Senior e-business Consultant working for the IBM zSeries e-business Services team. He has experience in WebSphere, Linux and implementation of the IBM Common Connection Framework involving z/OS and AIX systems. He holds a degree in Computer and Systems Engineering from Rensselaer Polytechnic Institute (RPI).

Thanks to the following people for their contributions to this project:

Rich Conway, Tamas Vilaghy, Holger Wunderlich	International Technical Support Organization, Poughkeepsie Center
Jim McCoy	Washington Systems center
Gary Puchkoff	IBM Poughkeepsie, WebSphere for z/OS development
Bob St John	IBM Poughkeepsie, zSeries performance

Notice

This publication is intended to help System Performance Analysts and System Administrators understand performance and tuning aspects when running WebSphere Application Server V4.0.1 for z/OS and OS/390, 5655-F31 on z/OS or OS/390. The information in this publication is not intended as the specification of any programming interfaces that are provided by WebSphere Application Server. See the PUBLICATIONS section of the IBM Programming Announcement for IBM WebSphere Application Server V4.0, 5655-F31, z/OS, 5694-A01 or OS/390, 5647-A01, for more information about what publications are considered to be product documentation.

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 Internet note to:

  redbook@us.ibm.com

- Mail your comments to the address on page ii.

Part 1

Overview

In this part we provide an overview of the WebSphere Application Server environment and the supporting z/OS operating system. We also talk about monitoring performance.

1

The z/OS environment

The audience for this book should have a basic understanding of the z/OS operating system and some knowledge of the WebSphere Application Server environment.

The intent of this redbook is to provide setup and tuning information and not an in-depth view of the components involved. For a high-level overview of the WebSphere Application Server environment, refer to *e-business Cookbook for z/OS Volume I: Technology Introduction,* SG24-5664. For a more detailed view of the installation and configuration of the WebSphere Application Server, refer to *e-business Cookbook for z/OS Volume II,* SG24-5981.

The overall e-business on z/OS environment consists of a number of components both in the z/OS operating system and the WebSphere Application Server. The z/OS operating system provides the supporting infrastructure, consisting of such things as directory services (LDAP), persistent storage (DB2), security (RACF), Workload Manager (WLM) and other subsystems. The performance of these subsystems can impact the performance of the WebSphere Application Server environment.

While this redbook talks about setting up and tuning the supporting infrastructure for Java on z/OS, it does not discuss the design and coding of efficient Java application programming for servlets or EJBs. For information on these subjects see our companion redbook *Writing Optimized Java Applications for z/OS*, SG24-6541.

The following overview briefly discusses the various components of both the z/OS operating system and the WebSphere Application Server that can be tuned to better the performance of the e-business environment.

Note: This redbook does not address performance or tuning related to CICS or IMS connectors.

1.1 Overview

One of the strengths of running WebSphere Application Server on a zSeries server is the z/OS environment itself. The z/OS operating system allows you to scale your Web server both horizontally and vertically. Using the z/OS Workload Manager (WLM), additional WebSphere Application servers can be started to meet a sudden increase in demand.

The z/OS SecureWay Server also provides a robust and industrial-strength security environment that protects your e-business and other corporate assets from being compromised.

Because they provide functions that are used by WebSphere, a number of components of the z/OS operating system environment can be tuned to both decrease response time and increase the overall throughput of Web-enabled applications:

- TCP/IP
- z/OS UNIX System Services
- HFS
- LDAP/DB2
- WLM
- SSL
- LOGGER/RRS
- JVM

We discuss these components in the following sections.

1.1.1 TCP/IP

The TCP/IP protocol is used as the basis for communication between the clients and the HTTP server. It is also used to communicate between the System Management User Interface and System Management Server and other portions of the WebSphere Application Server environment.

1.1.2 UNIX System Services

UNIX System Services (USS) provides the overall environment needed to run the WebSphere Application Server environment and the Java applications that run within it. The tight integration of the UNIX platform into the z/OS operating system creates a stable, secure foundation for an e-business, and allows you to exploit the portability of Java applications. You can run both your Web server and your backend applications on the same physical hardware, and eliminate the latency and overhead caused by interplatform communication.

1.1.3 Hierarchical File System

The Hierarchical File System (HFS) is used to store the Java code, the static files and the configuration files needed for running the application servers. It provides a means of separating applications and their related files into a logical grouping of files and directories.

1.1.4 LDAP

LDAP Directory services are used to a provide a method for finding Java objects.These objects such as servlets and Enterprise Java Beans (EJBS) form the basis of e-business reusability on the Web.

WebSphere Application Server V4.01 uses an LDAP server as a naming registry. The data that is used by LDAP is stored in DB2 tables. Both the naming server and the IR server use LDAP DLLs to directly store data into DB2.

For more about the SMS and IR servers, see 1.2, “WebSphere Application Server environment” on page 6.

The applications and associated files are actually stored in DB2 tables managed by the System Management server. The SM server talks directly to DB2.

The actual J2EE Application Servers use the Java Naming Directory Interface (JNDI) to LDAP to access the Java objects.

1.1.5 Workload Manager

The Workload Manager (WLM) is used to automatically scale the number of available servers and balance the workload between those servers. This functionality provided by the z/OS operating system provides increased flexibility for WebSphere when running on the zSeries platform.

1.1.6 SSL and security

SSL and security are needed in today's Web-enabled environment, and IBM zSeries and z/OS work together to lessen the impact of providing security by using Hardware Cryptographic Co-processors and the z/OS SecureWay Server to provide security with a minimum of overhead.

1.1.7 LOGGER/RRS

WebSphere for z/OS requires the use of the RRS Attach Facility (RRSAF) of DB2, which in turn requires that resource recovery services (RRS) be set up. RRS provides services to authorized resource managers, such as database programs and communications managers that manage distributed transactional communications.

Registration services, context services, and resource recovery services (RRS) are three separate z/OS components, but it is sometimes useful to think of them as a single function called recoverable resource management services (RRMS), the z/OS syncpoint manager.

For more information on system level resource recovery in z/OS, refer to *z/OS V1R2.0 MVS Programming: Resource Recovery,* SA22-7616.

1.1.8 JVM

The JVM exists in each of the server address spaces that is used to run the Java applications. Staying at the latest level maintenance is important because constant improvements in performance are being made within each release level. Another important factor is how Java memory is defined and the effects on garbage collection.The JVM has thresholds it uses to manage the JVM's storage. When the thresholds are reached, garbage collection (GC) gets invoked to free up unused storage. GC has a significant effect on Java performance.

1.2 WebSphere Application Server environment

Aside from the operating system components, there are also a number of components and subcomponents of the WebSphere Application Server environment.

It may help to visualize the WebSphere Application Server environment as consisting of the following three main components and several subcomponents:

- Basic environment
  - HTTP Server/WebSphere Application Server Plugin
  - Daemon Server
  - System Management Servers
  - Naming Servers
  - Interface Repository Servers
- Application Servers
  - J2EE Application Servers with the HTTP/HTTPS transport handler option
  - MOFW Application Servers
- System Management User Interface (SMUI)

For more information about these components, refer to *e-business Cookbook for z/OS Volume I: Technology Introduction,* SG24-5664.

1.2.1 The Basic Environment

A full WebSphere for z/OS run-time environment includes several components, as shown in Figure 1-1:

- HTTP Server and WebSphere Application Server plugin
- Daemon Server
- System Management Server
- Naming Server
- Interface Repository Server

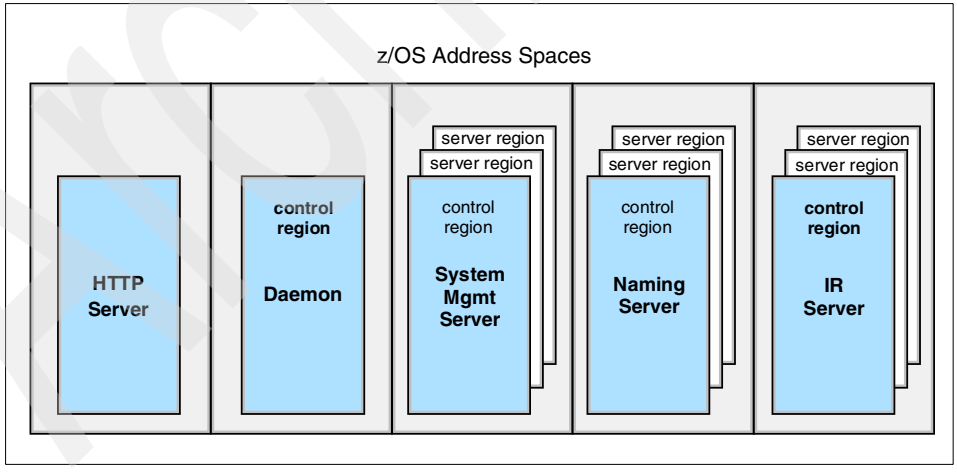


Figure 1-1 WebSphere Application Server environment

The Daemon Server consists of a single address space or Control Region, and is used to start the System Management Server (SMS), the Naming Server (Naming), and the (IR) Interface Repository Control Region servers.

The SMS, Naming and IR Server each consist of a Control Region and one or more Server Regions. Each of the Server Regions runs in its own z/OS address space.

Both the starting of, and the number of, individual Server Regions is managed by the Workload Manager (WLM); see 3.4, “Set up Workload Manager for WebSphere” on page 58 for more information regarding WLM.

This key connection, allowing the z/OS operating system Workload Manager to control the starting of additional servers, allows for a dynamic horizontal expansion of the WebSphere Application Server environment to respond to the asymmetrical needs of an e-business. This functionality is critical in today’s fast-paced environment.

HTTP Server and WebSphere Application Server plugin

The HTTP Server provides a method for serving static pages, CGI applications and (via the WebSphere Application Server plugin) Servlets and JSPs. The HTTP Server is usually started via a procedure in SYSx.PROCLIB and initializes a Plugin interface to the WebSphere Application Server.

On our system the HTTP Server procedure is named IMWEBSRV.

Daemon Server

The Daemon is also started via a procedure (in SYSx.PROCLIB) and brings up the WebSphere Application Server environment. Once the Daemon is started, it will start up the System Management, Naming and Interface Repository Control Region servers. The Daemon Server consists of a single instance (Control Region) within a z/OS address space.

On our system the Daemon Server startup procedure is named BBODMN.

Systems Management Server

The System Management Server (SMS) is used to communicate with the System Management User Interface running on a client machine. The purpose of the SM Server is to allow users to add new applications and control the operation of the application server environment. The SM server is also required to be present for all the other servers to obtain their configuration data.

The Systems Management Server consists of a Control Region and multiple Server Regions, each a separate address space, which can be started by Workload Manager as needed.

On our system the System Management Server Control Region startup procedure is named BBOSMS and the Server Region startup procedure is named BBOSMSS.

Naming Server

The Naming Server is used to locate CORBA objects. Like the SM Server, the Naming Server consists of a control region and multiple server regions.

Interface Repository Server

The Interface Repository Server is only used with CORBA applications. The Interface Repository Server also consists of a Control Region and multiple Server Regions.

1.2.2 Application Servers

In addition to the Basic Environment, there can be one or more of the following application servers, as shown in Figure 1-2:

- J2EE Application Server
- MOFW Server

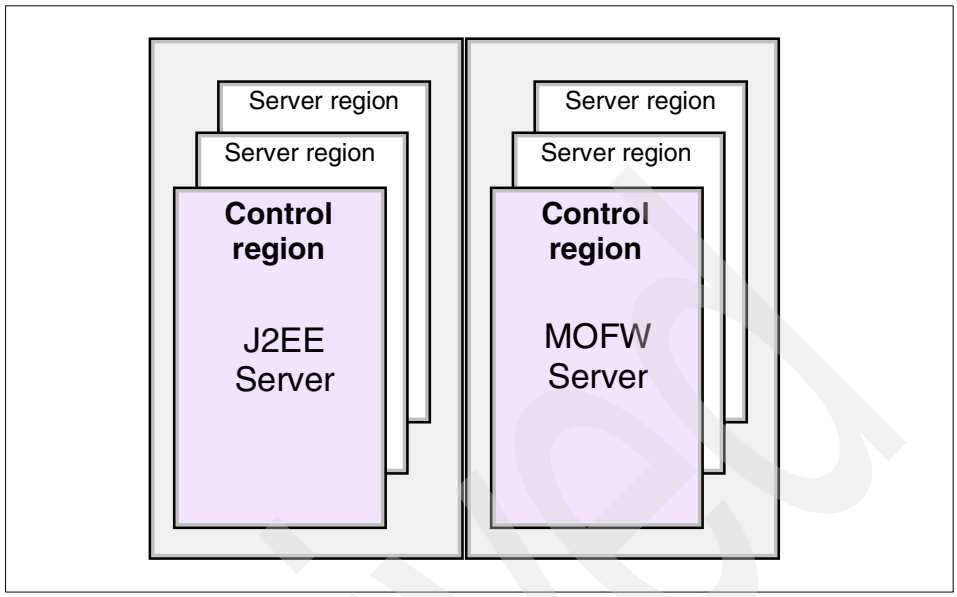


Figure 1-2 J2EE and MOFW servers

J2EE Application servers

This is where the EJBs and Web Containers live. One of the main functions of a J2EE Application server is to serve EJBs.

Enterprise Java Beans (EJBs) provide you with the ability to write your business objects on any Java-enabled platform and then run them unchanged on any other Java-enabled platform.

You can develop and even test your EJBs on Intel and Windows platforms and then run them, in production, on zSeries and z/OS platforms.

In addition, EJBs enable you to share and extend your business objects. The use of visual tools (and the fact that EJBs can dynamically disclose their capabilities) allows you to easily extend their capabilities and tie EJBs together. This in turn allows you to leverage your IT investment for the maximum amount of return.

IBM provides an Installation Verification Program (IVP) named BBOASR2A which can be used to model your J2EE application server environments.

Managed Object FrameWork Application Server

The Managed Object FrameWork (MOFW) Server is a CORBA-compliant server that runs in the WebSphere Application Server environment. This redbook does not discuss specific tuning of the MOFW environment, but the tuning for the Basic Environment is similar for both MOFW and J2EE Application servers.

IBM provides an IVP named BBOASR1A which can used to model your MOFW application server environments.

For more information about these components, refer to *e-business Cookbook for z/OS Volume I: Technology Introduction,* SG24-5664.

1.2.3 WebSphere Application Server overview

Two types of environments are supported in Version 4.0.1 of the WebSphere Application Server. You can run servlets that talk to EJBs from either a plugin (see Figure 1-3 on page 12), or a Web Container environment (see Figure 1-4 on page 13).

Plugin environment (simple configuration option)

The plugin environment is also called the simple configuration option. You have two choices when running servlets in the plugin environment: you can use either the WebSphere Application Server 3.5, or the WebSphere Application Server 4.0.1 plugin.

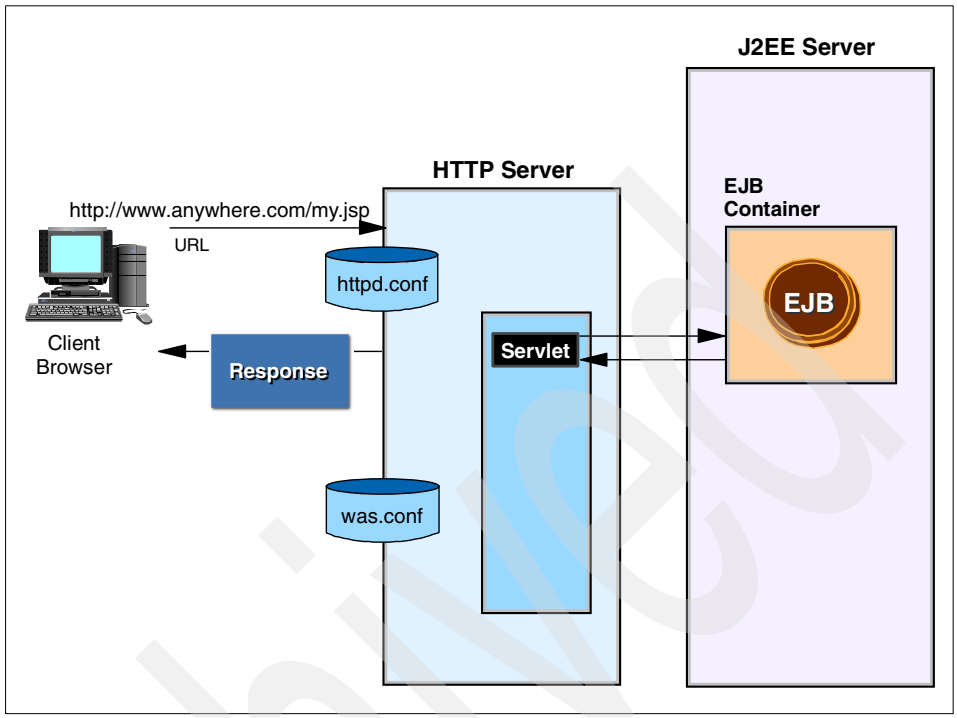


Figure 1-3 Servlets running in a WebSphere Application Server Plugin

Web Container environment

If you decide to run your servlets in the Web Container environment, you also have two choices. You can access the Web Container via the WebSphere Application Server 4.0.1 Plugin (see Figure 1-4 on page 13), or you can use the new HTTP/HTTPS Transport Handler (see Figure 1-5 on page 14) that's part of each J2EE server instance.

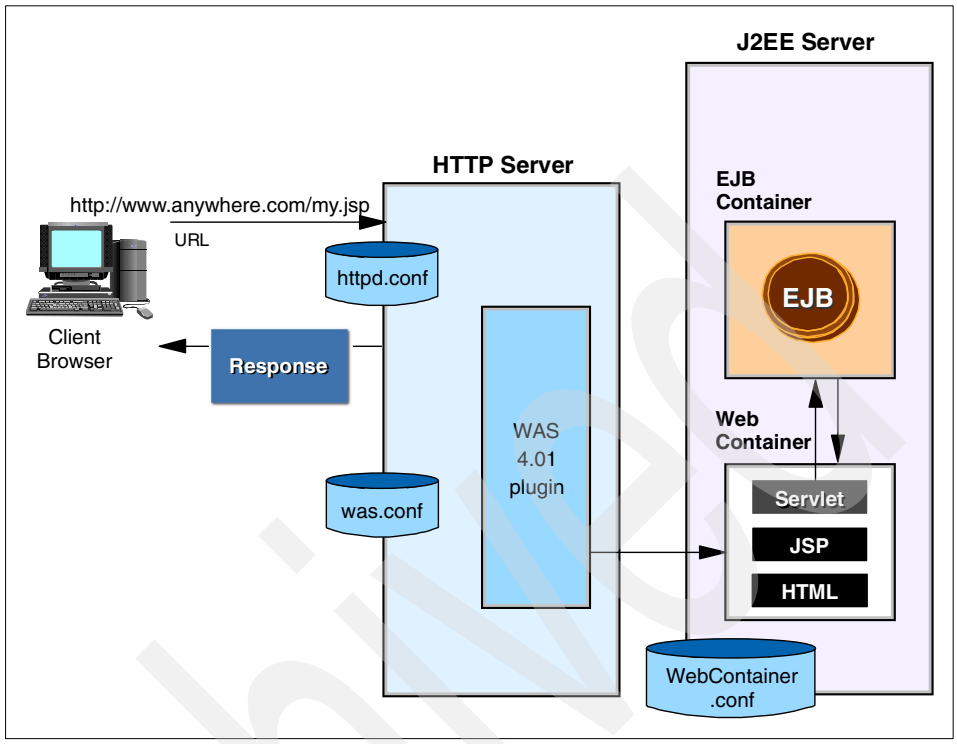


Figure 1-4 Servlets running in Web Container

Only the WebSphere Application Server 4.0.x plugin can be used to talk to servlets in Web Containers.

One of the benefits of having the Web Container and EJB Container in the same address space is the reduced amount of interaddress communication needed (see Figure 1-3 on page 12). Initial testing has shown this environment performs better than running servlets in the plugin itself.

HTTP Transport Handler

A new function in WebSphere Application Server Version 4.0.1 is the HTTP Transport Handler. The HTTP Transport Handler runs in the Control Region.

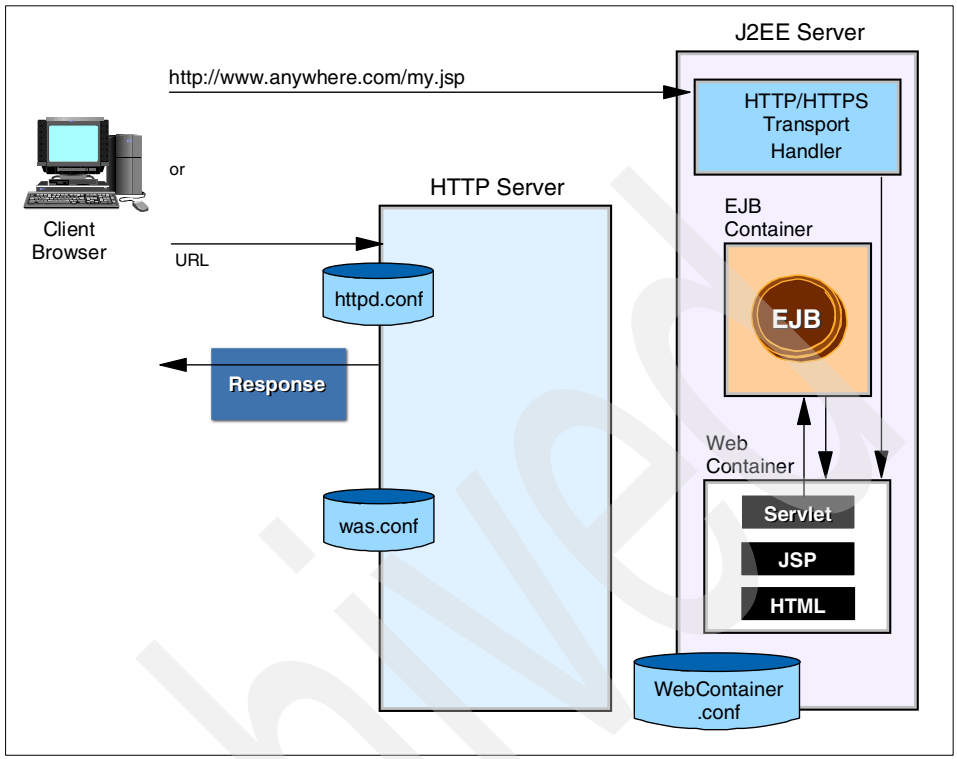


Figure 1-5 The HTTP Transport Handler

The Transport Handler is an HTTP/HTTPS listener integrated into the WebSphere Application Server 4.0.1 runtime environment. It is designed to listen for HTTP/HTTPS requests and route them to the Web Container. It is part of the Control Region, which brings significant performance improvements over the HTTP Server and plugin.

The Transport Handler can be viewed as the performance alternative for the HTTP Server and plugin. Typically you would use one or the other, although it is possible to use both concurrently in the same WebSphere configuration.

The level of your applications may also limit your configuration choices. If you intend to run servlets or JSP written to older specification levels, the HTTP Transport Handler may not be a valid option. For such applications, you may have to use the HTTP Server along with the V3.5 run-time environment.

Table 1-1 Levels of servlet, EJB, JSP, and Java

	Servlet API	EJB level	JSP level	JVM level
WebSphere 4.01 on z/OS	2.2	1.1, 1.0	1.1	1.3
WebSphere 3.5 SE on z/OS	2.1, 2.2	no support	0.91, 1.0, 1.1	1.3

1.2.4 System Management User Interface (SMUI)

System management of WebSphere on z/OS is handled through Administration and Operations applications. These applications run on a Windows workstation and communicate with the System Management server. They are often called the System Management User Interface (SMUI) or System Management End User Interface (SMEUI).

Note: For installation, you must always use the SMUI file bboinst.exe that comes with WebSphere for z/OS or OS/390. The file is located in directory /usr/lpp/WebSphere/bin/.

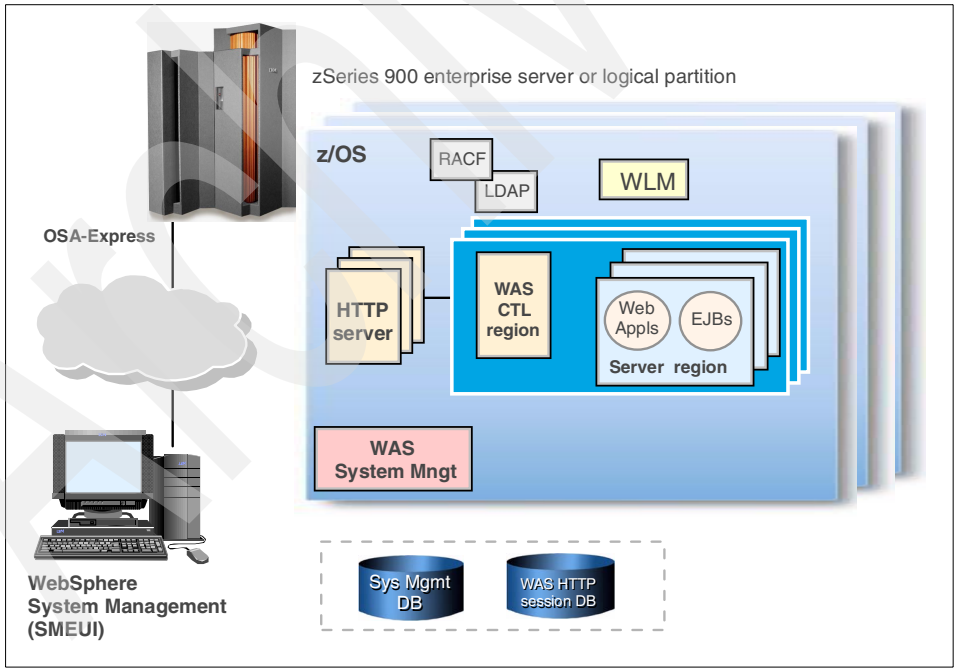


Figure 1-6 WebSphere System Management User Interface

The administration application allows you to display and modify WebSphere for z/OS applications and the environment in which they run. For a new EJB application to be deployed, the SMEUI does the following:

- Generates and compiles the stubs and skeletons for the enterprise bean.
- Sets up the security environment to host the enterprise bean according to its deployment descriptor.
- Sets up the transaction environment for the enterprise bean according to its deployment descriptor.
- Registers the enterprise bean, its environment properties, resource references, etc.

The operations application lets you manage WebSphere for z/OS servers and server instances. For more information on operation tasks and objects refer to *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface,* SA22-7838.

1.3 Monitoring performance

Monitoring performance starts with collecting the appropriate information and then analyzing it. The primary goal of tuning is to adjust system resource levels to manage performance to an acceptable level as defined by a Service Level Agreement (SLA).

Typically, monitoring results in reports showing the following:

- Processor utilization by workload type
- Processor storage allocation by workload type
- I/O activity by workload type
- I/O subsystem response time analysis
- Information reflecting levels of resource **contention.**

WebSphere information

In addition to this information, you can also collect and analyze information that is unique to the HTTP Server, to the Web, and to EJB containers.

The goal is to create a proactive process where problems are identified *before* users start complaining. The first step in this process is to define business objectives and identify a way to measure those objectives. If the objectives are not being met, then actions can be initiated to isolate and identify the problem. Once the cause of the problem has been identified, you can decide whether it is a tuning issue or the result of a lack of resources.

1.3.1 SMF records

SMF records can be collected to capture information about the performance of the system and the WebSphere Application Environment.

After the raw SMF data has been collected, it has to be interpreted and analyzed. There are several different programs that can be used to analyze the data, such as RMF, Tivoli Performance Reporter, or a Java utility that is provided with the WebSphere Application Server on z/OS.

For more information about the Java utility for interpreting SMF WebSphere Application Server data, see “SMF Record Interpreter” on page 33.

1.4 Configuration files

Once performance bottlenecks are identified, there are several configuration files that can be changed to tune the system. The files fall into two broad categories, z/OS system files and HFS file. These are the same files that are used to set up your initial configuration.

1.4.1 z/OS files

The following z/OS files are found in the SYSx.PARMLIB that your system installation uses. See the appendixes in this redbook for more detailed information concerning these files:

SYSx.PARMLIB(SMFPRMxx)	Systems Management Facility
SYSx.PARMLIB(BPXPRMxx)	USS parameters

The following z/OS files are found in the SYSx.PROCLIB that your system installation uses. These files are the startup procedures that we used:

SYSx.PROCLIB(IMWEBSRV)	Web Server - HTTP startup procedure
SYSx.PROCLIB(BBODMN)	Daemon server startup procedure
SYSx.PROCLIB(BBOSMS)	System Management server startup procedure
SYSx.PROCLIB(BBONM)	Naming server startup procedure
SYSx.PROCLIB(BBOIR)	Interface Repository server startup procedure.

1.4.2 HFS files

The following HFS files are used for tuning the WebSphere Application Server Environment:

httpd.conf	HTTP Server configuration file.
httpd.envvars	HTTP Server environment variables.
was.conf	Configuration file for Web applications (servlets) run in the HTTP Server Plugin.
webcontainer.conf	Configuration file for Web applications run in the Web Container of the J2EE Server.
current.env	J2EE Server Environment file.
slapd.conf	LDAP configuration file.
profile	User configuration file.

2

Monitoring the system

While our redbook recommends various settings to help set up your z/OS WebSphere Application Server environment, it's important to note that these are just initial starting points in tuning your environment. Since each environment is unique, you need to determine if the recommendations are appropriate. In addition, there are a number of settings that you will want to adjust as your system grows or changes.

2.1 Introduction

In order to determine what tuning should be done, you first need to measure the performance of your WebSphere Application Server environment.

One important source of information you can use to help you accomplish this task is System Management Facilities or SMF (see 2.3, “SMF Records” on page 21). In addition to system level information, it will provide specific information about the HTTP and WebSphere Application Servers and most subsystems running in your configuration.

Aside from collecting the SMF data, you will also need the ability to analyze it. There are a number of ways of doing this, such as by using RMF, Tivoli Performance reporter, or other available tools. A Java utility is also provided to analyze type 120 records (see 2.4, “SMF Record Interpreter” on page 33).

Once you have identified a particular bottleneck, then you can attempt to correct the imbalance or—if the system is already at maximum throughput, take other steps to increase the capacity (that is, more CPUs, storage, I/O paths, and so on).

Note: It is impossible to accurately measure performance without actually impacting the performance of what you are trying to measure. Every software measurement tool generates a certain amount of overhead (CPU time, memory, and so on). For this reason, the general recommendation is not to turn on specific SMF records unless you are actually going to use them.

With System Management Facilities, you can request that configuration and performance data be recorded to SMF datasets. With this recorded configuration and statistical data, you can analyze Web server health, throughput, and activity.

2.2 Enabling SMF recording

To enable SMF to receive the records, you must update the SMFPRMxx member of SYSx.PARMLIB and specify the types of records that you want recorded. See Appendix A, “SMFPRMxx” on page 125 for a detailed description of the SMFPRMxx member.

2.2.1 SYS operand

In the SMFPRMxx member, the **SYS** operand is used to turn on SMF recording, and this operand can be followed by the **TYPE** suboperand.

TYPE suboperand

The **TYPE** suboperand is used to select which SMF records types are collected. The following example shows record types 30, 71 through 79, 92, 103 and 120 records being collected:

```
SYS(TYPE(30,71:79,92,103,120))
```

Selecting Subtypes

You also have the option of selecting individual record subtypes, so you could choose to record a limited range of subtypes. For example, if you wanted to record all SMF type 92 records, with the exception of subtypes 10 and 11, you could code the following:

```
SYS(TYPE(92(1,2,4:7,12,13)))
```

Note: Types 3, 8 and 9 are not used. See “SMF type 92 - HFS records” on page 23 for more information regarding SMF type 92 records.

2.3 SMF Records

There are a number of SMF record types that can be collected to monitor the performance of your e-business environment:

Record Type 30	Address Space - accounting information
Record Type 92	USS HFS information
Record Type 103	HTTP Server information
Record Type 120	WebSphere Application Server and Container information
Record Type 100	DB2 statistics, records transaction data collected at event monitoring points.
Record Type 101	DB2 accounting, resources used during a transaction.
Record Type 102	DB2 performance.
Record Type 110	CICS/ESA Statistics, contains transaction data collected at event monitoring points.
Record Type 115 and 116	WebSphere MQ Statistics.
Record Type 118 and 119	TCP/IP Statistics. Telnet and FTP servers, API calls, and Telnet and FTP client calls.

Refer to the documentation for each subsystem for more information on the use of SMF records for DB2, CICS, WebSphere MQ or TCP/IP.

The following SMF record types can also be collected and analyzed to monitor the overall performance of your system:

Record Type 70	RMF Processor Activity
Record Type 71	RMF Paging Activity
Record Type 72	RMF Workload Activity and Storage Data
Record Type 73	RMF Channel Path Activity
Record Type 74	RMF Activity of Several Resources
Record Type 75	RMF Page Data Set Activity
Record Type 76	RMF Trace Activity
Record Type 77	RMF Enqueue Activity
Record Type 78	RMF Monitor I Activity
Record Type 79	RMF Monitor II Activity

2.3.1 SMF type 30 - Accounting records

Type 30 records are written to provide system-wide accounting information on a periodic basis. They consolidate data that is found in record types 4, 5, 20, 34, 35 and 40 (which simplifies accounting by installation-written post-processing routines), and they provide additional information.

SMF type 30 records provides resource usage information for traditional z/OS work units such as a TSO/E session, started task or batch job. They are well suited for traditional applications, but are usually of little value for analyzing activity for transactional or e-business workloads.

SMF writes type 30 records when:

- A work unit (such as a TSO/E session, started task, or batch job) starts. This subtype 1 record identifies the work unit, but contains no resource data.
- An SMF interval ends, if you requested interval accounting. For global interval recording without interval synchronization, this span of time is the same as the SMF global recording interval. For other intervals, this subtype 2 record contains the total resources used from the end of the previous interval until the end of the current interval.
- A work unit such as a TSO/E session, started task, or batch job, completes. The following lists key SMF type 30 subtype 2 and 3 record fields.

  Common address space work activity section:

SMF30JBN	Job Name
SMF30EXN	Program name of UNIX service process
SMF30CPT	CPU time
SMF30CPS	SRB time
SMF30WST	Storage frames
SMF30PGI	Page-in activity

SMF30NSW	Swap activity
SMF30TEP	Total blocks transferred

UNIX application/job resource usage section:

SMF30OPI	UNIX process ID
SMF30OFR	HFS reads to regular files
SMF30OFW	HFS writes to regular files
SMF30ODR	HFS directory reads
SMF30OSC	Total service request

For more information concerning SMF type 30 records see *z/OS V1R3.0 MVS System Management Facilities (SMF)*, SA22-7630.

2.3.2 SMF type 70-79 - RMF records

Understanding resource usage by workload is the first step to effective performance management. Before you go down to the application level, you should first obtain a view of all workloads in your system to understand your system behavior and try to identify where the problem is. RMF records, SMF type 70-79, provide the system data you need for system and workload analysis.

In the Workload Activity record (SMF record 72), RMF records information about workloads grouped into categories that you have defined in the Workload Manager policy. They can be workloads, service classes, or they can be logically grouped into reporting classes.

If you are performance-conscious, you probably want to enable recording of SMF records 70-72 on a regular basis so as to be able follow the performance of your system. Other RMF records may be activated when a performance problem has been detected and you need to do a more detailed analysis.

SMF records 70-79 are written at specific time intervals. It is good practice to synchronize the RMF recording interval with the SMF recording interval specified in member SMFPRMxx. Option SYNC(SMF) in SYS1.PARMLIB member ERBRMFxx allows you to synchronize the RMF recording interval with SMF.

SMF records 70-78 can be exploited through RMF reports. For more information on the use of RMF and the reporting of data, either online or through the postprocessor, see *Resource Measurement Facility User's Guide,* SC33-7990.

2.3.3 SMF type 92 - HFS records

Type 92 records are written at a UNIX Systems Services level and provide information about the HFS file system.

In order to collect information about the activity of a USS HFS mounted file system, you must be collecting SMF type 92 subtype 5 (unmount) records at the time the file system is mounted and at the time the file system is unmounted.

In order to collect information on the activity of a specific USS HFS file, you must be collecting SMF type 92 subtype 11 (close) records at the time the file is opened and at the time the file is closed.

The subtypes are as follows:

Subtype 1	Written when a file system is mounted
Subtype 2	Written after a file system is quiesced or suspended
Subtype 3	Not used
Subtype 4	Written after a file system is unquiesced or resumed
Subtype 5	Written when a file system is unmounted
Subtype 6	Written when a file system is remounted
Subtype 7	Written when a file system is moved
Subtype 8	Not used
Subtype 9	Not used
Subtype 10	Written when a file is opened
Subtype 11	Written when a file is closed
Subtype 12	Contains MMAP subtype information
Subtype 13	Contains MUNMAP subtype information

Recommendation: When running an e-business workload on z/OS, a very large number of HFS files may be opened and closed. Recording of SMF type 92 subtypes 10 and 11 may lead to a significant performance penalty because of the large number of records being written to the SMF datasets.

Unless you have a specific reason, it is recommended that you suppress the recording of SMF record 92 subtypes 10 and 11.

For more information about SMF type 92 records see *z/OS V1R3.0 MVS System Management Facilities (SMF)*, SA22-7630.

2.3.4 SMF type 103 - HTTP Server records

Type 103 records are written for the purpose of monitoring the IBM HTTP Server. Though information recorded here pertains only to activity occurring in the HTTP server address space, it is a good source of information.

There are two subtypes for HTTP Server type 103 records:

subtype 1	Configuration information
subtype 2	Performance information

Configuration record data is taken from the server configuration file, **httpd.conf**, and is written after the HTTP server is fully initialized.

Performance record data is accumulated continuously and written at intervals defined in the **httpd.conf** file by the **SMFRecordingInterval** directive.

Performance data

Some types of performance data that is recorded:

TotalCurrentThreads	Number of threads currently used
MaxThread	Maximum number of threads defined
Request	Cumulative number of requests since HTTP Server startup
RequestErrors	Cumulative number of request errors since startup
Response	Cumulative number of responses since startup
ResponseDiscard	Cumulative number of responses discarded since startup
InBytes	Cumulative number of bytes received since startup
OutBytes	Cumulative number of bytes sent since startup
InUnknowns	Cumulative number of unknown type bytes since startup
TotalTimeOuts	Cumulative number of timeouts since startup

Turning SMF 103 Recording on and off

In addition to enabling SMF 103 records in SYSx.PARMLIB member SPFPRMxx (see “Enabling SMF recording” on page 20), you must tell the HTTP Server to produce type 103 records.

There are three ways to accomplish this task:

- Turn on SMF in the HTTPD.CONF file.
- Specify `-smf` on the HTTP Server startup procedure.
- Issue a **MODIFY** operator command to turn on SMF recording for the HTTP Server.

HTTPD.CONF file

You can modify the following directives in the HTTPD.CONF file to turn SMF support on or off for type 103 records:

```
SMF                         on
SMFRecordingInterval        00:15
```

You can also specify the recording interval in the http.conf file. See Appendix E, “httpd.conf” on page 143 for a complete description of the http.conf file.

Startup Procedure

Use the following flags to turn the SMF support in the HTTP Server on and off when executing the SYS1.PROCLIB(IMWEBSRV) startup procedure:

-smf	turns SMF support on
-nosmf	turns SMF support off

This overrides what is specified in the httpd.conf file.

z/OS Modify command

You can also use the z/OS operator console MODIFY command to turn SMF on or off.

F IMWEBSRV,APPL=-smf	to turn on SMF
F IMWEBSRV,APPL=-nosmf	to turn off SMF

To actually be able to write type 103 record, the recording must also be enabled through the SYS keyword in SYS1.PARMLIB(SMFPRMxx). If SMF recording is disabled for SMF record 103, a MODIFY command to turn it on at the HTTP server level will not override the system option.

For more information concerning SMF type 103 records see *z/OS HTTP Planning, Installing and Using V5.3,* SC34-4826.

2.3.5 SMF Records 100-102, 110 and 115

These are SMF records for DB2, CICS and WebSphere MQ. Refer to the documentation for each subsystem for more information on the use of SMF records.

2.3.6 SMF type 120 - Application Server/Container records

Type 120 SMF records contain information about both the WebSphere Application Server and Container performance and usage. If you wish to record activity at the application level on your WebSphere Application server, SMF records type 120 are one primary source of information.

Note: The following description of SMF record 120 assumes WebSphere Application Server V4.01with PTF UQ90049 applied (service level L00PTF11).

As stated earlier, the general recommendation is not to turn on specific SMF records unless you're actually going to use them. Although at the time of writing there are not many tools available to exploit SMF records type 120, it is expected that this situation will change. The SMFBROWSER tool is a utility which allows the interpretation of SMF record 120; see "SMF Record Interpreter" on page 33.

If knowledge of WebSphere application performance is important, we recommend that you enable recording of SMF records 120.

If your choice is also to generate Server or Container interval records 120, it is probably a good idea to synchronize the record 120 interval with the SMF and RMF recording interval (see 2.3.2, "SMF type 70-79 - RMF records" on page 23).

Synchronizing recording intervals for SMF, RMF and WebSphere components will allow for consistent system workload and application performance information.

Note: If you enable recording for SMF record 120, make sure that you apply on your system the fix to APAR PQ55355. This fix brings a significant performance improvement.

SMF 120 record types

Two types of SMF records can be produced: *activity records* and *interval records*.

Activity records Gathered as each activity within a server is completed. An activity is a logical unit of business function. It can be a server or user-initiated transaction.

Interval records Consist of data gathered at installation-specified intervals and provide capacity planning and reliability information.

The above type records can be recorded for servers, containers, J2EE containers and Web containers, so there are a total of eight subtypes of records.

- Subtype 1, Server activity record
- Subtype 2, Container activity record (MOFW)
- Subtype 3, Server interval record
- Subtype 4, Container interval record (MOFW)
- Subtype 5, J2EE container activity record
- Subtype 6, J2EE container interval record
- Subtype 7, WebContainer activity record
- Subtype 8, WebContainer interval record

Server activity record - Subtype 1

The server activity SMF record is used to record activity that is running inside a WebSphere for z/OS Application Server. This record can be used to profile your applications to determine, in detail, what is happening inside the WebSphere transaction server.

A single record is created for each activity that is run inside a server or server instance. If the activity runs in multiple servers, then a record is written for each server. Each record contains up to two sections:

Server activity	Contains information about each activity that occurred within one server. There is one section per record.
Communications session	Contains information about each communication session. There can be multiple sections per record.

You can activate this record through the server definition of the Systems Management User Interface by checking the checkbox: **`Write Server Activity SMF Records`** for the selected server; see Figure 2-1 on page 32.

Container activity record - subtype 2

The purpose of the container activity SMF record is to record activity that is running inside a container located inside a WebSphere MOFW server. A single record is created for each activity that is run inside a container located in a WebSphere Application server. If the activity runs in multiple servers, then multiple records are written for the activity.

Each record contains multiple sections:

Container activity	Contains information about each activity that occurred within one container. There is one section per record.
Class	Contains information about all classes involved in this activity. There can be multiple sections per record.
Method	Contains information about all methods of classes involved in this activity. There can be multiple Method sections per class section.

You can activate this record through the server definition of the System Management User Interface by checking the checkbox: **`Write Container Activity SMF Records`** for the selected server; see Figure 2-1 on page 32.

Server interval record - subtype 3

The purpose of the server interval SMF record is to record activity that is running inside a WebSphere Application server. This record is produced at regular intervals and is an aggregate of the work that ran inside the server instance during the interval.

A single record is created for each server instance that has interval recording active during the interval. If a server has multiple server instances, then a record for each server instance is written and the records must be merged after processing to get a complete view of the work that ran inside the server.

You can activate this record through the server definition of the System Management User Interface by checking the checkbox **`Write Server Interval SMF Records`** for the selected server; see Figure 2-1 on page 32.

Container interval record - subtype 4

The purpose of the container interval SMF record is to record activity that is running inside a container located inside a WebSphere MOFW server. This record is produced at regular intervals and is an aggregate of the activities running inside a container during the interval.

A single record is created for each active container located in a WebSphere Transaction Server within the interval being recorded. If there is more than one server instance associated with a server, there will be a record for the container from each server instance. To get a common view of the work running in the container during the interval, you must merge the records after processing.

You can control recording of this record through the server definition of the System Management User Interface by checking the checkbox **`Write Container Interval SMF Records`** for the selected server; see Figure 2-1 on page 32.

J2EE container activity record - subtype 5

The purpose of the J2EE container activity SMF record is to record activity within a J2EE container that is located inside the WebSphere Transaction Server.

A single record is created for each activity that is run within a J2EE container located inside a WebSphere Transaction Server. Each record contains three sections.

J2EE container activity Contains information about each activity that occurred within one J2EE. There is one section per record.

Bean Contains information about all beans involved in this activity. There can be multiple sections per record

Bean method Contains information about all methods of beans involved in this activity. There can be multiple Bean method sections per Bean section.

You can control recording of subtype 5 through the J2EE server definition of the Systems Management User Interface by checking the checkbox titled **`Write Container Activity SMF Records`** for the J2EE Server; see Figure 2-1 on page 32.

J2EE container interval record - subtype 6

The purpose of the J2EE container interval SMF record is to record activity within a J2EE container that is located inside the WebSphere transaction server. This record is produced at regular intervals and is an aggregate of the activities running inside a J2EE container during the interval.

A single record is created for each active J2EE container located in a WebSphere transaction server within the interval being recorded. If there is more than one server instance associated with a server, a record for the container will exist for each server instance.

You can control recording of subtype 6 through the J2EE server definition of the Systems Management User Interface using the checkbox titled **`Write Container Interval SMF Records.`**

WebContainer activity record - subtype 7

The purpose of the WebContainer activity SMF record is to record activity within a Web container running inside a J2EE server.

The Web container is deployed within an EJB and runs within the EJB container. The Web container acts as a Web server handling HttpSessions and Servlets. The EJB container is not aware of the work the Web container does. Instead, the EJB container only records that the EJB has been dispatched. Meanwhile, the Web container gathers the detailed information, such as HttpSessions, Servlets, and their respective performance data.

A single WebContainer Activity record is created for each activity that is run within a Web container. Each record contains multiple sections:

WebContainer activity	Contains information about each activity that occurred within one Web container. There is one section per record.
HttpSessionManager	Contains information about all http sessions associated to one single activity. There is up to one section per record.
WebApplication	Contains information about all WebApplications involved in this activity. There can be multiple sections per record.

Servlet activity Contains information about each servlet associated with WebApplications involved in this activity. There can be multiple Servlet activity sections per WebApplication section.

The WebContainer SMF record is not uniquely selectable. It is activated and deactivated along with the activation and deactivation of SMF activity recording for the J2EE container.

WebContainer interval record - subtype 8

The purpose of the WebContainer interval SMF record is to record activity within a Web container running inside a J2EE server. The Web container execution environment consists of an EJB that is deployed into the EJB container. The Web container acts as a Web Server handling HttpSessions and Servlets. The EJB container is not aware of the purpose of the WebContainer activity record and only records that the EJB has been dispatched, but does not gather any of the detailed information, such as HttpSessions, Servlets, and their respective performance data.

A single WebContainer record is created for each Web container. In addition to data that is associated with an individual activity, there are some cases of Web container work that are performed outside the scope of an individual request. For example, some instances of http session finalization and http session invalidation are performed asynchronously. In such a case a WebContainer interval record would record this data.

The WebContainer interval record is not uniquely selectable. It is activated and deactivated along with the activation and deactivation of SMF interval recording for the J2EE container.

Turning SMF 120 recording ON and OFF

As with SMF HTTP records, in addition to enabling SMF recording at the system level (see “Enabling SMF recording” on page 20), you must also turn on the production of type 120 records in WebSphere.

When enabled in the SMF options (SMFPRMxx), recording of SMF record 120 at the server/container level is controlled via the Systems Management User Interface, from the Administration Console.

You can check and modify the status of SMF record 120 recording for each server or container, as shown on Figure 2-1 on page 32.

Four checkboxes allow you to turn *ON* and *OFF* server and container recording.

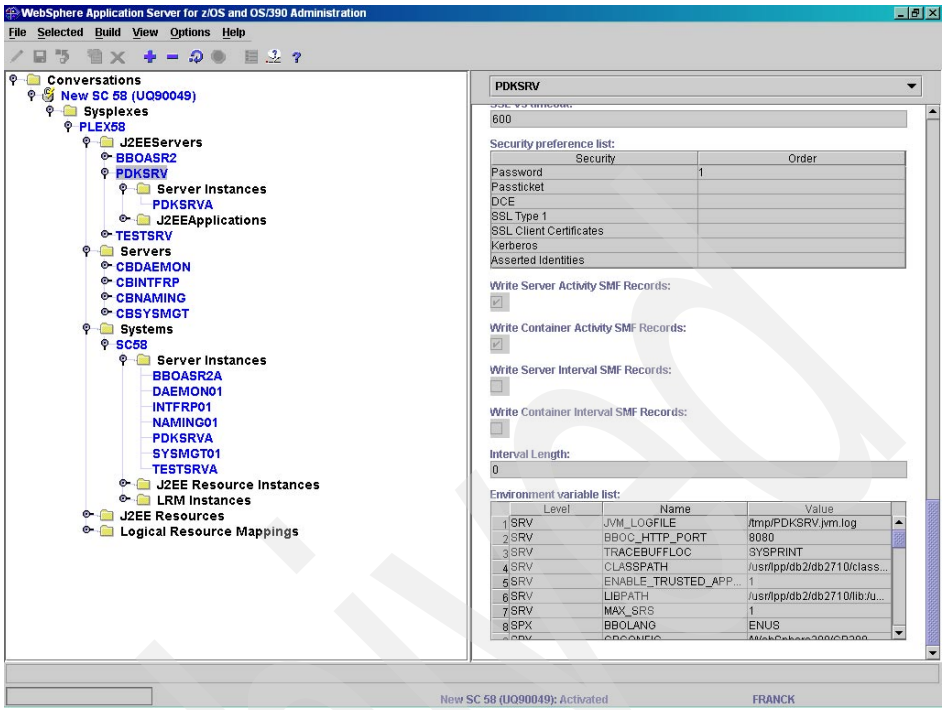


Figure 2-1 Checking / Modifying SMF record 120 recording status

Table 2-1 summarizes how record 120 subtypes are generated depending on which options you select.

Table 2-1 Recording controls for SMF record 120

	For a J2EE Server	For a MFOW Server
Write Server Activity SMF Records	subtype 1	subtype 1
Write Server Interval SMF Records	subtype 3	subtype 3
Write Container Activity SMF Records	subtype 5 subtype 7	subtype 2
Write Container Interval SMF Records	subtype 6 subtype 8	subtype 4

If you decide to activate interval recording, the length of time for the recording interval must also be set when you specify **Write Server/Container Interval SMF Records**.

- If you set the recording interval value to zero, WebSphere uses the value from the INTERVAL statement in the SMFPRMxx parmlib member. If no INTERVAL statement is specified in SMFPRMxx, the system default interval is 30 minutes.
- if you do not specify any interval value the default recording interval for SMF record 120 is set to one hour.

See *WebSphere Applications Server V4.0 for z/OS and OS/390: Operations and Administration,* SA22-7835 for more information on SMF type 120 records.

2.4 SMF Record Interpreter

The WebSphere for z/OS SMF Record Interpreter is a tool which enables the interpretation of complete output datasets from the IBM z/OS utility program IFASMFDP. It writes a header line for all SMF record types and a detailed dump for SMF record type 120. The version we used only interprets record 120 subtypes 1 to 6. Subtypes 7 and 8, that are added with PTF UQ90049, were not recognized.

The tool is a Java utility, so it needs to be interpreted and executed by a Java Virtual Machine (JVM) under the z/OS or OS/390 UNIX environment.

2.4.1 Install the z/OS SMF Record Interpreter

The tool can be downloaded from the WebSphere Application Server for z/OS and OS/390 Web site at:

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

Select **WebSphere Application Server for z/OS Downloads** to access the z/OS tools download site.

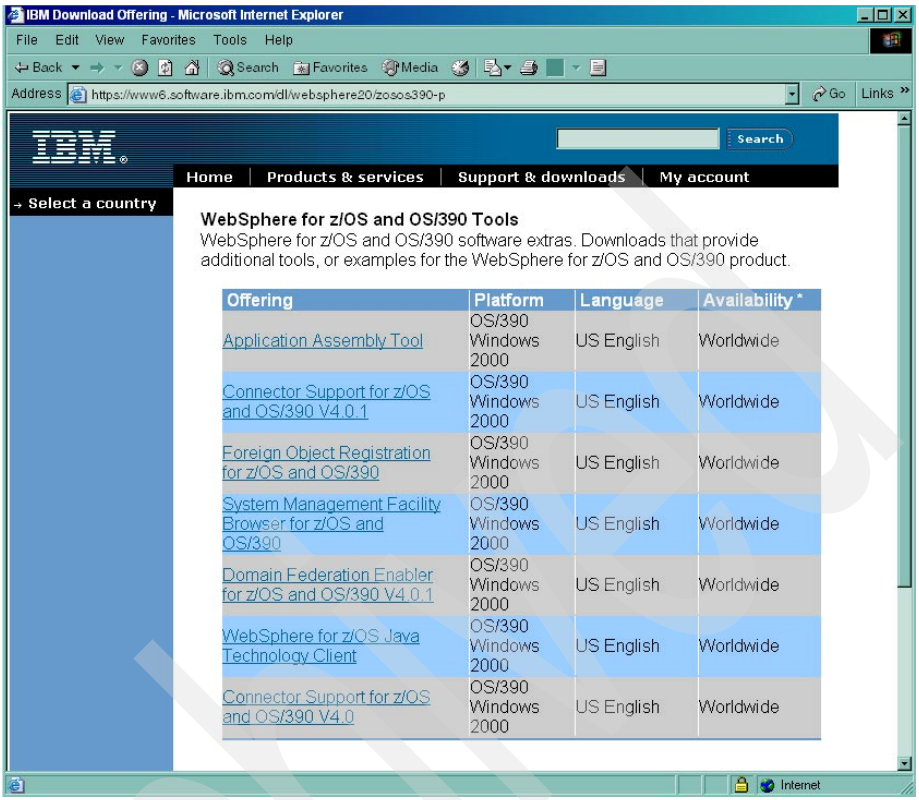


Figure 2-2 Obtaining the SMF Record Interpreter

After you have registered and accepted the license agreement, you can download the file smfbrowser.exe. This is a self-extracting file and will unzip on your MS Windows workstation into executable, documentation and source files.

- smfbrowser.pdf contains the documentation.
- smfbrowser.jar is the executable that you should load on your z/OS system. It is a Java utility, so it needs to be interpreted and executed by a Java Virtual Machine (JVM) under the z/OS or OS/390 UNIX environment. To use it from the z/OS UNIX environment you must:

  a. Upload it onto the tools directory on your z/OS system, using ftp (binary mode).

  b. Verify that the JAVA_HOME environment variable refers to the current Java installation, eg. JAVA_HOME=../usr/bin/java/J1.3 (this should be at least Java 1.3).

An enhanced version of the SMF Browser, called WAS V. 4.0.1 for z/OS SMF Summary Viewer, has been made available by the Washington Systems Center on the Advanced Technical Support Information Web Site. For the latest version, refer to document WP100244 in the white paper category at:

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

The SMF Summary Viewer is an extension of the program distributed from the WAS V4 for z/OS download site. It adds a summary report showing activity for each J2EE Server Instance, Bean, and Method from the SMF type 120 records.

2.4.2 Running the z/OS SMF Record Interpreter

To run the SMF Record Interpreter, you must first create a sequential file containing the SMF records. The IFASMFDP SMF utility can be used to extract SMF records from the system datasets and create a sequential dataset.

Since the tool only does not interpret any SMF records other than record 120, it is recommended that you filter out all other records. The following example dumps SMF records 120 from system datasets SYS1.SC58.MAN1 and SYS1.SC58.MAN2 into a sequential file named FRANCK.SC58T.SMF.

Example 2-1 Using IFASMFDP to copy SMF records into a sequential file

```
//JAVA31 JOB  999,'ITSO',
//         MSGCLASS=T,NOTIFY=&SYSUID,CLASS=A
//DUMP1    EXEC  PGM=IFASMFDP
//INSMF1   DD DSN=SYS1.SC58.MAN1,DISP=SHR
//INSMF2   DD DSN=SYS1.SC58.MAN2,DISP=SHR
//SMFDATA  DD DSN=FRANCK.SC58T.SMF,
//         DCB=(RECFM=VBS,LRECL=32760),
//         SPACE=(CYL,(5,10)),
//         UNIT=SYSALLDA,
//         DISP=(NEW,CATLG)
//*
//SYSPRINT DD    SYSOUT=*
//SYSIN    DD    *
       OUTDD(SMFDATA,TYPE(120))
       INDD(INSMF1,OPTIONS(DUMP))
       INDD(INSMF2,OPTIONS(DUMP))
```

The WebSphere for z/OS SMF Record Interpreter dumps all the WebSphere for z/OS-relevant data into a printable output file:

```
java -cp bbomsmfv.jar com.ibm.ws390.sm.smfview.Interpreter "SMF_file_Name"
```

For example, to interpret data from a cataloged sequential file named FRANCK.SC58T.SMF (previously created using the IFASMFDP utility, as described in Example 2-1) and send the output to file SMF58out.txt, you would go in the TSO OMVS shell and execute the command as shown in Figure 2-3.

```
IBM
Licensed Material - Property of IBM
5694-A01 (C) Copyright IBM Corp. 1993, 2001
(C) Copyright Mortice Kern Systems, Inc., 1985, 1996.
(C) Copyright Software Development Group, University of Waterloo, 1989.
All Rights Reserved.
U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
Disclosure restricted by GSA-ADP schedule contract with IBM Corp.
IBM is a registered trademark of the IBM Corp.
-------------------------------------------------------------
Set up environment variables for Java and Servlets for OS/390 -
-------------------------------------------------------------
PATH reset to .:/usr/lpp/java/IBM/J1.3/bin:/usr/lpp/Printsrv/bin:/bin:.
PATH is /var/iwl/bin:.:/usr/lpp/java/IBM/J1.3/bin:/usr/lpp/Printsrv/bin:/bin
:.:/u/franck:
FRANCK:/u/franck: >


 ===> java -cp bbomsmfv.jar com.ibm.ws390.sm.smfview.Interpreter "FRANCK.SC
58T.SMF" >SMF58out.txt
INPUT
ESC=¢  1=Help      2=SubCmd    3=HlpRetrn  4=Top       5=Bottom   6=TSO
       7=BackScr   8=Scroll    9=NextSess 10=Refresh  11=FwdRetr 12=Retrieve
```

Figure 2-3 Running the SMF record interpreter tool

The output of the z/OS SMF Record Interpreter would be saved in file SMF58out.txt and be available to be browsed or edited through ISPF.

2.4.3 Sample output

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

```
SMF file analysis starts ...

------------------------------------------------------------------------------
Record#: 8;
  Type: 120; Size: 372; Date: Thu Jul 11 15:07:15 EDT 2002;
  SystemID: SC58; SubsystemID: WAS; Flag: 94;
  Subtype: 1 (SERVER ACTIVITY);
```

```
  #Triplets: 3;
  Triplet #: 1; offset: 64; length: 32; count: 1;
  Triplet #: 2; offset: 96; length: 192; count: 1;
  Triplet #: 3; offset: 288; length: 84; count: 1;
  Triplet #: 1; Type: ProductSection;
    Version: 2; Codeset: IBM-1047; Endian: 1; TimeStampFormat: 1 (S390STCK64);
    IndexOfThisRecord: 1; Total # records: 1; Total # triplets: 3;
  Triplet #: 2; Type: ServerActivitySection;
    HostName          : PLEX58;
    ServerName        : PDKSRV;
    ServerInstanceName: PDKSRVA;
    ServerType        : J2EE Server;
    #ServerRegions: 1;
    ASID1: 252; ASID2: 0; ASID3: 0; ASID4: 0; ASID5: 0;
    UserCredentials: CBIVP2;
    ActivityType: 1 (method request);
    ActivityID                * b7e95044 99d19e04 000003fc 0000001a *
                              * 090c0216 -------- -------- -------- *
    WlmEnclaveToken           * 00000020 00000d27 -------- -------- *
    ActivityStartTime         * b7e95044 99d19e04 40404040 40404040 *
    ActivityStopTime          * b7e95044 9c4c0d40 40404040 40404040 *
    #InputMethods      : 1;
    #GlobalTransactions: 0;
    #LocalTransactions : 1;
  Triplet #: 3; Type: CommSessionSection;
    CommSessionHandle         * 23e04940 00000029 -------- -------- *
    CommSessionAddress: ip addr=9.12.14.67 port=4843;
    CommSessionOptimization: 2147483647 (unknown optimization);
    DataReceived: 523; DataTransferred: 1509;
-------------------------------------------------------------------------------
Record#: 9;
  Type: 120; Size: 2960; Date: Thu Jul 11 15:07:15 EDT 2002;
  SystemID: SC58; SubsystemID: WAS; Flag: 94;
  Subtype: 5 (J2EE CONTAINER ACTIVITY);
  #Triplets: 3;
  Triplet #: 1; offset: 64; length: 32; count: 1;
  Triplet #: 2; offset: 96; length: 628; count: 1;
  Triplet #: 3; offset: 724; length: 2236; count: 1;
  Triplet #: 1; Type: ProductSection;
    Version: 1; Codeset: Unicode; Endian: 1; TimeStampFormat: 1 (S390STCK64);
    IndexOfThisRecord: 1; Total # records: 1; Total # triplets: 3;
   Triplet #: 2; Type: J2eeContainerActivitySection;
    HostName          : PLEX58;
    ServerName        : PDKSRV;
    ServerInstanceName: PDKSRVA;
    ServerRegionASID  : 252;
    ContainerName     : Default;
    wlmEnclaveToken           * 00000020 00000d27 -------- -------- *
    activityType: 1 (method request);
```

```
    activityID                * b7e95044 99d19e04 000003fc 0000001a *
                              * 090c0216 -------- -------- -------- *
   Triplet #: 3; Type: BeanSection;
    AMCName: PDKLite::PDKLiteWeb_WebApp.jar::RemoteWebAppBean;
    AMCUuid                   * b7c4e46c de114ac2 00000134 00000006 *
                              * 090c0216 b7c4e46d 2116d881 00000134 *
                              * 00000006 090c0216 b7c4e46d 5f9abe46 *
                              * 00000134 00000006 090c0216 -------- *
    Type            : 2 (Stateless session bean);
    ActivationPolicy: 0 (Once);
    LoadPolicy      : 0 (At activation);
    PinPolicy       : 0 (For activation period);
    ReentrancePolicy: 0 (Not reentrant with transaction);
    LocalTransactionContainment: 0 (Session);
    AccessControl   : 0 (Container managed);
    LocalTransactionOutcome: 0 (Rollback);
    # Methods: 1;
    Triplet #: 3.1; offset: 620; length: 1616; count: 1;
    Triplet #: 3.1; Type: BeanMethodSection;
      MethodName: dispatch;
      # Invocations: 1;
      AverageResponseTime: 6 [sec*10**-3];
      MaximumResponseTime: 6 [sec*10**-3];
      Transaction Policy : 1 (TX_BEAN_MANAGED);
      InvocationIdentity : 0 (Caller);
      Persistence        : 0 (Caller);
      Roles              : ;
      Invocation Locale  : 0 (Caller);
      User session policy: 0 (Not supported);
    ejbLoad  #Invocations: 0; Average: 0 [sec*10**-3]; Maximum: 0 [sec*10**-3];
    ejbStore #Invocations: 0; Average: 0 [sec*10**-3]; Maximum: 0 [sec*10**-3];
    ejbActivate  #Invocations: 0; Average: 0 [sec*10**-3]; Maximum: 0
[sec*10**-3];
      ejbPassivate #Invocations: 0; Average: 0 [sec*10**-3]; Maximum: 0
[sec*10**-3];
-------------------------------------------------------------------------------
Record#: 10;
  Type: 120; Size: 820; Date: Thu Jul 11 15:07:15 EDT 2002;
  SystemID: SC58; SubsystemID: WAS; Flag: 94;
  Subtype: 7 (Unknown WebSphere SMF record subtype);
-------------------------------------------------------------------------------
.../...
SMF file analysis complete.
```

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

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

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

2.5 Displaying the number of threads (HTTP Server)

One way to interactively monitor performance is to periodically display statistics about HTTP Server thread usage by issuing the following modify command to display Server statistics:

```
F IMWEBSRV,APPL=-D STATS
```

The following display shows the total number of threads (`Threads running: 100`) as well as the number Idle Threads (`Threads idle: 91`) and the number of Active Inbound Connections (`Active Inbound Connections: 9`).

The total number of threads defined for the Web Server is split 50-50 between SSL and Non-SSL threads. In the following example, two SSL and four Non-SSL threads are reserved for system usage, leaving a total of 44 SSL and 41 Non-SSL threads waiting or available. ((100 - 6) - 9 active = 44 + 41)

```
IMW3502I Stats: Threads running: 100, Threads idle: 91, 983
Requests: 224582, Bytes rcvd: 95549122, Bytes sent: 701544390,
Active Inbound Connections: 9, Active Outbound Connections: 0.
Connections since last SMF: 101217,
DNS Max: 0.000000, DNS Min: 0.000000, DNS Avg: 0.000000,
Service Plugins Max: 818.749854, Service Plugins Min: 0.004946,
Service Plugins Avg: 0.291400,
CGI Max: 0.000000, CGI Min: 0.000000, CGI Avg: 0.000000,
SSL Handshake Max: 21.578145, SSL Handshake Min: 0.022530,
SSL Handshake Avg: 1.122447,
Proxy Response Max: 0.000000, Proxy Response Min: 0.000000,
Proxy Response Avg: 0.000000
Non-SSL Waiting Threads: 41, SSL Waiting Threads: 44,
Async I/O Waiting Threads: 0, Msg Queue Waiting Threads: 0,
```

In the following display, you'll notice the number of Active inbound connections has increased to the point (48) where there are no more Non-SSL Waiting Threads (0). Once again, note that there are 2 SLL and 4 Non-SSL threads reserved for system usage.

```
IMW3502I Stats: Threads running: 100, Threads idle: 52, 031
Requests: 140565, Bytes rcvd: 59590191, Bytes sent: 446958579,
Active Inbound Connections: 48, Active Outbound Connections: 0.
Connections since last SMF: 66476,
DNS Max: 0.000000, DNS Min: 0.000000, DNS Avg: 0.000000,
Service Plugins Max: 120.045755, Service Plugins Min: 0.004965,
Service Plugins Avg: 0.165007,
CGI Max: 0.000000, CGI Min: 0.000000, CGI Avg: 0.000000,
SSL Handshake Max: 25.670642, SSL Handshake Min: 0.010759,
SSL Handshake Avg: 1.070834,
Proxy Response Max: 0.000000, Proxy Response Min: 0.000000,
Proxy Response Avg: 0.000000
Non-SSL Waiting Threads: 0, SSL Waiting Threads: 46,
Async I/O Waiting Threads: 0, Msg Queue Waiting Threads: 0,
```

In the preceding situation, no new work can be started because of a lack of available threads.

2.6 Displaying CPU utilization for threads (HTTP Server)

In order to display the CPU utilization by thread ID, you first have to know the current Process ID (PID). By searching through the console log, you can find the PID assigned when the server was started in the following message:

```
IMW3534I PID: 83886220 SERVER STARTING
```

Using the PID obtained, you can issue a Display command:

```
D OMVS,PID=83886220
```

In the following example, you'll notice that a thread represented by the TCB at 0098E190 has accumulated an inordinate amount of CPU time 77309.866

```
BPXO040I 10.57.00 DISPLAY OMVS 092
OMVS     000E ACTIVE          OMVS=(00,FS)
USER     JOBNAME  ASID        PID       PPID STATE   START     CT_SECS
WEBSRV   IMWEBSRV 0037   83886220          1 HK    17.38.22  10777.750
LATCHWAITPID=         0 CMD=IMWHTTPD
THREAD_ID        TCB@     PRI_JOB  USERNAME   ACC_TIME SC  STATE
0E9DD0A000000000 009E98F8 OMVS                13.704 KIN  KU
0E9DDC8000000001 009D6D90                       .004 SPM JY V
0E9DE86000000002 009E9078                       .001 SPM JY V
0E9DF44000000003 009CE288          PUBLIC       70.934 STA JY V
```

```
0E9E0C0000000004 009D0190          PUBLIC       72.890 CTW JY V
0E9E17E000000005 009CC0F0 OMVS                  78.944 ANR JF V
0E9E23C000000006 009CC288 OMVS                  67.135 ANR JF V
0E9E2FA000000007 009C80F0 OMVS                  71.946 ANR JF V
0E9E3B8000000008 009C8288 OMVS                  58.614 ACP JF V
0E9E476000000009 009CA190 OMVS                  63.214 ANR JF V
0E9E53400000000A 009C4288 OMVS                  73.834 ANR JF V
0E9E5F200000000B 009C6190 OMVS                  60.717 ANR JF V
0E9E6B000000000C 009C00F0 OMVS                  56.973 ANR JF V
0E9E76E00000000D 009C0288 OMVS                  66.594 ANR JF V
0E9E82C00000000E 009C2288          PUBLIC      130.405 CTW JY V
0E9E8EA00000000F 009BC288 OMVS                  73.102 ANR JF V
0E9E9A8000000010 009BE190 OMVS     PUBLIC      341.515 SEL JF V
0E9EA66000000011 009BA0F0 OMVS                  65.763 ACP JF V
0E9EB24000000012 009BA288 OMVS                  73.272 ANR JF V
0E9F880000000024 00990190 OMVS                  65.736 ACP JF V
0E9F93E000000025 0099E0F0 OMVS                  55.842 ANR JF V
0E9F9FC000000026 0099E288          PUBLIC       84.912 CTW JY V
0E9FABA000000027 00958288 OMVS                  67.240 ANR JF V
0E9FB78000000028 009580F0 OMVS     PUBLIC       70.286 SEL JF V
0E9FC36000000029 0098E190          PUBLIC    77309.866 CTW JR V
0E9FCF400000002A 0098C288 OMVS                  75.926 ANR JF V
0E9FDB200000002B 0098A0F0 OMVS                  71.191 RCV JF V
0E9FE7000000002C 0098A288 OMVS                  65.546 ACP JF V
0E9FF2E00000002D 00988190 OMVS                  69.821 ACP JF V
```

A dump of the Web Server will allow level 2 support to determine what the TCB in question is doing that is causing excessive CPU utilization. In the preceding example, a poorly coded application was going through excessive garbage collection.

2.7 Application level tools

There are also several application level tools that can provide specific performance data, right down to a troublesome line of code.

These tools are only discussed at a high level in this section; for more detailed information, see our companion IBM Redbook *Writing Optimized Java Applications for z/OS,* SG24-6541.

- JProbe by Sitraka
- JInsight by IBM
- Introscope by Wily technology

2.7.1 JPROBE by Sitraka

More detailed information about JProbe can be found at the following site:

`http://www.sitraka.com/software/jprobe`

JProbe is a set of graphical tools for diagnosing code problems in Java applications. JProbe helps programmers understand the potential problems in the application. JProbe provides, in graphical form, information concerning various topics from memory usage to calling relationships.

Additionally, JProbe makes it possible to analyze interactions with Java modules outside the application being analyzed, even if source code for those modules is unavailable. However, if the source code is available, JProbe allows you to pinpoint the line of code that is causing the problem. JProbe identifies problems caused by interactions with third-party components, queries to a relational database via Java database connectivity(JDBC), and calls to other distributed elements of the application through remote method invocation(RMI).

The JProbe product line includes four diagnostic tools:

JProbe Profiler with an integrated Memory Debugger

The **Profiler** identifies where time is being spent inside an application, helping developers increase the efficiency and scalability of their code.

The **Memory Debugger** tracks down memory leaks and memory mismanagement, both of which can seriously hinder scalability

JProbe Threadalyzer

This tracks down threading issues such as deadlock and data race situations.

JProbe Coverage

This ensures complete test cases by identifying which parts of an application have and have not been executed during testing.

Starting a JProbe product brings up the JProbe Console. The JProbe Console, a Java application itself, executes in a separate JVM from the program it monitors. Your Java application runs in a different, instrumented JVM.

An *instrumented* JVM is a JVM with a profiling interface. Your Java 2 JVM must fully support Sun Microsystem's JVM profiling interface, JVMPI. Different JVMs are used to ensure that the collected data is specific to your Java application.

Collected data is stored in snapshot files, saved in a temporary directory specified by the user. You can choose to rename and relocate the snapshots to a permanent location. Additionally, you can create snapshots via either an interactive session or setting triggers to turn data collection on/off automatically. The JProbe environment in a remote session is shown in the following figure.

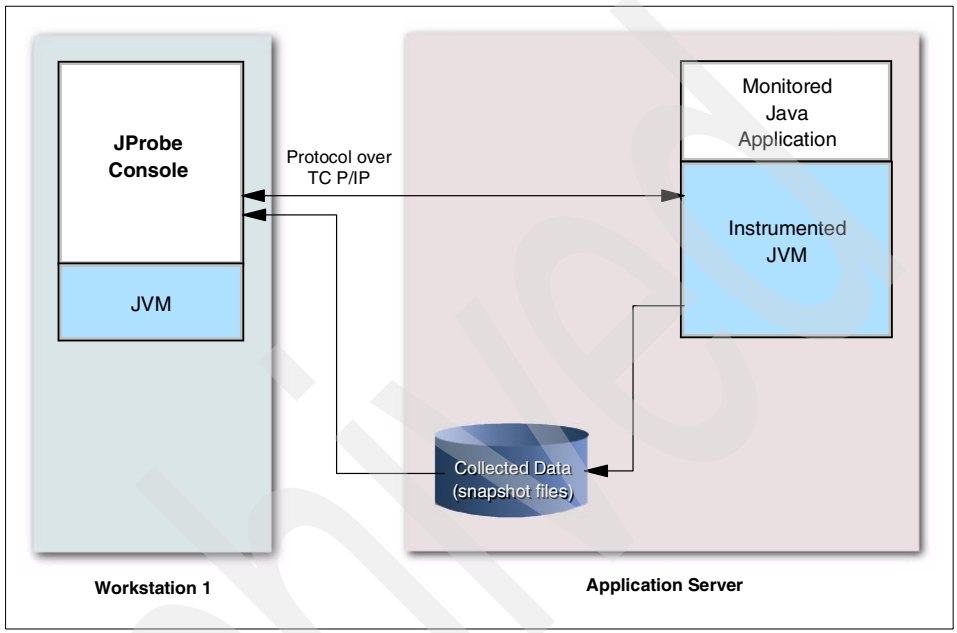


Figure 2-4 JProbe Environment in a remote session

2.7.2 JINSIGHT by IBM

More detailed information about JInsight can be found at:

http://www.alphaworks.ibm.com/tech/jinsight

Jinsight is a tool for visualizing and analyzing the execution of Java programs. It is useful for performance analysis, memory leak diagnosis, debugging, or any task in which you need to better understand what your Java program is really doing.

Jinsight brings together a range of techniques that let you explore many aspects of your program:

- **Visualization**. Visualizations let you understand object usage and garbage collection, and the sequence of activity in each thread, all from an object-oriented perspective.
- **Patterns**. Pattern visualizations extract the essential structure in repetitive calling sequences and complex data structures, letting you analyze large amounts of information in a concise form.
- **Information exploration**. You may specify filtering criteria to focus your study, or drill down from one view to another to explore details. Create your own units that precisely match features you are studying, and then use them as an additional dimension in many of the views.
- **Measurement**. Study measurements of execution activity or memory summarized at any level of detail, along call paths, and along two dimensions simultaneously.
- **Memory leak diagnosis.** Special features are provided to help you diagnose memory leaks.

To monitor the traces in your server, jinsight uses an application driven by one Servlet and an HTML page. You need to configure and install this application. The steps are the generation of a war file and then a ear file.

To create a war file for the jinsight application, you can use WebSphere Studio.

To configure and install the Jinsight application, modify the jinsight.htm and check that your deployment descriptor looks like Figure 2-5 on page 45.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE web-app (View Source for full doctype...)>
- <web-app>
    <display-name>Jinsight2.1-WebApplication</display-name>
    <welcome-file>jintrace.htm</welcome-file>
  - <servlet>
      <servlet-name>jintrace</servlet-name>
      <description>THE Jinsight servlet</description>
      <servlet-
        class>com.ibm.jinsight.tracing.TraceControlServlet</servlet-class>
    </servlet>
  - <servlet-mapping>
      <servlet-name>jintrace</servlet-name>
      <url-pattern>/jintrace</url-pattern>
    </servlet-mapping>
  </web-app>
```

Figure 2-5 Deployment descriptor of the Jinsight application

Note: After you deploy the Jinsight application and configure your J2EE server, remember to add a service directive in your httpd.conf file corresponding to the context root-uri you have chosen in the Application Assembly Tool.

2.7.3 Introscope by Wily Technology

You can find more detailed information about Wily Technology and Introscope at the following site:

`http://wilytech.com/contact.html`

Introscope works seamlessly with WebSphere Application Server and also with any standalone Java application. AutoProbe integration between Introscope and WebSphere allows automatic management of any Java application running in WebSphere without requiring any development effort.

When Java application classes are loaded by the WebSphere classloader into the JVM, Introscope AutoProbe automatically enables the Java application for performance monitoring. This allows Introscope to collect performance metrics from J2EE components, monitor interactions between components, and most importantly, identify performance bottlenecks with specific components.

Introscope can provide key live metrics on:

- Response Times
- Rates
- Counts

- JDBC call performance.
- CICS transaction performance
- MQ Series Messaging and connection performance
- File and Socket I/O performance

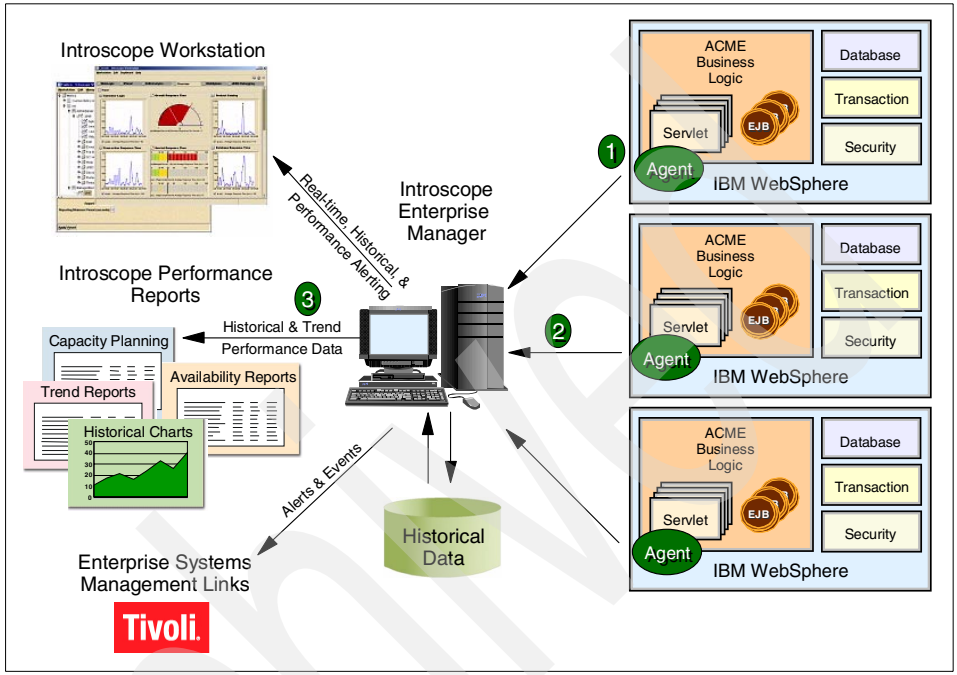


Figure 2-6 Introscope by Wily Technologies

Because of low operational overhead, it can be used for performance monitoring in live production applications.

Performance metrics can be persisted to a database and a variety of informative comparative reports can be generated, which can be used for trend analysis.

Part 2

Tuning

In this part we discuss how to tune the z/OS operating system and the WebSphere Application Server environment for Java Applications.

3

z/OS

This chapter describes the changes that can be made at a z/OS system level. Tuning your system for the WebSphere Applicaton Server assumes that your basic underlying z/OS system is already performing at acceptable service levels.

References

- *WebSphere Application Server V4.01 for z/OS and OS/390 - Operations and Administration,* SA22-7835
- *OS/MVS Initialization and Tuning Reference*, SC28-1752
- *OS/390 RMF Report Analysis*, SC28-1950
- *OS/390 MVS System Management Facilities*, GC28-1783
- *Hierarchical File System Usage Guide,* SG24-5482
- *OS/390 eNetwork Communication Server: IP User's Guide and IP Configuration Guide,* SC31-8513
- *z/OS MVS Planning: Workload Management*, SA22-7602

Web sites

http://www.ibm.com/servers/eserver/zseries/zos/unix/bpxa1tun.html
http://www.ibm.com/servers/eserver/zseries/ebusiness/perform.html
http://www.ibm.com/servers/eserver/zseries/zos/wlm/
http://www.ibm.com/software/webservers/appserv/zos_os390/library.html

3.1 Introduction

There are a number of things that can be done at a z/OS system level to help improve the performance of the WebSphere Application Server environment. The information in this chapter deals primarily with how to set up your z/OS environment for WebSphere Application Server.

Performance and tuning information for WebSphere on z/OS is detailed in the publication *WebSphere Application Server V4.01 for z/OS and OS/390 - Operations and Administration,* SA22-7835.

Performance information is subject to change. It is recommended that you always check with the latest version of the product manual. WebSphere for z/OS documentation is available on the Web at the WebSphere Application Server for z/OS and OS/390 Library Center:

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

3.2 What to check

In the following sections we discuss the areas you should particularly take a look at.

3.2.1 General storage recommendations

Virtual storage

Ensure that you do not underestimate the amount of virtual storage needed by the WebSphere Application Servers. Generally, they use significantly more virtual memory than traditional application servers on z/OS or OS/390.

The following recommendations represent the minimum storage needed to run the WebSphere Application Servers

Link pack area

We recommend you load the WebSphere Application Server run time in the link pack area (LPA) because the total size of all the load modules is large, and many address spaces need to refer to these load modules. The load modules for the run time comprise about 200 MB in size.

WebSphere runtime program objects are contained in PDSE libraries. Use the dynamic LPA option (PROGxx member in PARMLIB with `LPA,ADD`) to load the WebSphere runtime in LPA. Since modules added to the system by dynamic LPA processing are placed into CSA or ECSA storage, it is important to ensure that the system CSA and ECSA sizes are adequately defined to handle the additional storage requirements.

You should monitor ECSA after loading the WebSphere Application server run time into LPA. Remember also to increase the size of your CSA page data set accordingly.

Region size

The setting of REGION on the JCL for the startup procedures should be large (at least 384 Meg to run), and much larger if high throughput is required.

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

If you choose to place the load modules in steplib or in the link list, you must allow for an additional 200 Meg as part of each address space's region. If you choose to run from the link list, you will need a minimum of 328 Meg (200 Meg for load modules and 128 Meg for the dynamic area).

All of the WebSphere for z/OS JCL procedures are shipped with a default REGION=0M, which means they will be given as large a region as possible. Our recommendation is to run with a region size of 0 Meg; see "Region = 0 MB" on page 52 for more information.

User exits

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

If your IEFUSI exit routine limits the maximum region to a size smaller than what you need (128 Meg minimum when you run from the link pack area, or 328 Meg minimum when you run from the link list), you will get an abend.

Number of Address Spaces

A typical WebSphere for z/OS basic installation consists of at least 9 Address Spaces, each of which reference most of the 200 Meg of load modules. Keep in mind that the aggregate of all of these Address Spaces consumes a large amount of virtual and eventually real storage.

You can get an idea of the virtual storage usage through RMF or other performance monitors. We recommend that the server region procedures specify REGION=0M, which tells the operating system to give all the available region.

Real storage

From reading “Virtual storage” on page 50, you can see that the servers use a lot of virtual storage. Since real storage is needed to back the virtual storage, its usage is also high. Expect a requirement of at least 512 Meg of real storage for a small configuration. The control regions utilize around 18 Meg of that real storage.

The amount of real storage utilized in the server regions is dependent on the size of the JVM heapsize; for more information about the JVM heapsize, refer to JVM_HEAPSIZE under 12.1.4, “current.env file” on page 122.

If the HEAPSIZE is too large, there will be longer delays when garbage collection is eventually done. It also increases the amount of system resources needed for each of the application servers that can be started by Workload Manager. Since Workload Manager will start multiple application servers as the workload increases, there is no need to run with an extra large HEAPSIZE. It should only be set high enough to support the running of the largest application.

3.2.2 SYSx.PROCLIB (startup procedures)

Region = 0 MB

There are a number of procedures in SYSx.PROCLIB that get started, either in support of—or as part of—the WebSphere Application Server environment. For each of these procedures, it is best to set the region size to 0M (0 Meg). This allows each of executables to allocate the maximum amount of virtual storage, unless inhibited by the IEFUSI user exit.

The names of some of these procedures will vary in different installations. On our system they are named the following:

```
SYS1.PROCLIB(IMWEBSRV)
SYS1.PROCLIB(TCPIP)
<control regions>          <server regions>
SYS1.PROCLIB(BBODMN)
SYS1.PROCLIB(BBOSMS)       SYS1.PROCLIB(BBOSMSS)
SYS1.PROCLIB(BBONM)        SYS1.PROCLIB(BBONMS)
SYS1.PROCLIB(BBOIR)        SYS1.PROCLIB(BBOIRS)
```

See “TCP/IP” on page 79 for more information about TCP/IP setup recommendations.

Note: In addition to the above procedures, your installation will also have one or more J2EE application servers. Our recommendations are the same for each of their startup procedures

Recommendations: Set the region size to 0 MB for the above procedures in your proclib. By default, all of the WebSphere Application Server startup procedures are shipped with the region size set to 0 MB.

Any J2EE application servers that you add should also set the region size to 0 MB in both the control and server procedures.

```
See the following example for the J2EE application server Installation
Verification Procedure startup procedure:
******************************** Top of Data ************************
//BBOASR2 PROC SRVNAME='BBOASR2A',
//       PARMS=''
//  SET RELPATH='controlinfo/envfile'
//  SET CBCONFIG='/WebSphere390/CB390'
//BBOASR2 EXEC PGM=BBOCTL,REGION=0M,TIME=NOLIMIT,
// PARM='/ -ORBsrvname &SRVNAME &PARMS'
//BBOENV DD PATH='&CBCONFIG/&RELPATH/&SYSPLEX/&SRVNAME/current.env'
//CEEDUMP   DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//SYSOUT    DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//SYSPRINT  DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//
//* ============================================================== */
```

3.2.3 SYSx.PROCLIB (application servers)

LEPARMS HEAPPON

The LE Heap is the next level of storage management to be concerned with. For servers, IBM has compiled default values for HEAP and HEAPPOOL into the server main programs. These are good starting points for simple applications.

For your client programs that run on z/OS, we recommend that you at least specify HEAPP(ON) in the startup procedure of your client to get the default LE heappools.

Note: Make sure that if you use LE HEAPCHECK, you turn it off once you have verified that your code does not include any uninitialized storage.

To turn on heappools in the shell, set the environment variable _CEE_RUNOPTS:

```
export _CEE_RUNOPTS="HEAPP(ON)"
```

You can do this in a shell script which starts the application you want to run. Once the environment variable is set in a process, all child processes which are forked or spawned after that will also have this value set.

3.2.4 SMF recording

Recording of the various SMF record types is specified through the SMFPRMxx member of the system parmlib. See “Enabling SMF recording” on page 20 for more information on how to turn on/off SMF recording.

You can also refer to Appendix A, “SMFPRMxx” on page 125 for a complete description of the SMFPRMxx member in SYSx.PARMLIB.

It is important to understand that the recording of SMF data also has a performance impact. For example, in the case of Web transactions, the effect of large numbers of users causing the opening and closing of many HFS files can create a large number of type 92 SMF records and add to the system overhead.

To minimize the SMF recording overhead, the general recommendation is to turn off unnecessary SMF recording. Turn off recording of all SMF records you do not exploit.

However, in order to be proactive—and since SMF data provides valuable system and application resource usage—you should collect and analyze enough SMF data to be able to understand the performance level of your workload in order to recognize and prevent the occurrence of abnormal situations.

A compromise may be to activate SMF recording only during selected periods. Performance information can be collected periodically to help determine if resource contentions or bottlenecks are occurring.

To be able to easily control the recording of SMF, we suggest that you enable the appropriate SMF recording for your environment with the following caveats.

Accounting SMF type 30 records

These records are used for traditional TSO/E and batch applications and are not related to e-business environments. If you no longer have a need for these SMF records, it is recommended that you suppress their recording.

HFS SMF type 92 records

As stated earlier, it is recommended that you disable the recording of type 92 subtype 10 (open) and 11 (close) records, unless you have a specific need for them. For information about enabling the recording of SMF subtypes see “Enabling SMF recording” on page 20.

HTTP SMF type 103 records

Enable type 103 SMF record recording in SYSx.PARMLIB but do not turn on the HTTP Server's ability to write them unless you want to use them. For more information about the types of HTTP server records, see "SMF type 103 - HTTP Server records" on page 24.

SERVER/CONTAINER SMF type 120 records

Enable type 120 SMF record recording in SYS1.PARMLIB(SMFPRMxx) but only activate the recording when you use them. To dynamically enable/disable these records, see "SMF type 120 - Application Server/Container records" on page 26.

If you enable recording for SMF record 120, make sure that you apply on your system the fix to APAR PQ55355.

3.2.5 Component tracing

Unless directed by support personnel, you should turn off any component tracing. Issue the `D TRACE,COMP=ALL` console command to determine which traces are running on your system.

Example 3-1 Display Trace

```
D TRACE,COMP=ALL
IEE843I 11.53.52  TRACE DISPLAY 845
        SYSTEM STATUS INFORMATION
 ST=(ON,0064K,00128K) AS=ON  BR=OFF EX=ON  MT=(ON,024K)
  COMPONENT     MODE BUFFER HEAD SUBS
-------------------------------------------------------------
   SYSJES2       OFF         HEAD    3
     NO HEAD OPTIONS
-------------------------------------------------------------
 SYSRRS        MIN  0001M
    ASIDS      *NONE*
    JOBNAMES   *NONE*
    OPTIONS    COMPERR EVENTS ONLY
    WRITER     *NONE*
-------------------------------------------------------------
 SYSXES        ON   0168K  HEAD    1
    ASIDS      *NOT SUPPORTED*
    JOBNAMES   *NOT SUPPORTED*
    OPTIONS    CONNECT,CONFIG,RECOVERY
    WRITER     *NONE*
-------------------------------------------------------------
 SYSLOGR       ON   0002M
    ASIDS      *NONE*
    JOBNAMES   *NOT SUPPORTED*
    OPTIONS    CONNECT,LOGSTRM,DATASET,SERIAL,MISC,LOCBUFF,
               RECOVERY
```

```
    WRITER     *NONE*
------------------------------------------------------------
 SYSBBOSS      ON          HEAD    5
    ASIDS      *NOT SUPPORTED*
    JOBNAMES   *NOT SUPPORTED*
    OPTIONS    MINIMUM
    WRITER     BBOWTR
------------------------------------------------------------
 ... / ...
```

3.3 The System Logger

The System Logger is an z/OS component that allows an application to log data from one system, or across multiple systems across a sysplex.

Data in a log stream can be in one of two types of storage:

- *Interim storage*, where data can be accessed quickly without incurring the cost of DASD I/O.
- *DASD data log storage*, where data is hardened for long-term access, at the cost of a DASD I/O.

When the interim storage reaches a user-defined threshold, the log data is offloaded to DASD log data sets and interim storage is deleted.

There are two types of log streams: Coupling Facility logstreams, and DASD-only log streams.

In a Coupling Facility log stream, the interim storage for log data is in coupling facility list structures. In a DASD-only log stream, interim storage for log data is contained in local storage buffers on the system.

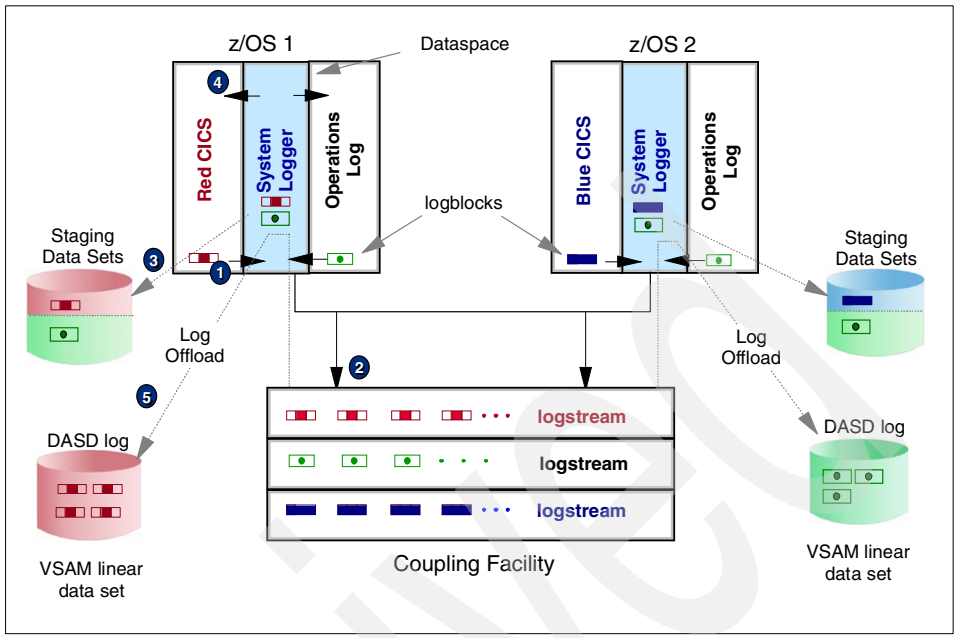


Figure 3-1 System Logger - a functional view

There is only one logger per z/OS system image. When setting up your WebSphere configuration, you may need to adjust the size of your logstreams if you already have subsystems configured to the system logger.

There is a throughput penalty for using the archive log, since it will require I/O activity. If you don't need the archive log, we recommend that you suppress it. However, it may be a good idea to use it until your application is stabilized.

For more information on RRS and the system logger, see the following documentation:

- *z/OS MVS Setting Up a Sysplex*, SA22-7625
- *z/OS MVS Programming: Resource Recovery*, SA22-7616

DASD-only logstream

When it's not possible to use CF logs, DASD-only logstream is the alternative.

Use disk devices with DASD Fast Write enabled and make sure the logs are allocated with large CI sizes. DASD Fast Write achieves high-speed performance when data is written directly into cache and non-volatile storage. This results in a faster I/O service time.

Coupling Facility logstream

In a Parallel Sysplex environment, the z/OS System Logger significantly reduces the complexity of managing multiple logs and allows a single-system view of log data. Refer to Figure 3-1 on page 57 for a functional view of the System Logger.

CF log data is always duplexed somewhere, either in staging data sets or in data space(s) associated with the Logger.

Monitoring

When the CF data is offloaded, log entries are deleted from the appropriate z/OS Logger data space(s) or staging dataset(s). Proper sizing of the logstreams is important for performance: too small, and you get reduced throughput since logger is offloading the logs too frequently; too large, and you could overflow your Coupling Facility.

You should monitor the logger to ensure that there is a sufficient size in the CF and that offloading is not impacting the overall throughput.

Use SMF type 88 records to monitor the system logger. The SMF records are written periodically and at disconnect time. SMF 88 type 1 records can help you to tune the logstream, pointing out interim storage usage and data set switches. SMF 88 type 11 records are used for structure tuning.

3.4 Set up Workload Manager for WebSphere

Workload Manager allows you to define performance goals and assign a business importance to each one. Once the definition of the goals in business terms is derived, the system decides how much resource (such as CPU and storage) should be given to meet the goal.

WebSphere Application Server V4.0.1 requires that z/OS run Workload Manager in goal mode. If your system runs in compatibility mode, you must implement goal mode. You also need to define workload management policies for WebSphere for z/OS servers and your business application servers.

For more information on Workload Manager, see *z/OS MVS Planning: Workload Management*, SA22-7602. For more information on Workload Manager and business applications, see *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Application*s, SA22-7836.

Note: For performance, verify that your system contains the fix for APAR OW48114.

3.4.1 WLM Application Environments

During the initial installation of the WebSphere Application Server environment, you need to define the WLM Application Environments for the System Management Server, Naming Server, and Interface Repository Server (you do not define an application environment for the Daemon Server). Without these definitions, WebSphere for z/OS will not start.

Note: To get started, you do not need to define special classification rules and work qualifiers, but you will want to do this for your production system.

Because the installation verification programs (IVPs) need application servers, you must also define a WLM Application Environment for the J2EE application server.

IBM provides startup procedures BBOASR2 and BBOASR1 for a J2EE and MOFW application server. Both of these types of servers are shown in the tables that follow.

Just like the WebSphere for z/OS run-time servers (with the exception of the Daemon), the J2EE servers for your business applications will have a control region and one or more server regions.

The following servers are started by the procedures, as shown in Table 3-1.

Table 3-1 Startup procedures

Server	Server name	Control Region start procedure	Server Region start procedure
Naming Server	CBNAMING	BBONM	BBONMS
System Management Server	CBSYSMGT	BBOSMS	BBOSMSS
Interface Repository Server	CBINTFRP	BBOIR	BBOIRS
J2EE application server	BBOASR2	BBOASR2	BBOASR2S

Server	Server name	Control Region start procedure	Server Region start procedure
MOFW application serve	BBOASR1	BBOASR1	BBOASR1S

For business application servers, J2EE and MOFW, you have to start the Control Regions yourself. For the WebSphere for z/OS run-time servers, however, you need only start the Daemon, which in turn starts the control regions for the System Management Server, Naming Server, and Interface Repository Server.

Workload Manager dynamically starts the server regions as work requests arrive. Thus, you must create WLM Application Environments that name server region start procedures to start, as shown in Table 3-1 on page 59. For example, specify BBOASR1S as the start procedure name that Workload Manager starts for the BBOASR1 server.

Use the ISPF application IWMARIN0 to define WLM Application Environments**.** Table 3-2 indicates default values for the run-time servers. Each new J2EE server that you create for a business application also needs to be defined to Workload Manager.

Table 3-2 Application Environment default settings

Runtime server	Application Environment (name)	Subsystem type	Procedure name	Start parameter	Server limit
Naming Server	CBNAMING	CB	BBONMS	IWMSSNM= &IWMSSNM	No limit
System Management Server	CBSYSMGT	CB	BBOSMSS	IWMSSNM= &IWMSSNM	No limit
Interface Repository	CBSYSMGT	CB	BBOIRS	IWMSSNM= &IWMSSNM	No limit
J2EE Application Server	BBOASR2	CB	BBOASR2S	IWMSSNM= &IWMSSNM	No limit
MOFW Application Server	BBOASR1	CB	BBOASR1S	IWMSSNM= &IWMSSNM	1

Control regions and SMS, Naming and IR servers should be assigned to SYSSTC or a high velocity service class. Application server regions should also be assigned a high velocity service class to get started.

The work running under the Server Regions is actually classified under the application environment using subsystem type CB. The recommendation is to use a service class with a response time goal.

When defining the classification rules, you should also define a default service class. If you do not specify one, Workload Manager will use SYSOTHER as a default, which may lead to unwanted results.

The work qualifiers supported for subsystem type CB are:

Collection Name The logical server group name defined using the CB system management utility.

Subsystem instance The CB-specific server name.

Sysplex name The sysplex name qualifier, which is useful if you have a common service definition in multiple sysplexes.

Userid The userid of the user requesting the CB service.

3.4.2 Define an Application Environment (ApplEnv)

This section shows how to invoke the Workload Manager ISPF application and define an Application Environment.

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

1. Open the main panel by issuing: `IWMARINO` in the ISPF Command Shell. This will display the WLM welcome ISPF panel shown on Figure 3-2 on page 62.

```
Command ===> ____________________________________________________________


                          W   W   L      M   M
                          W   W   L      MM MM
                          W W W   L      M M M
                          WW WW   L      M   M
                          W   W   LLLLL  M   M

                    Licensed Materials - Property of IBM

                    5647-A01 (C) Copyright IBM Corp. 2001.
                    All rights reserved.

                              ENTER to continue

  F1=Help      F2=Split     F3=Exit      F9=Swap     F10=Menu Bar F12=Cancel
```

Figure 3-2 Workload Manager initial panel

2. Either load a WLM definition from a WLM definition data set, or extract a working definition from the WLM couple data set; see Figure 3-3.

```
Command ===> ____________________________________________________________

               +-------------------------------------------+
               !        Choose Service Definition          !
               !                                           !
               ! Select one of the following options.      !
               ! 2_  1.  Read saved definition             !
               !     2.  Extract definition from WLM       !
               !         couple data set                   !
               !     3.  Create new definition             !
               !                                           !
               !  F1=Help       F2=Split      F5=KeysHelp  !
               !  F9=Swap      F12=Cancel                   !
               +-------------------------------------------+
                         ENTER to continue

F1=Help      F2=Split     F3=Exit      F9=Swap     F10=Menu Bar F12=Cancel
```

Figure 3-3 Select a Definition

Note that one needs to be authorized to access the service definitions in the WLM couple dataset. The user of IWMARIN0 must have update access to the RACF FACILITY class profile MVSADMIN.WLM.POLICY.

3. From the Definition Menu, select Application Environments, option 9, as shown in Figure 3-4.

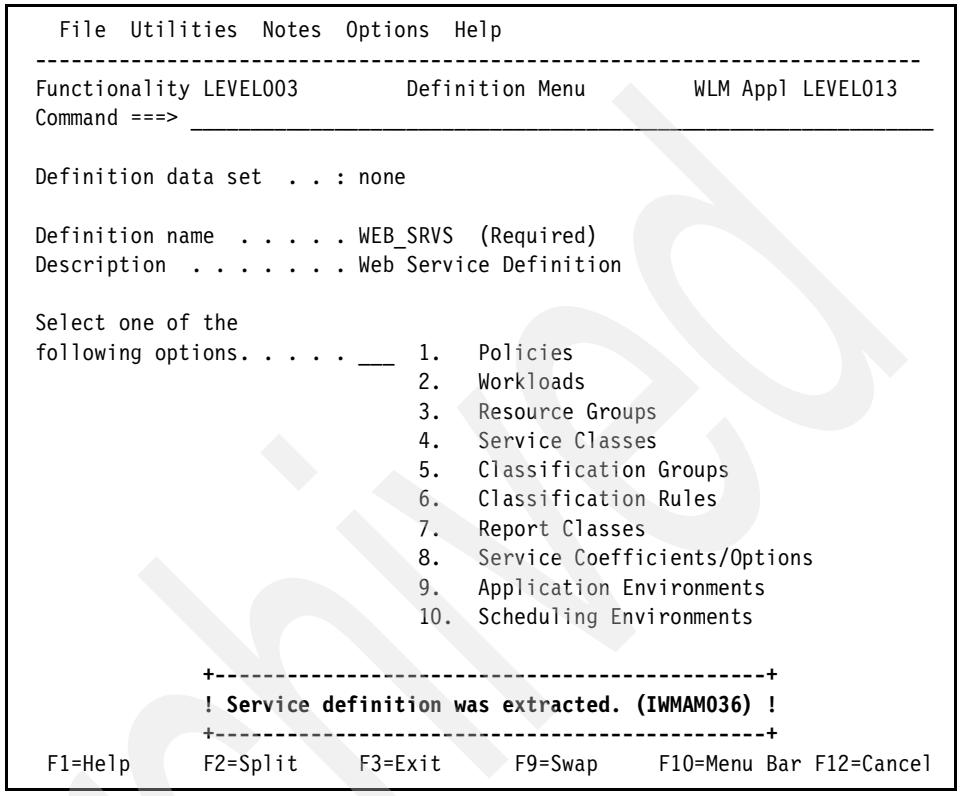


Figure 3-4 WLM definition menu

4. On the Application Environment selection list, enter 1 on any action line to create a new Application Environment (ApplEnv).
5. On the Create an Application Environment screen, fill in the appropriate fields; see Figure 3-5 on page 64.

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

Application Environment  . . . WEBAPP1_____________________  Required
Description  . . . . . . . . . Primary J2EE Application_____
Subsystem Type . . . . . . . . CB__  Required
Procedure Name . . . . . . . . WEBAPP1A
Start Parameters . . . . . . . iwmssnm=&iwmssnm_____________________
                               ___________________________________
                               ___________________________________

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


 F1=Help      F2=Split     F3=Exit      F4=Return    F7=Up        F8=Down
 F9=Swap      F10=Menu Bar F12=Cancel
```

Figure 3-5 Create new Application Environment

During the initial testing of new J2EE applications, you might want to limit the number of servers started to a single address space, to simplify the testing process. However, a setting of `No limit` allows Workload Manager to respond to the usual peaks in an e-business workload by starting additional servers and is the normal recommendation for J2EE application servers.

When the Workload Manager configuration is set to `No limit`, you can still control the maximum number of server regions by specifying values for `MAX_SRS=x` in the server's current.env file. It is recommended that you do not specify a minimum value for `MIN_SRS=` and use the system default.

6. After filling in the preceding screen, press F3 to exit. A message informs you that the Application environment was created.
7. You may also enter additional definitions for workload, service class, and classification rules for the new J2EE server and its applications. For more information, see *z/OS MVS Planning: Workload Management*, SA22-7602.

   Also, the Washington Systems Center has made additional material available on the Web, including service definitions intended to give examples of the various Workload Manager constructs, at the following site:

   `http://www.ibm.com/servers/eserver/zseries/zos/wlm/documents/documents.html`

8. When you have completed all the definitions, you must install and activate the new policy. Return to the Definition Menu screen and select **Utilities** on the menu bar. Then select **Install definition** from the drop-down menu, as shown:

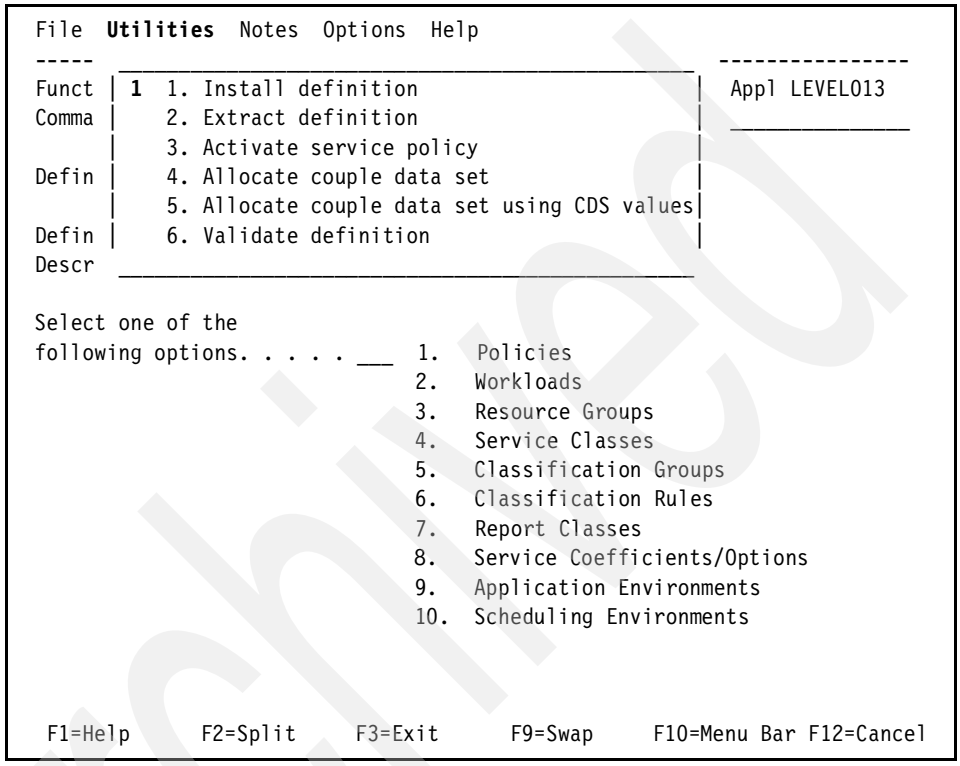


Figure 3-6 Install definition

9. The last step is to activate the new definition. Again, select **Utilities** on the menu bar, and then select **Activate service policy** from the pull-down menu.

4

UNIX Systems Services

This chapter provides information and recommendations that enable you to tune your z/OS UNIX for optimal J2EE application performance.

References

- *UNIX System Services Planning*, SC28-1890
- *UNIX Command Reference*, SC28-1892
- *WebSphere Application Server V4.0.1 for z/OS and OS/390 Installation and Customization*, GA22-7834

Web sites

http://www.ibm.com/servers/eserver/zseries/zos/unix/bpxa1tun.html
http://www.ibm.com/servers/eserver/zseries/ebusiness/perform.html

4.1 Introduction

In the zSeries environment, the HTTP Servers and J2EE Application Servers run on top of z/OS UNIX or UNIX System Services; therefore, most of your tuning will be done in the UNIX environment. UNIX literacy is a must.

z/OS UNIX has two modes:

Minimum mode If you do not specify OMVS= in the IEASYSxx member, or if you specify OMVS=DEFAULT, then kernel services start up in minimum mode when the system is IPLed. This mode is intended for installations that do not plan to use the z/OS shells, TCP/IP, or the applications that use the kernel services.

Full-function mode If you specify "FILESYSTEM TYPE(HFS)" on the OMVS= statement in the BPXPRMxx member, the kernel services start up in full-function mode when the system is IPLed. You have to set up full-function mode of z/OS UNIX to use HFS.

In order to run J2EE-compliant applications, you must be in full-function mode. To enable z/OS UNIX to run in full-function mode, you must complete these steps:

- Customizing z/OS UNIX
- z/OS UNIX Security
- Setting up z/OS UNIX Daemons

4.2 What to check

To customize z/OS UNIX, the following tasks need to be performed:

- Security preparation
- Customizing BPXPRMxx PARMLIB members
- Allocating other file systems
- Defining BPXPRMxx Parmlib members in IEASYSxx
- Customizing other PARMLIB members
- Preparing PROCLIB members
- Evaluating virtual storage needs
- Adding z/OS UNIX ISPF data sets
- Customizing the shell
- Prioritizing the kernel

- Creating HFS data sets for z/OS UNIX users
- Checking for setup errors

Note: Some of the steps might not pertain to you. Customize the relevant tasks according to your site requirements. These are explained in more detail in *WebSphere Application Server V4.01 for z/OS and OS/390 Installation and Customization,* GA22-7834.

The previous recommendations apply generally for any z/OS UNIX application; however, the following recommendations are related to Web and J2EE-compliant applications.

4.2.1 SYSx.PARMLIB(BPXPRMxx)

The BPXPRMxx PARMLIB member contains the parameters that control z/OS UNIX processing and the file systems. The system uses these values when initializing z/OS UNIX System Services.

Although not required to initially run the HTTP server, we recommend that you set the following values in the BPXPRMxx member in the PARMLIB. These are not necessarily the optimum values for your environment, but for a Web serving environment, they are more appropriate than the default values in BPXPRMxx. After the Web server is put into production, these values (and others) can be modified according to tuning requirements.

MAXFILEPROC This specifies the maximum number of files that a single process can have concurrently active or allocated.
A setting of 10000 is usually adequate for about 150 concurrent users. At higher request rates, a good approximation is the number of Web requests per second multiplied by 120 up to 65535.
Recommendation: Set MAXFILEPROC to 10000 initially.

MAXTHREADTASKS Specifies the maximum number of MVS tasks that a single process can have concurrently active for pthread_created threads.

Recommendation: Set between 1000 and 5000.

MAXTHREADS Specifies the maximum number of pthread_created threads, including running, queued, and exited but undetached, that a single process can have concurrently active.

Recommendation**:** Set MAXTHREADS to twice the value of MAXTHREADTASKS. See also Web server setting of “MAXACTIVETHREADS” on page 101.

The **SETOMVS** operator command can be used to dynamically change MAXTHREADS**,** MAXTHREADTASKS**,** and MAXFILEPROC settings. However, be sure to set these values in BPXPRMxx so that they will be retained after an IPL.

MAXSOCKETS Set at least as high as MAXFILEPROC. There is less of a performance penalty if this value is too high.

IPCSHMMPAGES The default of 256 should be sufficient when running the Web server in Standalone mode.

4.2.2 SYSx.PARMLIB(LPALSTxx)

This also applies to the Language Environment runtime. Placing Language Environment routines in the LPA/ELPA reduces the overall system storage requirement by making the routines shareable. Also, initialization and termination time is reduced for each application. For more information, see *z/OS V1R3.0 Language Environment Customization*, SA22-7564.

CEE.SCEELPA Data Set

As of OS/390 Release 6, there is a new SCEELPA data set that contains a subset of the SCEERUN modules, those that are reentrant, reside above the line, and are heavily used by z/OS UNIX System Services.

Recommendation: Place the CEE.SCEELPA data set in SYS1.PARMLIB(LPALSTxx).

4.2.3 SYSx.PARMLIB(LNKLSTxx)

This also applies to the Language Environment runtime.

CEE.SCEERUN Data Sets

Because the SCEERUN data sets has many modules that are not reentrant, you cannot place the entire data sets in the link pack area. The Language Environment modules not placed in LPA/ELPA should be placed in the link list.

Recommendation**:** Put the CEE.SCEERUN data sets in SYSx.PARMLIB(LNKLSTxx).

You can put only one version of the RTL into LNKLST and LPALST. If you need more than one version, use Run-Time Library Services (RTLS). This is described in the Language Environment Customization manual.

You may also add additional modules to the LPA using the Dynamic LPA capability (SET PROG=). This is preferable to the Modify Link Pack Area (MLPA) option, because it avoids the performance degradation that occurs with the use of MLPA.

Note: This assumes that your primary Language Environment (LE) level is the default LE level for this release. To verify your primary level, check that the library name (for example, CEE.SCEERUN) appears first in the link list concatenation in the LNKLSTxx parmlib member.

4.2.4 SYS1.PARMLIB(COFVLFxx)

Caching of the z/OS UNIX System Services UID and GID information improves response time performance because z/OS UNIX can go to cache for UID and GID instead of RACF database.

Recommendation**:** In the COFVLFxx parmlib member, make sure that VLF caching of UIDs and GIDs is specified; see Example 4-1.

Example 4-1 COFVLFxx

```
/*  Descriptive Name: Virtual Lookaside Facility (VLF)              */
CLASS NAME(CSVLLA)          /* Class name for Library Lookaside @P2C*/
      EMAJ(LLA)             /* Major name for Library Lookaside @P2C*/
CLASS NAME(IRRACEE)           /* CLASS NAME FOR RACF ACEE          */
      EMAJ(ACEE)              /* MAJOR NAME OF IRRACEE class       */
                              /* Default MAXVIRT = 4096 4K blocks  */
CLASS NAME(IRRGTS)            /* CLASS NAME FOR RACF GTS           */
      EMAJ(GTS)               /* MAJOR NAME OF IRRGTS class        */
                              /* Default MAXVIRT = 4096 4K blocks  */
CLASS NAME(IRRGMAP)           /* Open Edition GID to GROUP mapping */
      EMAJ(GMAP)              /* MAJOR NAME OF IRRGMAP class       */
                              /* Default MAXVIRT = 4096 4K blocks  */
CLASS NAME(IRRUMAP)           /* Open Edition UID to USER  mapping */
      EMAJ(UMAP)              /* MAJOR NAME OF IRRGMAP class       */
                              /* Default MAXVIRT = 4096 4K blocks  */
CLASS NAME(IRRSMAP)           /* Open Edition Security Packet      */
      EMAJ(SMAP)              /* MAJOR NAME OF IRRSMAP class       */
                              /* Default MAXVIRT = 4096 4K blocks  */
```

Make sure VLF is started. The following operator command starts VLF:

```
START VLF,SUB=MSTR,NN=xx
```

5

Hierarchical File System

This chapter contains information and recommendations that allow you to tune your HFS for optimal use by your J2EE applications running via WebSphere on z/OS.

References

- *z/OS UNIX System Services Command Reference*, SA22-7802
- *Hierarchical File System Usage Guide*, SG24-5482

Web sites

http://www.ibm.com/servers/eserver/zseries/zos/bkserv
http://www.asg.com/response/hfs_performance

5.1 Introduction

Where you place your application files in the HFS can make a difference as to how well your e-business application performs. Prior planning during the development of an application, so that the directory structure that it uses is as simple and uncomplicated as possible, can have a tremendous impact.

It also pays to mount those files on DASD that is relatively free from contention from batch and other online systems.

5.2 What to check

Do not share the HFS unless you need to. There is a performance penalty for sharing the HFS.

Most other recommendations have to do with where you place your files within the HFS.

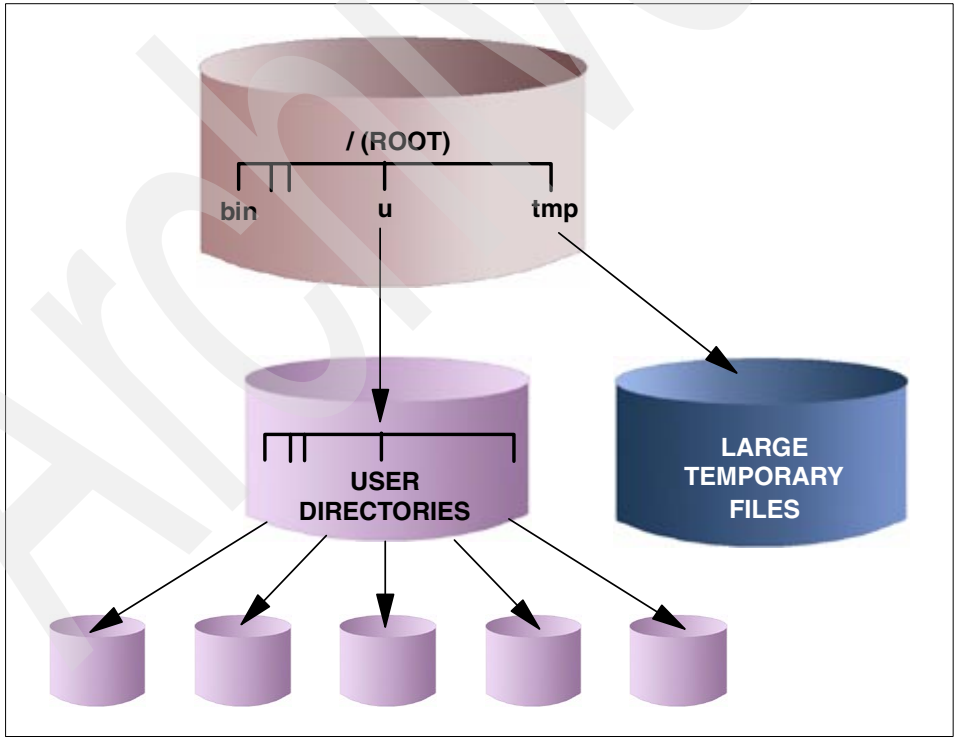


Figure 5-1 HFS organization

To reduce disk contention, define separate HFS files for users or groups. When installing new products or applications, define separate file system structures for each of them and make use of /tmp on a TFS, if possible.

If you have many HFS users, /tmp may need its own pack. Consider control unit caching with DASD Fast Write.

Example 5-1 An HFS structure

```
/bin
/samples
/SYSTEM/DEV/
       /ETC/httpd.conf
           /httpd.envvars
           /inetd.conf
           /inetd.pid
           /profile
       /TMP/
       /VAR/gateways
           /hosts
           /inetd.conf
           /resolv.conf
/u
/usr/include/
    /lib/
    /lpp/ldap/etc/slapd.conf
                 /slapd.envvars
     ...
/WebSphere390/CB390/apps
                  /controlinfo/envfile/PLEX58/BBOASR2A/
                                             /DAEMON01/
                                             /INTFRP01/
                                             /NAMING01/
                                             /PDKSRVA /backup/
                                                      /current.env
                                                      /jvm.properties
                                                      /trace.dat
                                                      /webcontainer.conf
                                             /SYSMGT01/
```

5.2.1 Dataset placement

The following are recommendations for improving performance by controlling where you place the HFS files.

- Put key HFS files on cached DASD with DASD Fast Write, because you'll get faster I/O and less contention.

- SMS 1.5 HFS enhancement is the recommended base level. With 1.5, the writes can be asynchronous.
- Avoid multiple extents in HFS datasets by consolidating to one extent using DFSMS dss.
- Spread high activity HFS datasets across multiple volumes to keep user HFS datasets separate from system HFS datasets.

5.2.2 Application HFS Files

Because an HFS is traversed in an hierarchical manner, permissions have to be checked on each directory level en route to a file. Note the following recommendations:

- Place application files as close to the root directory as possible.
- Avoid collecting unnecessary statistical data about HFS files. Be selective when collecting SMF statistics and log data.

5.2.3 GRS vs WRITEPROTECT

WRITEPROTECT (default is on) - protects against two systems, that are not in a Global Resource Serialization (GRS) complex, from both mounting a file system for read/write access.

If GRS is used, specify NOWRITEPROTECT when mounting a file system. Using GRS will save I/Os.

Use the **MOUNT** z/OS UNIX Services command to specify WRITEPROTECT or NOWRITEPROTECT; see Figure 5-2.

```
MOUNT FILESYSTEM(file_system_name)
   MOUNTPOINT(pathname)
   TYPE(file_system_type)
   MODE(RDWR|READ)
   PARM(parameter_string)
   SETUID|NOSETUID
   WAIT|NOWAIT
   SECURITY|NOSECURITY
   SYSNAME (sysname)
   AUTOMOVE | NOAUTOMOVE
```

Figure 5-2 UNIX System Service Mount command

5.2.4 Buffer pool parameters

The parameters depend on the coexistence of other database systems.

- For systems dedicated to HFS usage, we recommend FIXED = VIRTUAL and be equal to 50% of real storage capacity. Use **confighfs**.

Note: To change the Buffer Pool parameters, you need to use the **confighfs** z/OS UNIX Services command. The format is as follows:

confighfs [–l][–v n][–f n][–q][pathname][–x[n] size pathname]

Options:

–**l** Query HFS limits.

–**v n** Set virtual storage max to n (where n is in MB). Requires superuser authority.

–**f n** Set fixed storage min to n (where n is in MB). Requires superuser authority.

–**q** Query your global statistics.

pathname Query file system statistics for the file system containing each of the path names specified.

–**x** size pathname Extend the specified file system, where size is the amount to be extended suffixed by the extend unit of M, T, or C (for megabytes, tracks, or cylinders), and the pathname is a full or simple pathname to a file or directory in the file system to extend. Requires superuser authority.

–**xn** size pathname Extend the specified file system to a new volume with the same rules as above. Requires superuser authority. The following are internal debug options:

–**dn** Prints incoming and outgoing pfsctl buffers (where n is 0, 1, or 2).

–**t** Skips issuing the pfsctl.

On systems running shared HFS, this command should only be issued on the server system (file system owner) for the file system pointed to by the pathname. Issuing it on client systems results in fields of zeros. See *z/OS UNIX System Services Planning* for more information on HFS in a sysplex.

- To set virtual and fixed HFS buffer limits:

  ```
  confighfs –v 128 –f 32
  ```

- To extend the file system for your current directory 100 cylinders:

  ```
  confighfs –x 100c .
  ```

- If you need to get stats for the root file system and the file system mounted over /tmp, you would do the following:

  ```
  confighfs / /tmp .
  ```

Note: The period (.) in the preceding examples indicates the current directory.

6

TCP/IP

This chapter provides information and recommendations that will enable your Web- and J2EE-compliant applications to have better response time and system utilization.

References

- *IBM TCP/IP Performance Tuning Guide*, SC31-7188
- *MVS TCP/IP User's Guide,* SC31-7136
- *z/OS UNIX System Services Planning*, GA22-7800
- *z/OS Communications Server: IP Configuration Reference*, SC31-8776

6.1 Introduction

TCP/IP provides the transport for all HTTP requests, so validating the parameters for TCP/IP will ensure that the settings support transaction activity. TCP/IP, when not properly configured, can be the source of some significant response time delays. In a J2EE environment, this would be critical to remote method calls.

6.2 What to check

- First, ensure that you have defined enough sockets to your system and that the default socket time-out of 180 seconds is not too high.
- Next, check the specification of the port in the TCPIP profile dataset to ensure that `NODELAYACKS` is specified as follows:

  ```
  PORT 8082 TCP NODELAYACKS
  ```

  Changing this could improve throughput by as much as 50%. This is particularly useful when dealing with trivial workloads.
- Ensure that your DNS configuration is optimized so that lookups for frequently-used servers and clients are being cached. Caching is sometimes related to the name server's Time To Live (TTL) value.

  On one hand, setting the TTL high will ensure good cache hits. However, setting it high also means that, if the daemon goes down, it will take a while for everyone in the network to be aware of it.

  You can monitor your performance via RMF and SMF (record 118).

For more information about multiple TCP/IP stacks (Common INET), see *z/OS UNIX System Services Planning*, GA22-7800.

If you are running both legacy and Web applications on your system, make the TCP/IP z/OS dispatching priority and VTAM the same setting. Make buffer pools large enough to support traffic volumes. Use the TCP/IP shutdown statistics to estimate and tune the number and size of buffers. Buffer pool parameters are documented in the *TCP/IP User's Guide*.

6.2.1 HiperSockets

The zSeries HiperSockets function, also known as internal queued direct input/output (IQDIO), operates with minimal system and network overhead. HiperSockets eliminates the need to utilize I/O subsystem operations and the need to traverse an external network connection to communicate between LPARs. HiperSockets offers significant value in server consolidation by connecting many virtual servers.

Within a sysplex environment, HiperSockets can also be used instead of XCF links for IP traffic.

6.2.2 SYSx.PROCLIB(TCPIP)

Region size

The startup procedure for TCP/IP is usually located in SYSx.PROCLIB.

Recommendation Set TCP/IP region size to REGION=0M

```
//TCPIP PROC P1='CTRACE(CTIEZB00)',TCPPROF=TCPPROF,TCPDATA=TCPDATA
//*
//TCPIP EXEC PGM=EZBTCPIP,REGION=0M,TIME=1440,
//   PARM=&P1
```

This allows TCP/IP to allocate the maximum virtual storage above and below the 16 MB line.Check IEFUSI exit for restrictions to specifying the REGION parameter.

See “SYSx.PROCLIB (startup procedures)” on page 52 for information about other startup procedures that should run with a region size of 0 meg.

6.2.3 TCPIP.PROFILE

There are two values in the tcpip.profile dataset that can be changed in order to lead to performance enhancements. See Appendix C, “TCPIP profile” on page 133 for information on how to locate the TCPIP.PROFILE dataset.

DELAYACH This delays the transmission of Acks so they can be combined with other data returned to the client. The default is OFF.

Recommendation: Set DELAYACH ON. This change will enhance a system serving trivial servlets. This will cause a slight performance hit with more complex servlets.

SHAREPORT The default is OFF.

Recommendation: Leave SHAREPORT OFF.

6.2.4 HTTPD.CONF

DNS-Lookup Turning DNS-Lookup on decreases server performance. The default is OFF.

Recommendation: Keep DNS-Lookup OFF.

If you cannot disable DNS-Lookup, do the following:

- Language Environment (LE) has DNS cache use httpd.envvars: _EDC_IP_CACHE_ENTRIES=nn

Example 6-1 httpd.conf

```
   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
_______________________________________________________________________________
EDIT       /etc/httpd.conf                                 Columns 00001 00072
Command ===>                                                 Scroll ===> CSR
000101
000102 #        DNS-Lookup directive:
000103 #
000104 #        Instruct the server to look up hostnames of clients by
000105 #        setting DNS-Lookup to "on".
000106 #        NOTE: Turning DNS-Lookup "on" decreases server performance.
000107 #
000108 #        Default:  off
000109 #        Syntax:   DNS-Lookup <on | off>
000110 DNS-Lookup     off
000111
```

6.2.5 Advanced TCP/IP network

Advanced TCP/IP configurations include:

- The use of multiple TCP/IP stacks on OS/390 or z/OS. Note that there will be a performance penalty.
- Connection optimization, an OS/390 or z/OS function by which workload management and the DNS cooperate to route requests
- The IBM Network Dispatcher, which is a network router
- Bind-specific support, which allows you to control the use of TCP/IP resources in Multiple TCP/IP stacks

You may want to run multiple TCP/IP stacks on the same system to reduce the chances of having a single point of failure. For instance, you may have multiple OSA Features connecting your z/OS system to the network and want to assign a TCP/IP stack to each one; to do so, use the common INET physical file system (C_INET PFS). This physical file system allows multiple physical file systems (network sockets) to be configured and active concurrently.

Specify common INET through the NETWORK DOMAINNAME parameter of SYS1.PARMLIB(BPXPRMxx). For details, see *z/OS UNIX System Services Planning* and *z/OS Communications Server: IP Configuration Reference*, SC31-8776.

Connection optimization

This refers to a system configuration in which the Domain Name Server cooperates with Workload Manager (WLM) to route client requests throughout a sysplex. Characteristics of this configuration are:

- The domain name server (DNS) is replicated by setting up a secondary DNS on more than one system in the sysplex.
- The client needs to know the Daemon IP Name in order to connect to WebSphere for z/OS.

Each system in the sysplex has the same Daemon IP Name and Resolve IP Name. Workload Manager and the Domain Name Server determine the actual system to which client requests go. The client sees the sysplex as a single system, though its requests may be balanced across systems in the sysplex.

As part of workload balancing and maximizing performance goals, Workload Manager also routes work requests to systems in the sysplex. This function is possible because WebSphere for z/OS cooperates with Workload Manager, because the system references that a client sees are indirect; even requests from that same client may be answered by differing systems in the sysplex.

The implication for clients is that they should not cache IP addresses unless they can recover from failed connections. That is, if a connection fails, a client should be able to reissue a request, but because the IP address is an indirect address, a reissue of the request can be answered by another system in the sysplex.

Bind-specific support in WebSphere for z/OS

Bind-specific support in WebSphere for z/OS allows you to control the use of TCP/IP resources in WebSphere for z/OS. This support allows you to have the WebSphere for z/OS ORB and other products and applications on the same OS/390 or z/OS system without requiring the client code to configure unique ports.

In other words, this support allows use of port 900 by WebSphere for z/OS and other products and applications on the same system. This support allows the utilization of multiple TCP/IP stacks (Common INET) by the WebSphere for z/OS ORB, and the use of multiple IP addresses on the same TCP/IP stack.

To use bind-specific support, use the SRVIPADDR environment variable, which specifies the IP address in dotted decimal format. WebSphere for z/OS servers listen for client connection requests on this IP address. Because a given IP address is associated with a given TCP/IP stack, you could specify the SRVIPADDR variable in the environment file so that a WebSphere for z/OS server uses a specific TCP/IP stack.

Note: SRVIPADDR is found in the configuration.env or current.env files in the HFS. *CBCONFIG*/controlinfo/envfile/*SYSPLEX*/*SRVNAME*/current.env

You can edit the configuration .env file directly only *before* the bootstrap is completed. After bootstrap, you *must* use the Administration application to modify environment variables.

In addition, because you can define multiple IP addresses for a given TCP/IP stack, WebSphere for z/OS port 900 servers could share the same TCP/IP stack with other products and applications requiring port 900, because you made their IP addresses unique with SRVIPADDR.

Alternatively, you can, without the use of bind-specific support, define alternate ports for port 900 and the daemon, which are the only values defined by the CORBA standard. However, it is not clear that all client ORBs will easily support configuring the bootstrap port to something other than 900. Configure the ports for the daemon and system management server by specifying port numbers on the DAEMON_PORT and RESOLVE_PORT environment variables.

For more information about multiple IP addresses on the same TCP/IP stack, see *z/OS Communications Server: IP Configuration Reference,* SC31-8776.

7

LDAP

This chapter provides recommendations and guidelines on improving the performance of the LDAP component on z/OS.

References

- *Understanding LDAP*, SG24-4986
- *z/OS V1R2.0 SecureWay Security Server LDAP Server Administration and Usage Guide*, SC24-5923

Web sites

http://www.research.ibm.com/journal/sj/392/shi.html
http://www.ibm.com/servers/eserver/zseries/zos/security/ldap.html

7.1 Introduction

The z/OS Lightweight Directory Access Protocol (LDAP) provides a lightweight frontend to an X.500 directory or stand-alone directory service consisting of uniquely named entries organized in a tree structure. Entries are composed of attributes, types, and values which determine results of directory operations provided to the client including authentication, information search and retrieval, information modification, entry addition, and entry deletion.

The LDAP server can have either a Relational Database Manager (RDBM) or Transaction Database Manager (TDBM) backend. The TDBM is the preferred backend because it has all the recent focus on enhancements for both functionality and performance. Figure 7-1 shows an example of the LDAP tree structure, with each box representing a directory entry.

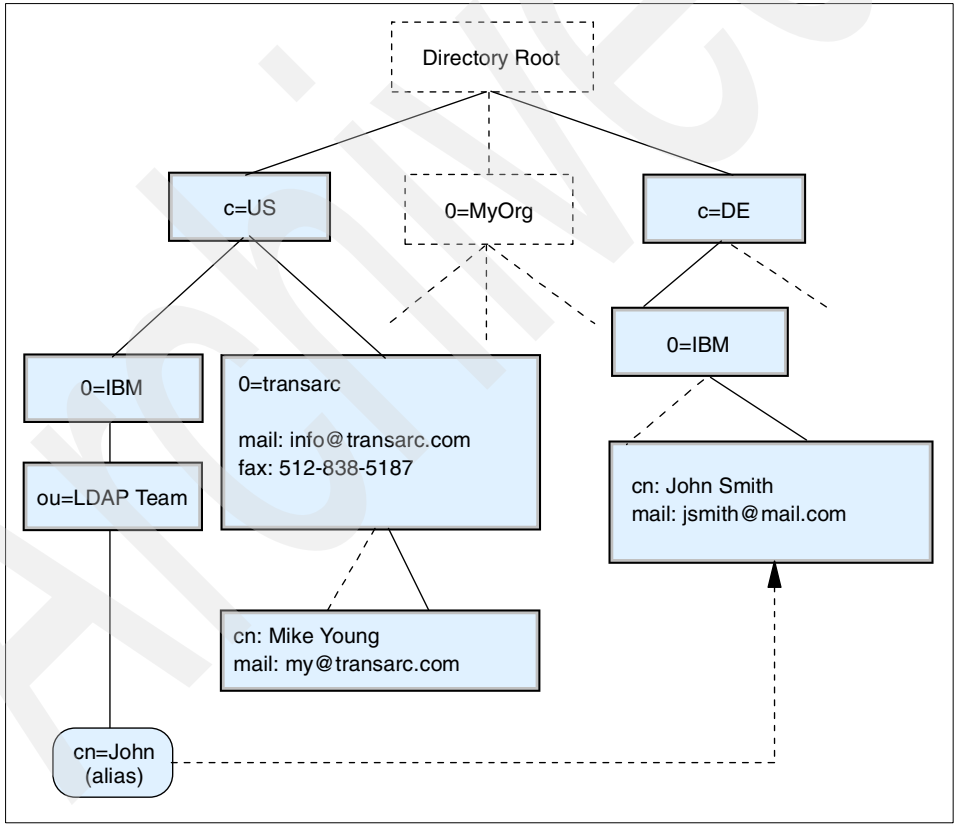


Figure 7-1 Example of a Directory Information Tree

7.2 What to check

There is very little general performance guidance when it comes to LDAP. Much of the tuning needed would be in DB2, or would involve TDBM database choices related to DB2 performance, and there is a wealth of information on that subject.

Note: The `maxThreads` and `waitingThreads` options are no longer evaluated by the LDAP server and should be removed from the configuration file prior to running the LDAP server. Both of these options have been deprecated by the `commThreads` option that is now available.

Use TDBM for your LDAP server, as opposed to RDBM. The TDBM backend improves the performance of your LDAP server, and IBM has announced that RDBM will not be supported after z/OS V1R3.

A customization dialog is available to convert from RDBM to TDBM. Follow the dialog to allocate target data sets, define variables, generate the customization jobs, and view the migration instructions. The procedure is detailed in *WebSphere Application Server V4.0.1 for z/OS and OS/390 Installation and Customization,* GA22-7834.

7.2.1 slapd.conf file

See Appendix D, "slapd.conf" on page 139 for information about how to locate the slapd.conf file. Note the following:

commThreads

The `commThreads` option is used to specify the number of threads created at LDAP server initialization. Previously, `maxConnections` and `maxThreads` had to be the same. This is no longer a requirement with `maxConnections` and `commThreads`.

Recommendation**:** It is recommended that `commThreads` be set to approximately two times the number of CPUs that are running in your LPAR. However, this is a general rule depending upon the IO activity that your LDAP server experiences.

maxConnections

This refers to the maximum number of concurrent client connections that the LDAP server should accept.

In z/OS Release 2, the `maxConnections` option should be set to the expected number of concurrently connected clients that the LDAP server will allow. Previously, this option was limited by **maxThread**s, which has been deprecated in z/OS Release 2.

Prior to z/OS Release 2, both `maxConnections` and `maxThreads` were strongly suggested to be the same value.

Recommendation**:** The value specified for this option should take into account that the server uses approximately 10 descriptors for internal functions, and will use more, depending upon the number of additional sockets used as passive sockets for connection attempts by clients.

The maximum number of client connections is further restricted by the following:

- The maximum number of files a single process can have concurrently active

  The `MAXFILEPROC` statement in BPXPRMxx and the `maxfileproc` option on the RACF **`altuser`** command are used to set the limit. Only processes with superuser authority can adjust the limit beyond the limit specified by MAXFILEPROC. Attempts to exceed this limit by non-superuser processes may be audited by the security manager.

- The maximum number of sockets allowed by the TCP/IP socket file system

  The `MAXSOCKETS` option on the NETWORK statement for BPXPRMxx sets this limit. Setting these limits too high can affect system performance by using too many resources and deprive other functions of their share of the same resources.

When migrating from a previous version of the configuration file, remove the **maxThreads** option and add ten to the current value of `maxConnections`.

7.2.2 JNDI name and LDAP

The JNDI name maps into a tree-like data structure in LDAP. The more levels there are in the JNDI name, the deeper the tree structure in LDAP.

It is required that a JNDI name for an object be unique in the name space. This is often accomplished by including things like package names in the JNDI name. This generally produces a unique JNDI name—but will also produce a deep LDAP.

Recommendation: Keep the tree structure as flat as possible. The deeper the levels, the more times LDAP will have to traverse the tree to reach the reference.

7.2.3 Fast Write DASD

Fast Write DASD achieves high-speed performance when data is written directly into cache and non-volatile storage. It allows an IO to complete before the data is actually written to the DASD volume.This results in a faster IO service for the jobs using the dataset.

Recommendation: Make sure that the DB2 database tables used by the LDAP Server are on disks with DASD Fast Write enabled.

7.2.4 Multiple LDAPs

If possible, keep the WebSphere Application Server for z/OS LDAP tables separate from other LDAP tables. Why? Because you need to back up the WebSphere z/OS LDAP with the WebSphere for z/OS system management database as a unit, and performing such a coordinated backup is easier if the WebSphere for z/OS LDAP tables are separate from the other LDAP tables.

Additionally, if you need to restore the WebSphere for z/OS environment, restoring the WebSphere for z/OS LDAP tables will not interfere with LDAP tables used by other applications.

LDAP server instances operating in a multi-server mode, either with or without dynamic workload management enabled, will not perform replication even if replication objects are present in the RDBM database.

7.2.5 DB2 active logs

DB2 records all data changes and other significant events in logs. By having this record of changes, DB2 can recreate those changes in the event of a failure. DB2 writes each log record to a disk (VSAM dataset) called the active log. These logs contain transactions which have not yet been committed or rolled back.

Recommendation**:** Check the size of your DB2 for z/OS logs. They may need to be larger because of the number of transactions WebSphere for z/OS generates.

7.2.6 BP32K buffer pool

Buffer pools, also known as virtual buffer pools, are areas of virtual storage in which DB2 temporarily stores pages of table spaces or indexes. Buffer pools reside in the DBM1 address space (PRIMARY). BP32K is a buffer pool name in DB2 for 32 K logical page-size tables.

Recommendation: Increase the BP32K buffer pool to at least 100. This is done by entering the command **ALTER BUFFERPOOL (BP32K) HPSIZE (100**) on the DB2 Commands panel. To get to the DB2 Commands panel, do the following:

- Enter the DB2I option for your installation on the ISPF PRIMARY OPTION MENU screen.
- Enter the DB2 COMMANDS option on the DB2I PRIMARY OPTION MENU screen.
- Enter **ALTER BUFFERPOOL (BP32K) HPSIZE (100)** on the command line.

7.2.7 Additional guidelines for DB2 and LDAP

1. Some direct LDAP tuning is possible in the initial TDBM database/table/index creation via values specified in the SPUFI file. This is discussed in SPUFI file comments included in Appendix 1.3.1 in *z/OS V1R2 SecureWay Security Server LDAP Server Administration,* SC24-5923.

   In particular, the buffer pool selection controls the page size (see item 3) of the given table spaces. Also, there are a couple of indexed columns whose size may be tailored, depending on the characteristics of the installation.

2. Make sure the database stays tuned via periodic maintenance with the REORG and RUNSTATS utilities.

3. For buffer pools, the general DB2 recommendation is to put indexes and tables in separate buffer pools. Tuning buffer pool sizes for the LDAP database also becomes important, especially for search performance. If the entire database can fit in the bufferpools (and paging is not a problem), I/O can be avoided. This may not be realistic for many customer databases, though, since many of them are looking at very large databases. Therefore, they should probably monitor the performance of DB2 (with DB2 performance monitor or comparable products) when they use LDAP with TDBM and make adjustments accordingly.

4. As previously mentioned, the buffer pool choices (essentially the page size) of the various tablespaces can influence performance. SPUFI generally recommends 4 K, but does allow the customer to alter it. At this point we don't yet have the experience to say that 4 K is always the best (and it probably isn't always the best).

   However, it's conceivable that certain customer data would be better suited to larger page sizes; for example, if your average entry table row size is 2200 bytes, you'll only get one entry per 4 K page. It might be preferable to increase this page size to get better space utilization, which in turn could decrease buffer pool demands and I/O rates. There are no benchmarks on this to date, so authoritative results are not guaranteed, but it seems like a good candidate for consideration and eventual experimentation.

5. Consider compression with the LDAP entry tablespace and the LDAP search tablespace. This is not necessarily a clear decision. Processor utilization will probably increase, but some response times might improve, particularly if table scans occur frequently on some complex searches.

8

Security

This chapter provides recommendations and guidelines on improving performance of the security components on z/OS.

References

- *e-business Enablement Cookbook for z/OS, Volume II,* SG24-5981

Web sites

http://www.ibm.com/security/products/prod_s390pcicc.shtml
http://publibz.boulder.ibm.com/epubs/pdf/gska1a10.pdf
http://www.ibm.com/servers/eserver/zseries/zos/unix/perform/webtun.html
http://www.research.ibm.com/journal/sj/403/guski.html

8.1 Introduction

As the Internet becomes the basis for electronic commerce, and as more businesses automate their data-processing operations, the potential for unauthorized disclosure of sensitive data increases. Online databases are becoming increasingly large and complex. Sensitive data is transmitted on communication lines, and often stored offline.

As a result, the efficient, economical protection of enterprise-critical information has become increasingly important in many diverse application environments.

Figure 8-1 shows users interacting with enterprise server via various communication protocols, including TCP/IP, RACF and others.

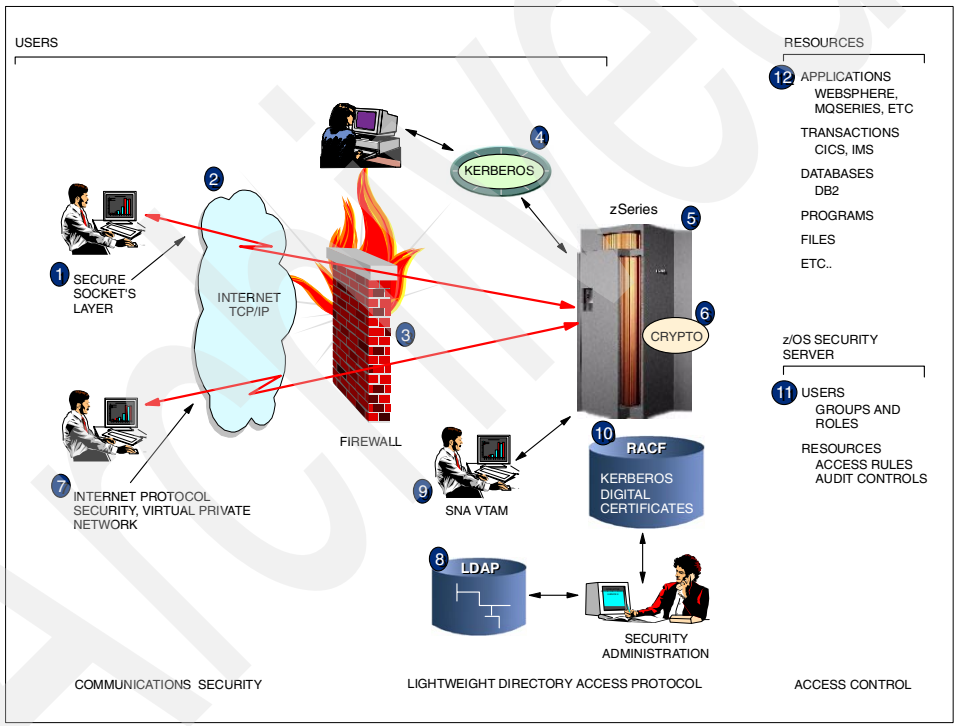


Figure 8-1 z/OS security overview

8.2 What to check

The following areas should be checked.

8.2.1 RACF

The operating system security service is known as the System Authorization Facility (SAF). All security requests go through this facility. The call may be directed to RACF, or to a user-supplied security manager, or to both. This is the mechanism by which security managers provided by other vendors are implemented on z/OS.

Recommendation: Security as a performance price, do not enable any security feature that you do not need. For example, do not enable the EJBROLE in RACF if you do not need EJBROLES.

RACLIST

If you use the following items, put them into memory via the RACLIST operand:

- ACEE
- GTS
- UID/GID
- CBIND
- EJBROLE

Recommendation: Using RACLIST with the **SETROPTS** command improves performance by copying generic and discrete profiles for the designation general-resource class.

The general format is **SETROPTS RACLIST (classname) REFRESH,** that is:

```
SETROPTS RACLIST (EJBROLE) GENERIC (EJBROLE) REFRESH
```

Surrogate user IDS

If you use %%SAF%% with the PUBLIC surrogate user ID, this provides protection of your server resources without a high CPU cost.

If you require Web clients to have a unique SAF-based user ID and password on the Web server system, specifying %%SAF%% with %%CLIENT%% will give you more robust security but at a higher CPU cost.

Recommendation**:** Use surrogate user IDs, if that is acceptable from a security point of view.

DASD Fast Write

Fast Write DASD achieves high-speed performance when data is written directly into cache and non-volatile storage. It allows an IO to complete before the data is actually written to the DASD volume.This results in a faster IO service for the jobs using the dataset.

Recommendation: Put your RACF data set on a control unit with caching and the DASD Fast Write feature. Instructions for enabling caching are in the documentation for DFSMS.

Clean up the RACF database periodically.

8.2.2 Secure Socket Layer (SSL)

SSL is a communications protocol that provides secure communications over an open communications network (for example, the Internet). The SSL protocol is a layered protocol that is intended to be used on top of a reliable transport, such as Transmission Control Protocol (TCP/IP). SSL provides data privacy and integrity, as well as server and client authentication based on public key certificates.

Once an SSL connection is established between a client and server, data communications between client and server are transparent to the encryption and the integrity added by the SSL protocol.

SSL protocol begins with a “handshake”. During the handshake, the client authenticates the server, the server optionally authenticates the clients, and the client and server agree on how to encrypt and decrypt information. In addition to the handshake, SSL defines the format used to transmit encrypted data.

The System SSL is part of the Cryptographic Services element of z/OS. It repackages the SSL function in a set of DLLs and makes the SSL APIs available for use by z/OS applications.

Recommendation**:** When designing security applications such as logon JSPs, spend as little time as possible in SSL mode, since it is costly in terms of processor usage.

8.2.3 SSL PCI Cryptographic Coprocessor

The zSeries hardware has a cryptographic hardware engine that is supported by z/OS called Peripheral Component Interface Cryptographic Coprocessor (PCICC) that provides the capability for rapid response to customer requirements. PCICC is flexible in that it makes it simple to add new functions that execute within the secure hardware boundary, and keeps pace with the exponentially growing demands of SSL.

The cryptographic coprocessors are used by the system SSL component of z/OS in the phases of the SSL session, as follows:

- The handshake phase, where asymmetric cryptography (RSA in the case of z/OS) is used

During this phase, the SSL client that has received the server's RSA public key in the server's certificate encrypts with this key a random number that will be used as a key for symmetric encryption during the data transfer phase. This random number is thus passed protected by this asymmetric encryption to the server. The server then gets the clear value of the random number by decrypting what has been passed to it, using its RSA private key.

This operation is extremely costly from the computing standpoint and should be the task of the cryptographic coprocessors. Otherwise, it would be performed by software only and will consume a large amount of MIPS.

- In the data transfer phase, both the SSL client and server switch to symmetric encryption at the end of the handshake. They will be using the random number that was previously mentioned as the encryption key.

  They both have a choice of encryption algorithms to use during this data transfer phase, as per parameters set up in the client and the server. If they use Data Encryption Standard (DES) or Triple DES (which can be slower than DES but billions of times more secure than DES, if used properly), then the encryption workload can still be offloaded to the coprocessors at the z/OS server.

  Other possible encryption algorithms in this phase are RC2 (a 64-bit symmetric block cipher) and RC4 (a symmetric stream cipher). If either is selected, then the encryption/decryption will be performed in the z/OS server by software only (however, note that the symmetric encryption consumes far less resources than asymmetric encryption).

Recommendation: You can install up to eight PCICC features in a single server to provide increased cryptographic processing capacity as you expand your usage of e-business applications requiring cryptographic processing. The IBM PCICC feature coexists with and augments the IBM CMOS Cryptographic Coprocessor. The recommendation is that if your processor supports hardware cryptography, it should be enabled. This can reduce CPU time required for SSL requests by up to 80%.

If you are not sure whether you are using hardware cryptography, you can edit the HTTP server start up proc or the httpd.envars file and add: `GSK_SSL_HW_DETECT_MESSAGE = 1`. As a result, you will get the information message: `Crypto hardware is/is not being used by SSL on initialization`. The message can be found in the system log.

SSL will encrypt every object in your page, so another thing you can do to improve performance is to avoid large objects inside the SSL pages.

9

The IBM HTTP Server

This chapter describes the changes that can be made to the IBM HTTP Server for optimal performance in a J2EE environment.

References

- *z/OS HTTP Server Planning, Installing and Using Version 5.3*, SC34-4826
- *z/OS UNIX System Services Planning,* GA22-7800

Web sites

http://www.ibm.com/servers/eserver/zseries/zos/unix/perform/webtun.html

9.1 Introduction

Note: HTTP Server performance tuning for J2EE applications does not automatically mean performance tuning for HTML, JSPs, Servlets, and so on. To monitor resource usage, you can use z/OS Display commands, the usage monitor and the log analyzer.

9.1.1 General comments

Note the following, for optimal performance:

- Have -notrace enabled on in the http server start procedure.
- Follow z/OS UNIX tuning guidelines, especially those relating to placing modules in LPA. Configure your web server libraries in the linklist or LPA and avoid STEPLIBs.
- Make sure that the Web Server is non-swappable.
- Configure your HFS and RACF data sets on cached DASD with the DASD fast write feature enabled. Closely monitor the HFS containing the logs.
- Do not have other SERVICE plugins enabled.
- Set LE parms all31on, libs(1k,,free), stor(,,,8) if a storage shortage occurs during the create thread.
- Following is a useful starting point for IHS 5.3 with WAS 3.5:
  - _CEE_RUNOPTS=AN(2M,512K,ANY,KEEP),
  - HEAP(8M,2M,ANY,KEEP),
  - STACK(48K,16K,ANY,KEEP),
  - ALL31(ON),
  - BELOWHEAP(8K,2K,KEEP),
  - LIBSTACK(1K,1K,FREE)
- Check the Storage report for recommendations.
- Do not run CGIs in WebSphere Application Server.
- If you do need to run CGIs, control them using Workload Manager.

If the Web server is installed in the default path, /usr/lpp/internet/, your configuration files are copied to the /etc/ directory. Otherwise, they are copied to the /<install_path>/etc/ directory. The following configuration and environment files are used:

- The HTTP server configuration file, httpd.conf
- The HTTP server environment file, httpd.envvars

9.1.2 The httpd.conf file

The httpd.conf definitions should be optimized for the workload running on your application server.

General recommendations

- Since it will be searched/parsed for every request, streamline the httpd.conf dataset.
- Eliminate all extraneous statements in the httpd.conf server configuration file. Map, Pass, Service, and Redirect statements are especially expensive.
- Redirect statements cause the http server to instruct the browser to send a request to another server. This results in two requests for each URL.
- Specify Pass, Exec, and other mapping directives in the most probable match sequence to shorten directory searches. When there are fewer directories in the pathname, there are fewer directory lookup operations.

Following are values that you can check within the httpd.conf file:

IMBEDS

Specify Imbeds OFF if you are not using Server Side Include (SSI). SSI causes the http server to parse every byte of HTML text to search for SSI directives. This may increase CPU time.

MAXACTIVETHREADS

Each time your server receives a request from a client, it uses one or two threads to perform the requested action. If no threads are available, the server holds requests until more threads are available. The MaxActiveThreads setting in your configuration file specifies the maximum number of active threads.

These threads are divided between SSL and normal mode processing; note the following points:

- DNS lookup uses additional threads.
- Too few threads results in queuing, which will increase the response time.
- Too many threads increase memory usage. Experience has shown that a value greater than 200 can cause storage shortages.
- Lower the value when running the http server in scalable mode; raise the value for many static requests.
- The value can be overridden by:
  - BPXPARM: MAXTHREADTASKS
  - RACF, USERID OMVS-Seg: THREADSMAX
  - SMFEXIT: IEFUSI

Recommendation: If the Web server is running in Standalone Server mode, a value of 150 or less is recommended for the MaxActiveThreads directive. If the percentage of static pages served is high (for example, 60% or more), a higher setting may be needed. The setting of MaxActiveThreads must be lower than the UNIX System Services MAXTHREADTASKS value described in “SYSx.PARMLIB(BPXPRMxx)” on page 69.

MAXPERSISTREQUEST

This refers to the number of requests processed from a client before the socket is closed by Web Server. The default is 5. Increasing the value reduces the overhead of creating and closing sockets, but it also reduces the availability of the worker thread to process requests for other clients.

SSLMODE

Enabled or disabled.

Recommendation: Do not request SSL mode on unless you use it, since it requires additional CPU cycles.

ACLS and **USEMETAFILES**

Recommendation**:** Specify Use `ACLS NEVER` and `UseMetaFiles OFF` unless you are exploiting these options.

PERSISTTIMEOUT

The default value is 10sec, correct values depend on workload characteristics.

Listen(Backlog)

TCP/IP-related, this specifies the number of listen backlog client connections for the server to carry. The initial configuration file setting is `ListenBacklog 128`.

Use this directive to specify the number of listen backlog client connections you want the server to carry before sending `connection refused` messages to clients. This number depends upon the number of requests that your server can process in a few seconds, and it should not be set higher than the number the server can process before the clients time out and abort the connection from their end. Requests involving secure transactions take longer than, for instance, client requests that do not require users to logon and give a password.

Note: If the ListenBacklog value is greater than the SOMAXCONN value (found in the TCPIP.PROFILE dataset) supported by TCP/IP, SOMAXCONN is used instead.

9.1.3 Method of starting HTTP

Do not execute production HTTP servers from the UNIX shell unless the WLM Policy is configured to provide adequate resources and priority.

An HTTP server that is run from a UNIX shell inherits the performance specifications of the address space that started it. Normally, TSO or Telnet address spaces are defined with multiple performance periods, and only the first period has adequately high priority.

9.2 Caching

The general rules regarding caching are as follows:

- Cache close to the client, but try to reach as many clients as possible.
- If you preallocate cache sizes (TFS), perform your calculations carefully.
- Use common sense; for example, avoid caching a Servlet in the client's Web cache.

Basically you have two options: use `filecache` in USS, or static cache through the `LocalFileCache` in the HTTP server configuration file.

The USS `filecache` command should be issued from within the Initialization shell script file `/etc/rc`.

The `LocalFileCache` directive in the IBM HTTP Server has the following characteristics:

- It is a static cache for static content.
- It can check for file updates with `LiveLocalCache on`, it then uses more CPU.

Recommendation: Use IBM HTTP server local cache

- LiveLocalCache off
- CacheLocalMaxFiles 200
- CacheLocalMaxBytes 2 M

FastResponseCacheAccelerator (FRCA)

This is a dynamic cache for static pages. You enable it in httpd.conf with the `EnableFRCA on` directive. It is very powerful for static pages but it does not work well with static content in WebSphere Application Server.

Recommendation: Turn `EnableFRCA` off if you are running WebSphere Application Server only.

Note: Due to the fact that all data coming out of WebSphere Application Server will be treated as dynamic, any static pages served by WebSphere Application Server will not be cached in FRCA.

If you have an application with a high amount of static data served out of WebSphere Application Server (WebAppl or .war), we recommend you let the static part be served by IBM HTTP server. To do so, you have to add pass statements accordingly in IHS.

10

The HTTP Transport Handler

When V4.0.1 of WebSphere Application Server for OS/390 was delivered in October 2001, one of the new features was an integrated Transport Handler.

The Transport Handler is an HTTP listener integrated into the WAS 4.0.1 runtime environment. It is designed to listen for HTTP requests and route them to the Web container. It resides on the same system image as the control address space, which brings significant performance improvements over the HTTP Server and plugin.

References

- *e-business Enablement Cookbook for z/OS Volume II: Infrastructure,* SG24-5981
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Operations and Administration*, SA22-7835
- *WebSphere Application Server: Assembling J2EE Applications,* SA22-7836

10.1 Introduction

This chapter describes the HTTP, or HTTPS, Transport Handler.

The HTTP or HTTPS Transport Handler components, which exist in the control region of a J2EE Server, can be enabled to listen for HTTP(S) requests directly. The Control Region will route it to the Web container in the appropriate Server Region for processing.

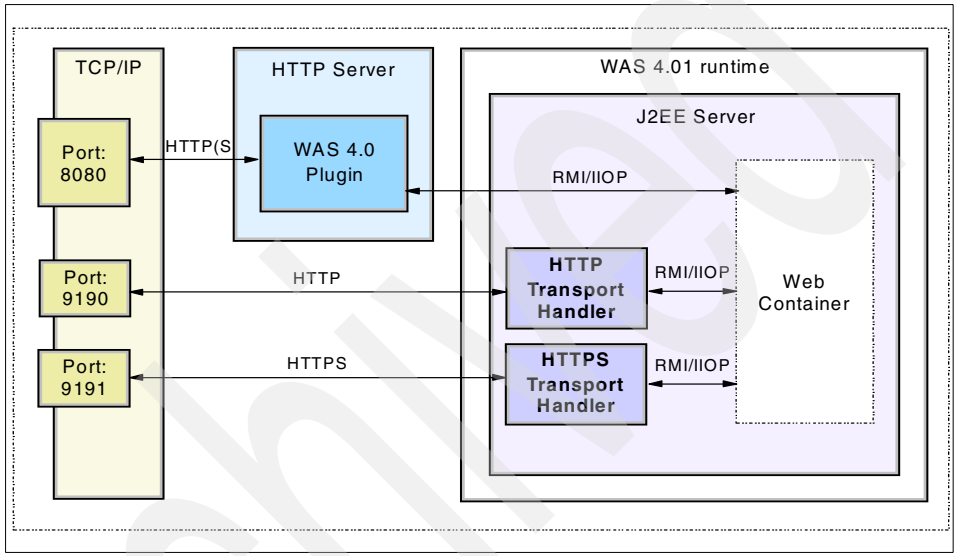


Figure 10-1 The Transport Handler mechanism

Figure 10-1 illustrates the Plugin providing one path for a browser to reach webapps, and the Transport Handlers providing another path. Note that the two processes (Plugin and Transport Handler) are on different ports.

This graphic illustrates a key point: there is really no coordination or connection between the Plugin and the Transport Handlers. They are two different things providing essentially the same basic function.

10.2 Overview

The following is an overview of how the mechanism works, so you can have a better understanding of how to tune it.

1. New parameters are added to the current.env file found in the private directory of your J2EEapplication server. The parameters have the form

BBOC_HTTP_PORT=n and **BBOC_HTTP_SSL_PORT=n** where n is the port number on which you wish the HTTP/HTTPS Transport Handler to listen.

2. The server is started and the Transport Handler function binds to and starts listening on the specified port.
3. In the webcontainer.conf file for your application server's Web container, you have defined virtual hosts, and you have applications that bind to those virtual hosts. This process is exactly the same as was the case with the Plugin.
4. Someone at a browser somewhere issues a URL, and that URL finds its way to your TCP stack and gets picked up by the port on which the transport handler is listening. This is business-as-usual network processing.
5. The Transport Handler takes the request off the port and compares it against virtual host and context-root pairs it knows about from the webcontainer.conf file.

   If there is a match, the request flows to the container to be serviced by the application. If no match occurs (for example, the virtual host is wrong, or the context-root string does not map to any bound application), the request is rejected.

10.2.1 Setup

Figure 10-2 on page 108 illustrates the setup; as shown, do not try to reach other instances with the HTTP Transport Handler from one specific instance.

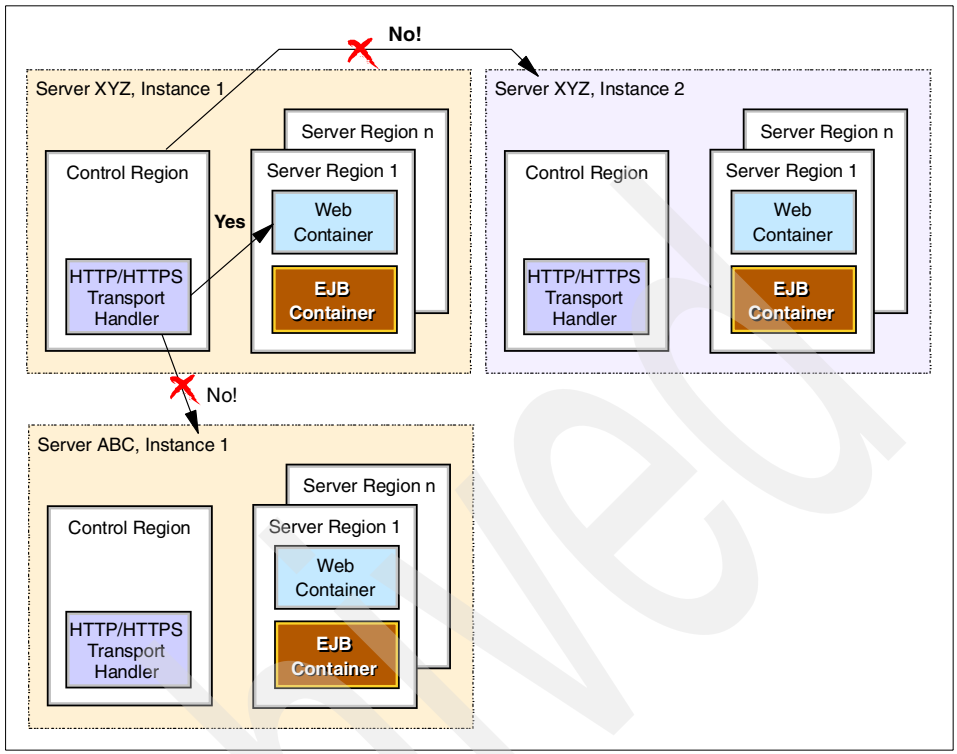


Figure 10-2 HTTP/HTTPS Transport Handler

10.2.2 Effects of coexistence

Figure 10-3 on page 109 shows both functions implemented.

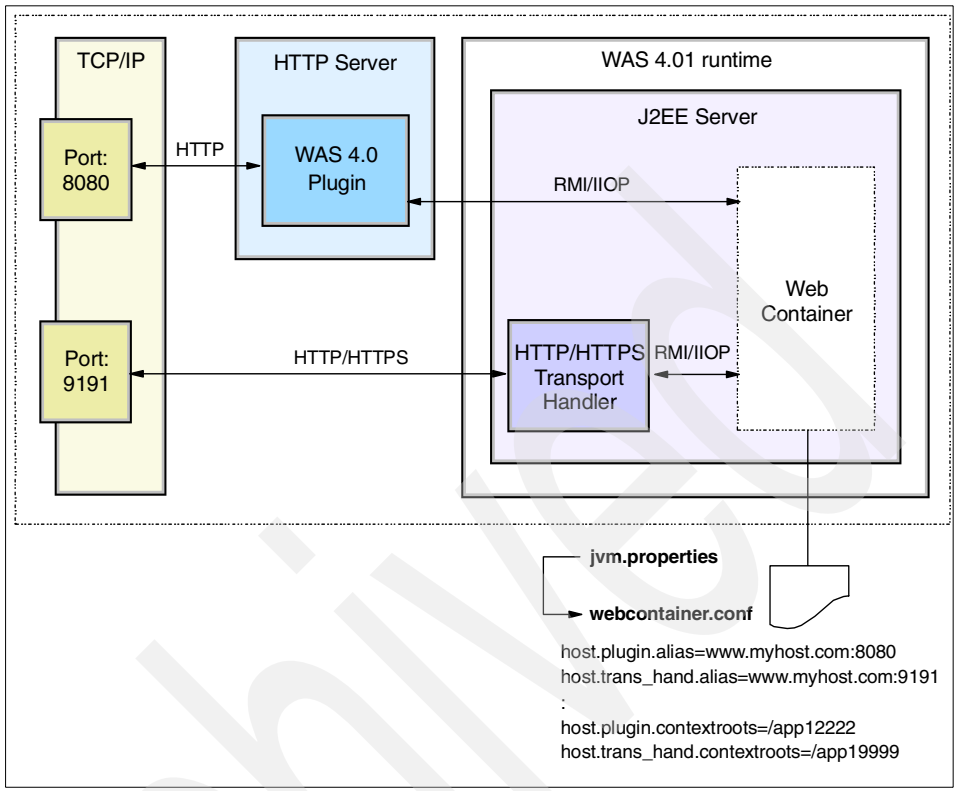


Figure 10-3 Transport Handler and HTTP Server

One effect of this configuration is you cannot bind the same application to both the Plugin and Transport Handler virtual host. The application's context root will be matched against the `contextroots=` values in the `webcontainer.conf` file and bound to a virtual host (this is business-as-usual processing and not new with the Transport Handler).

If you code the same context root string (for example, **`/PolicyIVP)`** on two different `host.<name>.contextroots=` statements, in an attempt to bind the application to two different virtual hosts, the results are unpredictable.

You would be able to accomplish this if you deployed the application into two different *server*s, but not if the application is in the same server.

10.2.3 How many HTTP/HTTPS handlers are needed

You need as many HTTP/HTTPS transport handlers as you have J2EE application server instances configured. The Transport Handler is implemented as a function of the server control region. It is intended to be the HTTP/HTTPS listening device for that server instance. Therefore, if you had 10 J2EE application server instances configured, each with the Transport Handler configured, you would have 10 Transport Handlers in the mix.

Since each Transport Handler will try to bind to the port specified by the parameter `BBOC_HTTP_PORT` or `BBOC_HTTP_SSL_PORT` you could have a conflict if you specify the same value for each instance of the Transport Handler.

If each instance is on a different LPAR, or a different TCP/IP stack, or if you have configured TCP/IP port sharing (though not a recommended configuration), you will not have any problem regarding this aspect. Otherwise, the server will fail to start if the port is already taken.

Care must be exercised to make sure you do not code a port value that will conflict with one already in use.

10.3 What to check

In the following sections we describe areas you should take a look at.

10.3.1 current.env

There are two ways of configuring the HTTP/HTTPS Transport Handler:

- Manually edit the current.env file and include the new parameter. The downside to this is that the update will be lost the next time a conversation is activated in the SMUI tool. This is acceptable for initial ad hoc testing, but is not a recommended permanent solution.
- Use the SM tool and update the *Environment Variable List* for either the server or the Server Instance. With this method the change will persist, even if subsequent conversations are activated.

Clearly, using the SM administrative tool is the proper method.

BBOC_HTTP_INPUT_TIMEOUT
BBOC_HTTP_SSL_INPUT_TIMEOUT

This refers to the time in seconds that the J2EE server will allow for the complete HTTP/HTTPS request to be received before cancelling the connection. The default value is 10 seconds.

Recommendation: `BBOC_HTTP_INPUT_TIMEOUT=0`

BBOC_HTTP_OUTPUT_TIMEOUT
BBOC_HTTP_SSL_OUTPUT_TIMEOUT

This refers to the time, in seconds, that the J2EE server will wait from the time the complete HTTP/HTTPS request is received until output is available to be sent to the client. The default value is 120 seconds.

Recommendation: `BBOC_HTTP_OUTPUT_TIMEOUT=0`

BBOC_HTTP_PERSISTENT_SESSION_TIMEOUT
BBOC_HTTP_SSL_PERSISTENT_SESSION_TIMEOUT

This specifies the time, in seconds, that the J2EE server will wait between requests issued over a persistent connection from an HTTP/HTTPS client. After the server sends a response, it uses the persistent timeout to determine how long it should wait for a subsequent request before cancelling the persistent connection. The default value is 30 seconds.

Recommendation: `BBOC_PERSISTENT_SESSION_TIMEOUT=0`

BBOC_HTTP_SESSION_GC
BBOC_HTTP_SSL_SESSION_GC

This is an integer value indicating the maximum number of HTTP requests that will be processed over a single connection from an HTTP/HTTPS client. When the maximum number of requests have been processed, the client connection will be closed. Set this value to 0 or 1 to turn off persistent connection processing.

Recommendation: `BBOC_HTTP_SESSION_GC=50`

General recommendations

The Transport Handler is designed to listen for and accept HTTP/HTTPS requests from the network and get them to the Web container. The Transport Handler can be viewed as a performance alternative for the HTTP Server and plugin. Typically you would use one or the other, although it is possible to use both concurrently in the same WebSphere configuration.

Environment variable descriptions

Following are environment variable descriptions:

BBOC_HTTP_IDENTITY=*USER_ID*
BBOC_HTTP_SSL_IDENTITY=*USER_ID*

This specifies a valid SAF user ID which will be used as the current security principal for this HTTP/HTTPS request. The user ID will be treated as an authenticated user by the Web container. If this variable is not specified, the request will be executed under the SCO's "local Identity" (`Local_Identity()`).

Recommendation: `BBOC_HTTP_IDENTITY=SECURITY1`

BBOC_HTTP_LISTEN_IP_ADDRESS=*IP_ADDRESS*
BBOC_HTTP_SSL_LISTEN_IP_ADDRESS=*IP_ADDRESS*

This specifies the IP address, in dotted decimal format, that WebSphere for z/OS J2EE servers use to listen for HTTP client connection requests. This IP address is used by the server to bind to TCP/IP. Normally, the server will listen on all IP addresses configured to the local TCP/IP stack. However, if you want to fence the work or allow multiple heterogeneous servers to listen on the same port, you can use `BBOC_HTTP_LISTEN_IP_ADDRESS`.

The specified IP address becomes the only IP address over which this control region receives inbound HTTP/HTTPS requests.

Example: `BBOC_HTTP_LISTEN_IP_ADDRESS=9.117.43.16`

11

The IBM WebSphere Application Server

This chapter describes general WebSphere for z/OS tuning guidelines and performance monitoring procedures for Web or J2EE applications.

References

- *WebSphere Application Server V4.0.1 for zOS and OS390 Operations and Administration,* SA22-7835
- *WebSphere Application Server V4.0.1 Installation and Customization*, GA22-7834

Web sites

http://www7b.boulder.ibm.com/wsdd/zones/bp/
http://www.ibm.com/software/webservers/appserv/zos_os390/library.html

11.1 Introduction

The tuning of your WebSphere Application Server is highly dependent on the type of applications you are serving.

Note: Most of these recommendations are for full functioning production systems; if you are still running a test system, then some of these recommendations might not apply.

11.2 Before you start

Customization of WebSphere involves several different tasks:

- Configuring the environment - LNKLST, LPA, APF, RACF and so forth
- Creating customized WebSphere and LDAP servers
- Defining the initial runtime configuration for WebSphere
- Installing the System Management User Interface (SMUI) on your workstation

These tasks are explained in detail in *WebSphere Application Server V4.01 for z/OS and OS/390 Installation and Customization,* GA22-7834.

11.3 What to check

In the following sections we describe areas you should take a look at.

11.3.1 current.env file

First review the WebSphere for z/OS configuration. A simple way to do this is to look in your application control and server regions in SDSF. When each server starts, the runtime prints out the current configuration data in the joblog.

Note: The environment variable `SHOW_SERVER_SETTINGS=YES` ensures all configuration values are printed out.

TRACEALL To start, ensure that you are not collecting more diagnostic data than you need. Check the WebSphere tracing options to ensure that `TRACEALL=0 or 1`, and that `TRACEBASIC` and `TRACEDETAIL` are not set.

Recommendations:

- If you use any level of tracing, including `TRACEALL=1`, ensure that you set `TRACEBUFFLOC to BUFFER`. Note that `TRACEALL=1` will write exceptions to the CTRACE log, as well as to the ERROR log.
- To reduce memory requirements, you can set `TRACEBUFFERNUMBER=4` and `TRACEBUFFERSIZE=128`, which will get 512 KB of storage for the trace buffers (the minimum allowed).
- Disable JRAS tracing. To do this, look for the following lines in the JVM properties file:

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

  Either set both lines to `disable`, or delete the two lines altogether.

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

WebSphere for z/OS provides usability to the application trace and debug configuration in each server definition. Unless you are using Object Level Trace (OLT) or distributed debugger, you should set the DEBUG allowed field to NO. This will prevent the WebSphere for z/OS runtime from calling the OLT/debug interfaces for each method, which will improve performance. Make sure the Application Trace and Debug checkbox in the server definition is not checked.

11.3.2 Locations of executable programs

The next thing to review in the configuration is where your program code is located. IBM recommends that you install as much of the WebSphere for z/OS code itself in LPA as is reasonable, and the remainder in the linklist. This ensures that you have eliminated any unnecessary steplibs which can adversely affect performance.

It is recommended that for simplicity, any C/C++ application code be located in the HFS.

You should ensure that only server regions have visibility to your application code. The control region usually runs with no steplibs, since all the code required is located in system locations. Verify that the STEPLIB DD in the control region and server region procs do not point to anything unnecessary.

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

Review the PATH statement, found in the was.conf file, to ensure that only required programs are in the PATH and that the order of the PATH places frequently referenced programs in the front. If you are using Java, refer to Chapter 12, “The JVM environment” on page 121.

11.3.3 Storage

For detailed information, refer to the following sections:

- “Virtual storage” on page 50
- “Real storage” on page 52
- “LEPARMS HEAPPON” on page 53

11.3.4 J2EE server properties

Implement these tips whether you are using the WebSphere Application Server as a plugin to the IBM HTTP Server, or are using the HTTP Transport Handler.

Use the System Management User Interface (SMEUI) to set the properties in the current.env file for a J2EE Server, as shown in Figure 11-1 on page 117.

- Server region stack size (in bytes) = 0

  A specification of zero causes the default value of one million bytes to be used. Set the property to zero for both your production and development environments.

- Production J2EE server = Check.
- Debugger Allowed = Uncheck.
- Insure that you turn off JDBC tracing in the file specified on the DB2SQLJPROPERTIES environment variable, by commenting out the following DB2 property: `#DB2SQLJ_TRACE_FILENAME= file name`
- Allow Server Region Recycling: Uncheck.

  WebSphere for z/OS has a feature called *server region recycling* (previously named *server region garbage collection*). This gives the installation a threshold for the number of transactions to execute in a specific Server Region before the Server Region is thrown away. This can be very helpful in improving the performance of applications that have storage leaks.

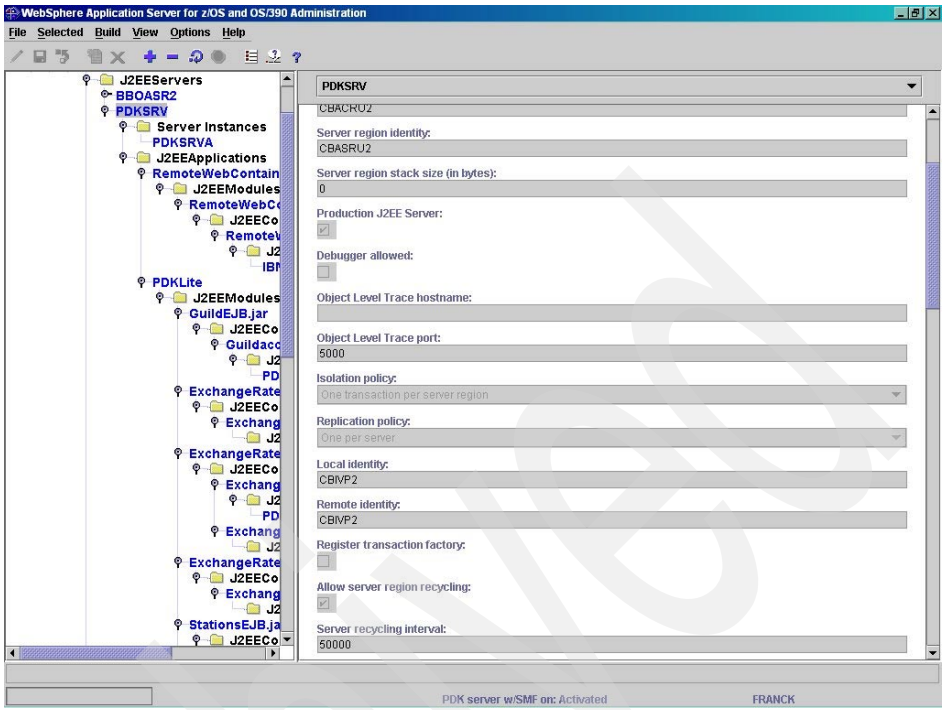


Figure 11-1 J2EE server properties

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

When your application code is leaking storage, it can cause each storage obtain to get slower and slower. After the specified number of transactions, you get a new SR and storage obtains are fast again. This produces a somewhat saw-toothed performance curve where performance starts good and degrades until the SR is recycled and performance once again improves.

Clearly the long-term solution for application leaks is to fix the application, but Server Region recycling can help when you can't fix the application immediately.

11.3.5 WebSphere V3.5 run-time environment

Check these performance tips if you use an IBM HTTP Server for z/OS or OS/390 platforms to handle requests for Web applications, using WebSphere Application Server Version 3.5 or WebSphere Application Server V4.0.x with the V3.5 run-time environment shipped with Versions 4.0.x.

Set the properties in the was.conf file:

- appserver.loglevel=WARNING or ERROR
- appserver.nodetach=false

  You can set this property to true for better performance. However, you cannot do this if either of the following is true:

  a. The property `jdbcconnpool.<pool-name>.connectionidentity=thread` is specified in the was.conf file.
  b. Your application creates Java threads other than the threads created for you by the WebSphere Application Server.

  When you specify `appserver.nodetach=true`, also set `JAVA_PROPAGATE=NO` in the httpd.envvars file.
- session.tableoverflowenable=true
- session.dbenable=false
- jdbcconnpool. <pool-name> .connectionidentity=connspec

  Applications ported from the distributed versions of WebSphere Application Server typically code the `connspec` value on this property. The `consspec` value on the z/OS and OS/390 platforms works the same way as the distributed version.
- deployedwebapp.<Web-application-name>.autoreloadinterval=0

  Set this property to 0 in production to turn off servlet reloading. Servlet reloading impacts performance and is only recommended in development or test environments.
- webapp.< Web-application-name>.jsplevel=1.0

  Use JSP specification level 1.0 and later when possible. Use the highest JSP specification allowed for your level of the WebSphere Application Server to receive the best performance improvement.
- #appserver.tracelevel=*=debug=enabled

  Keep this property commented out, unless requested by support personnel.

Setting properties in the jvm.properties file

If you implement WebSphere Application Server V4.0x and run applications in a J2EE server set environment variables in the current.env file using the SMUI administration tool.

If you have applications running in the WebSphere Application Server plug-in, set properties in the jvm.properties file. In Version 3.5 of the WebSphere Application Server, the default name of the file is `debug_global.properties` but you may have changed the name to `jvm.properties` or some other name.

Set the properties as follows:

- appserver.product.java.jvmconfig.jit=jitc

  Alternatively, you can comment out the property to turn on the JIT compiler, `#appserver.product.java.jvmconfig.jit=jitc`
- appserver.product.java.jvmconfig.mx=256m
- appserver.product.java.jvmconfig.ms=256m
- appserver.product.java.jvmdebug.verbosegc=true

  You can set this property to true and obtain the trace in production. The effect on performance is relatively small.

12

The JVM environment

This chapter contains information about how to tune the WebSphere Application Server JVM environments to provide better performance when running Java applications.

References

- *WebSphere Application Server V4.0 for z/OS and OS/390: Installation and Customization,* GA22-7834
- *IBM Host Integration: A Practical Approach to Performance Planning*, SG24-5960

Web site

http://www.ibm.com/servers/eserver/zseries/software/java/perform.html

12.1 What to check

The following sections discuss areas which should be checked.

12.1.1 JDK release level

Although Java has matured significantly over the last few years, it is a relatively new technology. Almost every release of the JDK brings new functions and performance enhancements. PTFs too can bring improvements. By keeping current with the latest maintenance level, you'll be able to take advantage of this progress.

Recommendation: Make sure that you are running at the most current level of the JVM, with any applicable PTFs installed.

12.1.2 zip and jar files

Keep in mind that zip and jar files can be used to combine many class files into one file for easier loading or file transfer. When done properly, this can improve application load time.

Recommendation**:** Avoid adding unneeded class files, which increases file size and memory usage. One common mistake is to produce a zip file of every utility and builder class, when the application may need only a small proportion of the class files to execute. This results in increased load time.

12.1.3 CLASSPATH

When the JVM is looking for a class, it searches the directories in the order they are given in CLASSPATH.

Recommendation**:** You can improve the start-up times of your Java applications by reordering the paths in the CLASSPATH environment variable and placing the most used libraries earlier in the CLASSPATH. It is no longer necessary to add system classes (classes.zip in JDK 1.1.8) to CLASSPATH.

12.1.4 current.env file

In this file there are a number of parameters that can be adjusted.

Each of these parameters should be changed using the System Management User Interface (SMUI), and not by editing the current.env file.

Note: Any changes made to the current.env file as it exists in the HFS will be overwritten when you either activate a conversation via the SMUI or perform a cold start.

JVM_COMPILER The overhead for interpreting bytecodes is expensive. Compiling the bytecodes into native machine code with the just-in-time (JIT) compiler and then executing in native machine code is much faster. Laboratory measurements have shown that running with the JIT compiler turned on can significantly improve performance for some applications.

You do not have to specify `JAVA_COMPILER` at all, in which case the default is to set the JIT compiler on. If you use the environment variable, a null value `JAVA_COMPILER=` turns the JIT compiler on. Any other value turns the JIT compiler off (see APAR PQ51175).

Recommendation**:** Do not specify `JAVA_COMPILER` or specify a null value **"**`JAVA_COMPILER=`**"** to turn the JIT compiler on.

JVM_HEAPSIZE If the HEAPSIZE is too large, there will be longer delays when garbage collection is eventually done. It also increases the amount of system resources needed for each of the application servers that can be started by WLM. Since WLM will start multiple application servers as the workload increases, there is no need to run with an extra large HEAPSIZE. It should only be set high enough to support the running of the largest application(s).

If the HEAPSIZE is too small, applications may fail because of an "out of memory" condition, or more frequent garbage collection may cause a great deal of CPU overhead. Set `JVM_ENABLE_VERBOSE_GC = 1` for a Garbage Collection report, which you can use to determine how big the `HEAPSIZE` should be.

Recommendation: Set to 256 M and then tune downward. The `JVM_HEAPSIZE` should be set to 256 M and should be equal to the `JVM_MINHEAPSIZE` so that garbage collection is only done when the heap is full. Make sure that the region size for each application server is large enough to hold the `HEAPSIZE`. A region size of 0 MB will allocate the largest possible region size allowed by the installation.

JVM_MINHEAPSIZE The JVM_MINHEAPSIZE should also be set to 256 M and equal the value set for JVM_HEAPSIZE so that garbage collection is only done when the heap is full.

Recommendation: Set to 256 M and then tune to match `JVM_HEAPSIZE`.

JVM_DEBUG Do not turn on (1) unless you are debugging a problem.

Recommendation**: L**eave blank.

JVM_ENABLE_VERBOSE_GC

We recommend to enable this option as it has small performance cost and provides a lot of useful information.

Recommendation**:** Turn on (1)

A

SMFPRMxx

In this appendix we discuss the SMFPRMxx member, which can be found in SYSx.PARMLIB and controls the recording of SMF records.

For more information about the SMFPRMxx member, see *MVS Initialization and Tuning Reference,* SC28-1752 and refer to Chapter 66, “SMFPRMxx, System Management Facilities (SMF) Parameters”.

SMFPRMxx overview

The SMFPRMxx member allows you to control how system management facilities (SMF) work at your installation. You can use SMFPRMxx parameters to:

- Identify the system on which SMF is active
- Specify global values for interval recording and synchronization that SMF, RMF, and other requestors can use to schedule the execution of their interval functions
- Specify the datasets to be used for SMF recording
- Specify the system identifier to be used in all SMF records
- Select the SMF record types and subtypes SMF is to generate
- Allow the operator to change the SMF parameters established at IPL
- Specify the job wait time limit
- Specify whether SMF is to invoke installation-supplied SMF exit routines
- Specify whether the SMF dump program is to attempt to recover from abends
- Specify the system response when SMF has used all of the buffered storage in its address space
- Specify the system response when the last SMF dataset is filled and no other datasets are available for use

You can dynamically modify SMF execution by using the **SET SMF** command to specify a different SMFPRMxx parmlib member, or by using the **SET SMF** command to replace one or more previously defined SMF parameters.

Example SMFPRMxx member

```
Check ==>  ***************************** Top of Data ******************************
           ACTIVE                       /*ACTIVE SMF RECORDING*
           DSNAME(SYS1.SC58.MAN1,SYS1.SC58.MAN2,SYS1.SC58.MAN3)
           NOPROMPT                     /*DON'T PROMPT THE OPERATOR*/
           REC(PERM)                    /*TYPE 17 PERM RECORDS ONLY*/
           MAXDORM(3000)               /* WRITE AN IDLE BUFFER AFTER 30 MIN*/
           STATUS(010000)              /* WRITE SMF STATS AFTER 1 HOUR*/
           JWT(0010)                   /* 522 AFTER 10 MINUTES*/
           SID(SC58)                   /* SYSTEM ID IS 3090         */
           LISTDSN                     /* LIST DATA SET STATUS AT IPL*/
           LASTDS(MSG)                 /*DEFAULT TO MESSAGE          */
           NOBUFFS(MSG)                /*DEFAULT TO MESSAGE          */
Check ==>            SYS(TYPE(30,70:79,103,120),
                     EXITS(IEFU83,IEFU84,IEFU85,IEFACTRT,
                     IEFUJV,IEFUSI,IEFUJP,IEFUSO,IEFUTL,IEFUAV),
                     INTERVAL(SMF,SYNC),DETAIL)
           SUBSYS(STC,EXITS(IEFU29,IEFU83,IEFU84,IEFU85,IEFUJP,IEFUSO))
           ***************************** Bottom of Data ***************************
```

Selected syntax

In this section, we describe some of the more important statements and parameters of SMFPRMxx:

ACTIVE | NOACTIVE This specifies whether SMF recording is to be active.

Default: ACTIVE

INTVAL(mm) This specifies the length of time (in minutes) from the end of an SMF global recording interval to the end of the next interval. For example, if you specify INTVAL(15), the SMF global recording interval ends every 15 minutes. INTVAL is a global interval value that other requestors, such as RMF, can use to schedule interval functions to execute in conjunction with the SMF interval function.

Choose a global interval value (INTVAL) that divides evenly into 60 (1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, and 60). Otherwise, the system can synchronize only the first interval.

Value Range: 1-60.
Default: 30
Recommendation: 15

Note: Only SMF records can be controlled by the INTVAL and SYNCVAL parameters. The INTVAL and SYNCVAL parameters can influence other record types, such as when activated by the RMF Monitor I .SYNC(SMF).option to write record types 70 through 79.

SYNCVAL(mm) This specifies the global synchronization value (in minutes) for the SMF global recording interval, synchronizing the recording interval with the end of the hour on the TOD clock. For example, if you specify SYNCVAL(15), the global recording interval is synchronized to 15 minutes past the hour. If you also specify INTVAL(30), SMF global recording intervals end at 15 minutes and 45 minutes past the hour.

When you specify the SYNCVAL parameter for interval synchronization, specify the global interval value with the INTVAL parameter (unless you accept the default for INTVAL).

Value Range: 00-59.
Default: 00

Note: Only SMF records can be controlled by the INTVAL and SYNCVAL parameters. The INTVAL and SYNCVAL

parameters can influence other record types, such as when activated by the RMF Monitor I .SYNC(SMF).option to write record types 70 through 79.

SYS (options) This specifies the SMF recording options and exits for the entire system. The options are as follows:

TYPE Record types and subtypes to be collected.
INTERVAL Time intervals between recording.
EXITS Exits that are to receive control at various points in SMF processing.
DETAIL The level of SMF data collection for TSO users and started tasks. If the same option is specified more than once, the system uses the first valid operator reply. The following information describes the options in greater detail.
TYPE {aa,bb(cc)}
NOTYPE ({aa,bb:zz })
{aa,dd(cc:yy),...}
{aa,bb(cc,...)}

TYPE specifies the SMF record types and subtypes that SMF is to collect. aa, bb, dd, and zz are the decimal notations for each SMF record type. cc and yy are the decimal notations for the SMF record subtypes. A colon (:) indicates the range of SMF record types (bb through zz) to be recorded or the range of subtypes (cc through yy for SMF record type dd) to be recorded.

Subtypes are valid on SMF record types 0-127.

NOTYPE specifies that SMF is to collect all SMF record types and subtypes *except* those specified. aa, bb, and zz are the decimal notations for each SMF record type. cc and yy are the decimal notations for each subtype. A colon (:) indicates the range of SMF record types (bb through zz) or the range of subtypes (cc through yy for SMF record dd) that are *not* to be recorded.

Value Range: 0-255 for SMF record types, 0-32767 for subtypes.
Default: TYPE (0:255), all types and subtypes.

B

BPXPRMxx

In this appendix we discuss the BPXPRMxx member, which is found in SYSx.PARMLIB and is used to define the UNIX Systems Services (USS) environment on your system.

For more information about this member, refer to *MVS Initialization and Tuning Reference,* SC28-1752.

BPXPRMxx overview

BPXPRMxx contains the parameters that control the z/OS UNIX System Services (z/OS UNIX) environment and the file systems. IBM recommends that you have two BPXPRMxx parmlib members, one defining the values to be used for system setup, and the other defining the file systems. This makes it easier to migrate from one release to another, especially when using the ServerPac method of installation.

How to specify which member

After installation is complete, the operator needs to specify OMVS=xx in the IEADYDxx parmlib member. To specify which BPXPRMxx parmlib member to start with, the operator can include OMVS=xx in the reply to the IPL message, or OMVS=xx in the IEASYSxx parmlib member. The two alphanumeric characters represented by xx are appended to BPXPRM to form the name of the BPXPRMxx parmlib member.

To modify BPXPRMxx parameter settings without re-IPLing, you can use the **SET OMVS** operator command, or you can dynamically change the BPXPRMxx parmlib members that are in effect by using the **SET OMVS** operator command.

Example BPXPRMxx member

```
******************************** Start of File ****************************
MAXPROCSYS(400)
MAXPROCUSER(16)
MAXUIDS(200)
MAXFILEPROC(10000)
MAXTHREADTASKS(5000)
MAXTHREADS(10000)
MAXPTYS(100)
MAXFILESIZE(1000)
MAXCORESIZE(4194304)
MAXASSIZE(41943040)
MAXCPUTIME(1000)
MAXMMAPAREA(4096)
MAXSHAREPAGES(32768)
PRIORITYPG(7,7,7,7,7,6,5,999,3,2,1)
PRIORITYGOAL(CICS4,CICS4,CICS4,CICS3,CICS2,CICS1,TSO2,TSO1,BAT3,BAT2)
IPCMSGNIDS(500)
IPCMSGQBYTES(262144)
IPCMSGQMNUM(100000)
IPCSEMNIDS(500)
```

Check ===>

```
IPCSEMNOPS(25)
IPCSEMNSEMS(25)
IPCSHMMPAGES(256)
IPCSHMNIDS(500)
IPCSHMNSEGS(10)
IPCSHMSPAGES(262144)
FORKCOPY(COW)
SUPERUSER(BPXROOT)
TTYGROUP(TTY)
CTRACE(CTCBPX23)
STEBLIBLIST('/etc/steplib')
USERIDALIASTABLE('/etc/tablename')
SYSPLEX(YES)
VERSION('REL9')
FILESYSTYPE TYPE(HFS)
ENTRYPOINT(GFUAINIT)
PARM('SYNCDEFAULT(0)FIXED(2)VIRTUAL(128)')
ROOT FILESYSTEM('OMVS.ROOT')
TYPE(HFS)
MODE(RDWR)
SYSNAME(SY1)
AUTOMOVE
MOUNT FILESYSTEM('OMVS.USER.JONES')
TYPE(HFS)
MOUNTPOINT('/u/jones')
MODE(RDWR)
SYSNAME(SY1)
NOAUTOMOVE
FILESYSTYPE TYPE(INET)
ENTRYPOINT(EZBPFINI)
NETWORK DOMAINNAME(AF_INET)
DOMAINNUMBER(2)
MAXSOCKETS(2000)
TYPE(INET)
STARTUP_PROC(OMVS)
STARTUP_EXEC('OMVS.ROOT(REXX01)',A)
RUNOPTS('RTLS(ON)LIBRARY(SYSCEE)VERSION(OS24)')
SYSCALL_COUNTS(YES)
MAXQUEUEDSIGS(1000)
****************************** End of File *********************************
```

C

TCPIP profile

In this appendix, we discuss the TCPIP profile dataset.

Locate the TCPIP profile dataset

To locate the TCPIP profile dataset, check the procedure used to start TCP/IP on your system. Within the procedure a DD statement with ddname PROFILE points to the dataset.

On our system, the procedure is named TCPIP and is in SYS1.PROCLIB. This resolves to dataset TCP.SC58.TCPPARMS(TCPPROF).

```
******************************** Top of Data ***********************
//TCPIP PROC P1='CTRACE(CTIEZB00)',TCPPROF=TCPPROF,TCPDATA=TCPDATA
//*
//TCPIP EXEC PGM=EZBTCPIP,REGION=0M,TIME=1440,
//   PARM=&P1
//STEPLIB   DD DSN=TCPIP.SEZATCP,DISP=SHR
//SYSPRINT  DD SYSOUT=*,DCB=(RECFM=VB,LRECL=137,BLKSIZE=0)
//SYSERR    DD SYSOUT=*,DCB=(RECFM=VB,LRECL=137,BLKSIZE=0)
//SYSERROR  DD SYSOUT=*
//ALGPRINT  DD SYSOUT=*,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//CFGPRINT  DD SYSOUT=*,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//SYSOUT    DD SYSOUT=*,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//CEEDUMP   DD SYSOUT=*,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//SYSERROR  DD SYSOUT=*
//PROFILE   DD DSN=TCP.&SYSNAME..TCPPARMS(&TCPPROF),
//          DISP=SHR,FREE=CLOSE
//SYSTCPD   DD DSN=TCP.&SYSNAME..TCPPARMS(&TCPDATA),DISP=SHR
//SYSABEND  DD SYSOUT=*
******************************** Bottom of Data ********************
```

Example TCPIP profile

```
DATASETPREFIX TCP
TCPCONFIG       RESTRICTLOWPORTS
TCPSENDBFRSIZE  65535
UDPCONFIG       RESTRICTLOWPORTS
; ---------------------------------------------------------------------
; AUTOLOG the following servers
; ---------------------------------------------------------------------
; Reserve ports for the following servers.
PORT
    20 TCP OMVS NOAUTOLOG      ; OE FTP Server
    21 TCP FTPDMVS1 BIND 9.12.14.206
    21 TCP FTPD1 BIND 9.12.2.22
    21 TCP OMVS                ; OE FTPD control port
    23 TCP INTCLIEN            ; Telnet 3270 Server
    23 TCP OMVS BIND 9.12.2.22 ; OE telnet server
    80 TCP OMVS                ; OE Web Server
```

```
    111 UDP OMVS               ; OE Portmapper Server
    111 TCP OMVS               ; OE Portmapper Server
    443 TCP OMVS               ; OE Web Server SSL Port
    512 TCP OMVS               ; OE Remote Execution Server
    513 TCP OMVS               ; OE Rlogin Server
    514 TCP OMVS               ; OE Remote Shell Server
    514 UDP OMVS               ; OE SyslogD Server
    515 TCP OMVS               ;
    900 TCP SYSMGT01           ; WebSphere System Mgmt server
   1389 TCP BBOLDAP            ; WebSphere System LDAP server
   5555 TCP DAEMON01           ; WebSphere Daemon server

;  1414 TCP CSQ1CHIN           ; MQ Channel Initiator

   7101 TCP OMVS SHAREPORT     ; Web Servers

 PORTRANGE 10000 2000 TCP OMVS ; TCP 10000 - 11999
 PORTRANGE 10000 2000 UDP OMVS ; UDP 10000 - 11999
 SOMAXCONN 1000
; -------------------------------------------------
; Hardware definitions:
; -------------------------------------------------
 DEVICE OSA2220   LCS          2220
 LINK   OSAL2220  IBMTR    0   OSA2220
 DEVICE OSA2200   LCS          2200
 LINK   OSAL2200  IBMTR    0   OSA2200

; ------------------------------------------------------------------------
; HOME internet (IP) addresses of each link in the host.
; ------------------------------------------------------------------------
HOME
   9.12.2.22     OSAL2220
   9.12.14.206   OSAL2200
; PRIMARYINTERFACE OFFLAN1
GATEWAY
; Direct Routes - Routes that are directly connected to my interfaces.
; Network First Hop Link Name Packet Size  Subnet Mask    Subnet Value
  9          =     OSAL2220     4096      0.255.255.0    0.12.2.0
  9          =     OSAL2200     4096      0.255.255.0    0.12.14.0
;
; Indirect Routes - Routes that are reachable through routers
; Network  First Hop  Link Name Packet Size  Subnet Mask  Subnet Value
; 193.12.2 130.50.10.1   TR1       2000       0
; 10.5.6.4 193.5.2.10    ETH1      1500       HOST
; Network  First Hop  Link Name Packet Size  Subnet Mask  Subnet Value
  DEFAULTNET 9.12.2.75 OSAL2220  4096         0
;
; ------------------------------------------------------------------------
; RouteD Routing Information
```

```
; ;  Link     Maxmtu   Metric   Subnet Mask     Dest Addr
;BSDROUTINGPARMS false
;    OSAT2200  4096       0     255.255.255.0   0
;ENDBSDROUTINGPARMS
;
; ---------------------------------------------------------------------
; TRANSLATE
; ---------------------------------------------------------------------
; Turn off all tracing.  If tracing is to be used, change the following
; line.  To trace the configuration component, for example, change
; the line to ITRACE ON CONFIG 1

ITRACE OFF
;
; ---------------------------------------------------------------------
ASSORTEDPARMS
; NOFWD
  RESTRICTLOWPORTS
ENDASSORTEDPARMS
; NOFWD            issues the informational message EZZ0334I
; RESTRICTLOWPORTS issues the informational message EZZ0338I
; ---------------------------------------------------------------------
IPCONFIG NODATAGRAMFWD
; ---------------------------------------------------------------------
; Configure Telnet
; ---------------------------------------------------------------------
TelnetParms
  Port 23                       ; Port number 23 (std.)
  CodePage ISO8859-1 IBM-1047   ; Linemode ASCII, EBCDIC code pages
  Inactive 0                    ; Let connections stay around
  PrtInactive 0                 ; Let connections stay around
  TimeMark 600
  ScanInterval 120
; SMFinit std
; SMFterm std
  WLMClusterName
    TN3270E
  EndWLMClusterName
EndTelnetParms
;
; TelnetParms
;   Secureport 992 Keyring HFS /tmp/telnet.kdb
; EndTelnetParms
;
; ---------------------------------------------------------------------
; BEGINVTAM: Defines the VTAM parameters required for the Telnet server.

BeginVTAM
  Port 23 ; 992
```

```
;     ; Define logon mode tables to be the defaults shipped with the
;     ; latest level of VTAM
    TELNETDEVICE 3278-3-E NSX32703 ; 32 line screen
    TELNETDEVICE 3279-3-E NSX32703 ; 32 line screen
    TELNETDEVICE 3278-4-E NSX32704 ; 48 line screen
    TELNETDEVICE 3279-4-E NSX32704 ; 48 line screen
    TELNETDEVICE 3278-5-E NSX32705 ; 132 column screen
    TELNETDEVICE 3279-5-E NSX32705 ; 132 column screen
;     ; Define the LUs to be used for general users.
    DEFAULTLUS
        SC58TC01 SC58TC02 SC58TC03 SC58TC04 SC58TC05
        SC58TC06 SC58TC07 SC58TC08 SC58TC09 SC58TC10
        SC58TC11 SC58TC12 SC58TC13 SC58TC14 SC58TC15
        SC58TC16 SC58TC17 SC58TC18 SC58TC19 SC58TC20
        SC58TC21 SC58TC22 SC58TC23 SC58TC24 SC58TC25
        SC58TC26 SC58TC27 SC58TC28 SC58TC29 SC58TC30
        SC58TC31 SC58TC32 SC58TC33 SC58TC34 SC58TC35
        SC58TC36 SC58TC37 SC58TC38 SC58TC39 SC58TC40
        SC58TC41 SC58TC42 SC58TC43 SC58TC44 SC58TC45
        SC58TC46 SC58TC47 SC58TC48 SC58TC49 SC58TC50
        SC58TC51 SC58TC52 SC58TC53 SC58TC54 SC58TC55
        SC58TC56 SC58TC57 SC58TC58 SC58TC59 SC58TC60
        SC58TC61 SC58TC62 SC58TC63 SC58TC64 SC58TC65
        SC58TC66 SC58TC67 SC58TC68 SC58TC69 SC58TC70
        SC58TC71 SC58TC72 SC58TC73 SC58TC74 SC58TC75
        SC58TC76 SC58TC77 SC58TC78 SC58TC79 SC58TC80
    ENDDEFAULTLUS
    LUSESSIONPEND
    DEFAULTAPPL TSO
    LINEMODEAPPL TSO
    ALLOWAPPL TSO* DISCONNECTABLE
; ; RESTRICTAPPL IMS
; ;   USER USER1
; ;     LU TCPIMS01
; ;   USER USER2
; ;   USER USER3
; ;     LU TCPIMS31 LU TCPIMS32 LU TCPIMS33
    ALLOWAPPL *
ENDVTAM

; Start all the defined devices.
;
START OSA2220
START OSA2200
```

D

slapd.conf

In this appendix, we discuss the slapd.conf file.

Locate the slapd.conf file

Since naming conventions may not be the same in your system, you can check the current.env file for any of the application servers; they contain a pointer to the location of the slapd.conf file that is being used.

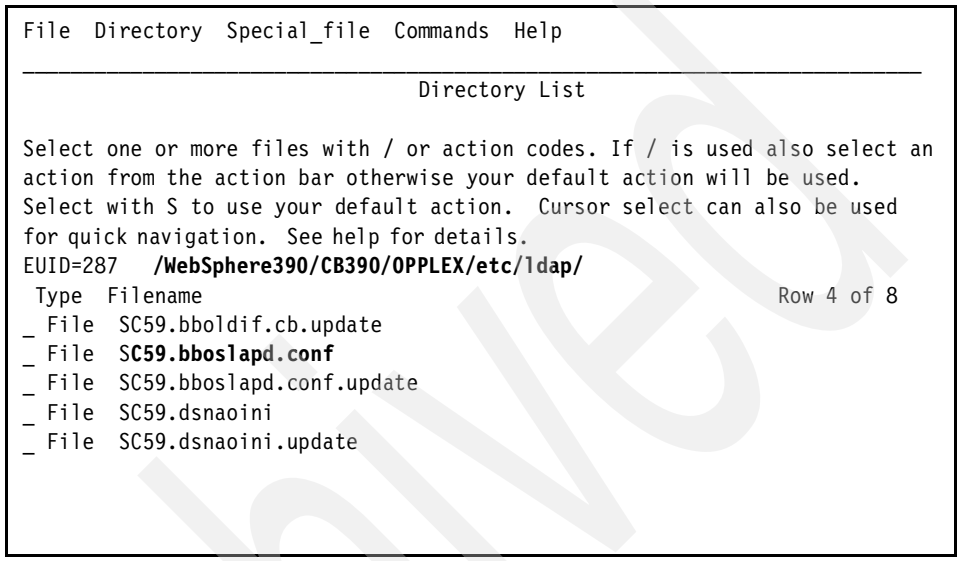


Figure D-1 The slapd.conf file

As shown, on our test system the slapd.conf file is located in directory `WebSphere390/CB390/OPPLEX/etc/ldap`.

Our sysplex name is OPPLEX; yours will differ.

Example bboslapd.conf file

```
******************************** Top of Data ********************************
#SC59.bboslapd.conf file:
#----------------------------
# PROPRIETARY-STATEMENT:
# Licensed Material - Property of IBM
# 5655-F31 (C) Copyright IBM Corp. 2001
# All Rights Reserved.
# U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
# Disclosure restricted by GSA-ADP schedule contract with IBM Corp
#-------------------------------------------------------------
#  Attribute and object class definitions
```

```
#---------------------------------------------------------------
include /usr/lpp/ldap/etc/slapd.at.system
include /usr/lpp/ldap/etc/slapd.cb.at.conf
include /usr/lpp/ldap/etc/slapd.at.conf
include /usr/lpp/ldap/etc/slapd.oc.system
include /usr/lpp/ldap/etc/slapd.cb.oc.conf
include /usr/lpp/ldap/etc/slapd.oc.conf
#---------------------------------------------------------------
#  ... / ...
#---------------------------------------------------------------
port           1389
commthreads 10
maxconnections 200
timelimit      3600
sizelimit      500
#sysplexGroupName BBOOPPLEX
#sysplexServerName OPPLEX
adminDN        "cn=CBAdmin"
adminPW        secret
#---------------------------------------------------------------
#  DB2 database access definitions
#
#  Note: LDAP code doesn't handle the dsnaoini living in the
#        HFS until R10.  So we will include a //DSNAOINI DD PATH=
#        statement in the LDAP server, BBONMS and BBOIRS procs.
#---------------------------------------------------------------
database rdbm  GLDBRDBM
#multiserver    y
#dsnaoini       not used check JCL
servername     DB7A
databasename   BBOLDAP
dbuserid       BBOLDAP
tbspaceentry   BBOENT
tbspace32k     BBO32K
tbspace4k      BBO4K
#tbspacemutex  BBOMUTX
suffix         "cn=localhost"
suffix         "o=WASNaming,c=US"
suffix         "o=BOSS,c=US"
****************************** Bottom of Data ******************************
```

Check ===>

E

httpd.conf

In this appendix, we discuss the httpd.conf file.

Locate the httpd.conf file

By default, the httpd.conf file is located in directory \etc\, as shown:

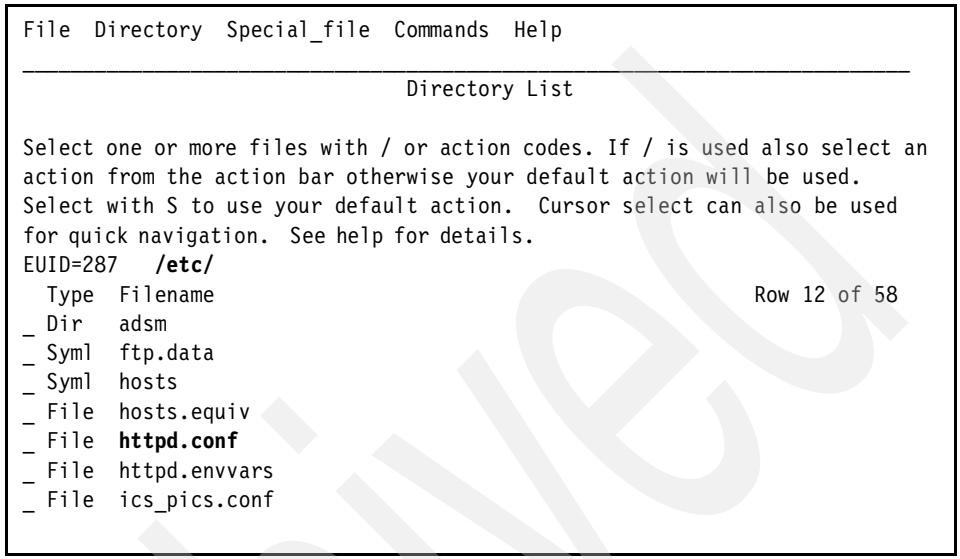


Figure E-1 The httpd.conf file

Example httpd.conf file

```
# ====================================================================
# @(#)53 1.9  src/web/etc/httpd.conf, web, web4a1J 6/21/95 10:19:08
# COMPONENT_NAME: web httpd.conf
# FUNCTIONS:
# ORIGINS: 10  26  27
# (C) COPYRIGHT International Business Machines Corporation 1997
# All Rights Reserved
# This software is subject to the IBM Software Agreement
# Restricted Rights for U.S. government users, and applicable export
# regulations. IBM is a registered trademark and IBM HTTP Server
# is a trademark of IBM Corporation.
# ====================================================================
# ================================================================ #
#       Basic directives
## ================================================================
InstallPath     /usr/lpp/internet
ServerRoot     server_root
# Hostname directive:
BindSpecific  off
```

```
Check ===>  DNS-Lookup     off
            Port     80
Check ===>  imbeds on SSIOnly
            # InheritEnv SERVER_NAME
            # DisInheritEnv REMOTE_HOST
            UserId     %%CLIENT%%
            PidFile  /usr/lpp/internet/server_root/httpd-pid
            #Bounce on
            ============================================================#
            #        Logging and Reporting directives
            # ===========================================================
            # ProxyAccessLog   /usr/lpp/internet/server_root/logs/httpd-proxy
            # CacheAccessLog   /usr/lpp/internet/server_root/logs/httpd-cache
            AccessLog     /usr/lpp/internet/server_root/logs/httpd-log
            AgentLog      /usr/lpp/internet/server_root/logs/agent-log
            RefererLog    /usr/lpp/internet/server_root/logs/referer-log
            ErrorLog      /usr/lpp/internet/server_root/logs/httpd-errors
            CgiErrorLog   /usr/lpp/internet/server_root/logs/cgi-error
            LogFormat     Common
            LogTime       LocalTime
            LogToSyslog off
            # Nolog    *.*.*.com
            #   ==================================
            #   *** log purge/archive directives ***
            #   ==================================
            AccessLogArchive none
            ErrorLogArchive  none
            AccessLogExpire 0
            ErrorLogExpire  0
            AccessLogSizeLimit 0
            ErrorLogSizeLimit  0
            #   ==================================
            #   *** access log filter directives ***
            #   ==================================
            AccessLogExcludeUserAgent ICS-ProxyAgent/4.2
            #   ========================================
            #   *** log report configuration directives ***
            #   ========================================
            AccessReportDoDnsLookup off
            AccessReportRoot /usr/lpp/internet/server_root/pub/reports
            #   ============================
            #   *** Default access report ***
            #   ============================
            AccessReportTemplate  Top50  {
            AccessReportDescription    Top 50 most frequently requested files
            AccessReportTopList        50
            ReportProcessOldLogs append
            ReportDataArchive none
            ReportDataExpire 0
```

```
ReportDataSizeLimit 0
DoReporting   on
================================================================== #
#        Method directives
====================================================================
Enable     GET
Enable     HEAD
Enable     POST
Enable     TRACE
Enable     OPTIONS
Disable    PUT
Disable    DELETE
Disable    CONNECT
====================================================================#
#        Directories and Welcome page directives
====================================================================
Welcome     Welcome.html
Welcome     welcome.html
Welcome     index.html
Welcome     Frntpage.html
AlwaysWelcome     on
#   ===================================
#   *** Directory browsing directives ***
#   ===================================
DirAccess             off
DirReadme             top
DirShowIcons          on
DirShowDate           on
DirShowSize           on
DirShowDescription    on
DirShowBrackets       on
DirShowCase           on
DirShowHidden         off
DirShowBytes          off
DirShowMaxDescrLength  25
DirShowMaxLength       25
DirShowMinLength       15

UseMetaFiles off
MetaDir      .web
MetaSuffix   .meta
================================================================== #
#        Error Message Customization directive
====================================================================
# Example:
# ErrorPage scriptstart /errors/html/scriptstart.html
====================================================================#
   User directory directives
====================================================================
```

```
# Example:
# UserDir public
# ------------------------------------------------------------------ #
#        User authentication and document protection
# ------------------------------------------------------------------#
The following rules will allow anyone that knows your WEBADM
# password to use the Web Server remote configuration application.
#
Protection IMW_Admin {
        ServerId        IMWEBSRV_Administration
        AuthType        Basic
        PasswdFile      %%SAF%%
        Mask            WEBADM,webadm
}
Protect /admin-bin/* IMW_Admin WEBADM
Protect /Docs/admin-bin/* IMW_Admin WEBADM
Protect /reports/*   IMW_Admin WEBADM
Protect /Usage*      IMW_Admin WEBADM
#   ============================
#   *** CAServlet & WAS directives ***
#   ============================
Service servlet/* /usr/lpp/WebSphere/AppServer/lib/libadpter.so:AdapterService
#   ============================
#   *** HTImage directives ***
#   ============================
# The following rules invoke the htimage imagemap processor to handle
# image maps in either the CERN format or the NCSA format.
#
service /cgi-bin/htimage*       INTERNAL:HTImage*
service /cgi-bin/imagemap*      INTERNAL:HTImage*
service /Usage*                 INTERNAL:UsageFn
service /admin-bin/trace*       INTERNAL:TraceFn
# ------------------------------------------------------------------
#        WebSphere directives
# ------------------------------------------------------------------
ServerInit /usr/lpp/WebSphere/AppServer/bin/was350plugin.so:init_exit
/usr/lpp/WebSphere,/etc/was.conf
Service webapp/examples/*
/usr/lpp/WebSphere/AppServer/bin/was350plugin.so:service_exit
Service    /servlet/*
/usr/lpp/WebSphere/AppServer/bin/was350plugin.so:service_exit
ServerTerm /usr/lpp/WebSphere/AppServer/bin/was350plugin.so:term_exit
# ------------------------------------------------------------------#
#        Mapping rules
# ------------------------------------------------------------------ #
Pass /admin-bin/webexec/*  /usr/lpp/internet/server_root/admin-bin/webexec/*
Exec            /cgi-bin/*           /usr/lpp/internet/server_root/cgi-bin/*
Exec            /admin-bin/*         /usr/lpp/internet/server_root/admin-bin/*
Exec            /Docs/admin-bin/*    /usr/lpp/internet/server_root/admin-bin/*
```

```
#        URL translation rules; If your documents are under
#   /usr/lpp/internet/server_root/pub/ then this single rule does the job:
#
Pass /icons/*            /usr/lpp/internet/server_root/icons/*
Pass /Admin/*.jpg        /usr/lpp/internet/server_root/Admin/*.jpg
Pass /Admin/*.gif        /usr/lpp/internet/server_root/Admin/*.gif
Pass /Admin/*.html       /usr/lpp/internet/server_root/Admin/*.html
Pass /Docs/*             /usr/lpp/internet/server_root/Docs/*
Pass /reports/javelin/* /usr/lpp/internet/server_root/pub/reports/javelin/*
Pass /reports/java/* /usr/lpp/internet/server_root/pub/reports/java/*
Pass /reports/* /usr/lpp/internet/server_root/pub/reports/*
Pass /img-bin/*          /usr/lpp/internet/server_root/img-bin/*
Pass /CAServlet/* /usr/lpp/internet/server_root/CAServlet/C/*
Pass /ServletExpress/resources/* /usr/lpp/internet/server_root/CAServlet/C/*
Pass /ServletExpress/*  /       usr/lpp/WebSphere/AppServer/web/*
# *** ADD NEW PASS RULES HERE ***
Pass /*                  /usr/lpp/internet/server_root/pub/*
# ------------------------------------------------------------------ #
#        Performance directives.
# ------------------------------------------------------------------ #
```

Check ===>

```
MaxActiveThreads  10000
MaxPersistRequest 5
ServerPriority -10
# ------------------------------------------------------------------ #
#        Timeout directives
# ------------------------------------------------------------------ #
InputTimeout    30 secs
OutputTimeout   2 minutes
ScriptTimeout   2 minutes
PersistTimeout  5 secs
# ------------------------------------------------------------------ #
#        Security directives.
# ------------------------------------------------------------------ #
```

Check ===>

```
SSLMODE on
sslport    443
normalmode  on
keyfile key.kdb
SSLClientAuth   off
SSLX500CARoots local_only
```

Check ===>

```
UseACLs protectonly
SSLV2Timeout 100
SSLV3Timeout 1000
SSLCipherSpec 21
SSLCipherSpec 23
SSLCipherSpec 26
SSLCipherSpec 27
SSLCipherSpec 22
SSLCipherSpec 24
```

```
SSLCipherSpec 34
SSLCipherSpec 35
SSLCipherSpec 3A
SSLCipherSpec 39
SSLCipherSpec 33
SSLCipherSpec 36
MaxContentLengthBuffer   100 K
# ------------------------------------------------------------------ #
#        Proxy caching directives
# ------------------------------------------------------------------ #
Caching off
CacheDefaultExpiry     ftp:*          1 hour
CacheDefaultExpiry     http:*         0 days
CacheDefaultExpiry     ftp:*          1 day
CacheDefaultExpiry     gopher:*       2 days
CacheExpiryCheck on
CacheNoConnect off
CacheTimeMargin  2 minutes
CacheLimit_2 4000 K
CacheSize 5 M
Gc on
GcDailyGc 03:00
GcMemUsage 500
# ------------------------------------------------------------------ #
#        File caching directives
# ------------------------------------------------------------------ #
LiveLocalCache off
CacheLocalMaxFiles 200
CacheLocalMaxBytes 2 M
CacheLocalFile   /usr/lpp/internet/server_root/pub/Frntpage.html
CacheLocalFile   /usr/lpp/internet/server_root/Admin/lgmast.gif
CacheLocalFile   /usr/lpp/internet/server_root/Admin/lgsplash.gif
# ------------------------------------------------------------------ #
#       SMF directives
# ------------------------------------------------------------------ #
SMF     None
SMFRecordingInterval 00:15
# ------------------------------------------------------------------ #
#        File caching directives
# ------------------------------------------------------------------ #
EnableFRCA off
FRCACacheSize 25000
FRCACacheEntries 1000
FRCAMaxFileSize 1 M
FRCAVirtualHost auto
# ------------------------------------------------------------------ #
#        SNMP directives
# ------------------------------------------------------------------ #
SNMP   off
```

Check ===>

Check ===>

Check ===>

```
SNMPCommunity   public
WebMasterEMail   webmaster
# ================================================================== #
#        Icon directives
# ================================================================== #
# Text Icons
AddIcon          /icons/html.gif         HTML text/html
AddIcon          /icons/html.gif         HTML text/x-ssi-html
AddIcon          /icons/text.gif         TXT  text/*
# Image Icons
AddIcon          /icons/image.gif        IMG  image/*
# Audio Icons
AddIcon          /icons/sound.gif        AU   audio/*
AddIcon          /icons/movie.gif        MOV  video/*
# Application Icons
AddIcon          /icons/tar.gif          TAR  multipart/*tar
AddIcon          /icons/compress.gif     CMP  x-compress x-gzip
# Misc.
AddIcon          /icons/telnet.gif       TEL  application/telnet
AddIcon          /icons/index.gif        SRCH application/search
AddIcon          /icons/ls123.gif        123  application/x-123
AddIcon          /icons/lsflw.gif        FLW  application/x-freelance
AddIcon          /icons/acrobat.gif      PDF  application/pdf
AddIcon          /icons/binary.gif       BIN  binary
AddDirIcon       /icons/dir.gif          DIR
AddBlankIcon     /icons/blank.gif
AddParentIcon    /icons/back.gif         UP
AddUnknownIcon   /icons/unknown.gif      ???
# ================================================================== #
#        Request Processing directives
# ================================================================== #
SuffixCaseSense   off
```

F

was.conf

In this appendix, we discuss the was.conf file.

Access the was.conf file

You can access the was.conf file as shown:

```
File  Directory  Special_file  Commands  Help
_____________________________________________________________________________
                              Directory List

Select one or more files with / or action codes. If / is used also select an
action from the action bar otherwise your default action will be used.
Select with S to use your default action.  Cursor select can also be used
for quick navigation.  See help for details.
EUID=287   /usr/lpp/WebSphere/WebServerPlugIn/properties/
  Type  Filename                                              Row 5 of 14
_ Dir   IBM
_ File  media.properties
_ File  was.conf
_ File  was.conf.template
_ File  was.ini
_ File  webapp.xsl
_ File  xmlconfig.xsl
```

Figure F-1 Accessing the was.conf file

Example was.conf file

```
############################################################################
# (C) COPYRIGHT 2000-2001 IBM Corporation. All rights reserved.
#
# ------------------------------------------------------------------------ #
#
#   This file contains the initial configuration properties for the
#   WebSphere Application Server for OS/390 version 4.01.
#
#   This properties file is used during installation verification
#   of the Application Server.
#
#   For a description of the individual properties, see the
#   WebSphere Application Server Planning, Installing, Using Guide.
#
# ------------------------------------------------------------------------ #
appserver.version=4.01
appserver.usesystemclasspath=false
appserver.libpath=
appserver.classpath=
```

```
appserver.name=defaultServletEngine
appserver.jvmpropertiesfile=
appserver.loglevel=WARNING
appserver.logdirectory=
appserver.workingdirectory=
appserver.configviewer=
appserver.jspbasehrefadd=true
appserver.nodetach=false
objectleveltrace.enabled=false
session.enable=true
session.urlrewriting.enable=false
session.cookies.enable=true
session.protocolswitchrewriting.enable=false
session.cookie.name=sesessionid
session.cookie.comment=servlet Session Support
session.cookie.maxage=-1
session.cookie.path=
session.cookie.secure=false
session.invalidationtime=180000
session.tablesize=1000
session.tableoverflowenable=false
session.dbenable=false
session.dbjdbcpoolname=SessionJDBCConnectionPool
session.dbtablename=
session.domain=
#
# ================================================================ #
#   Connection Pool Definitions
# ================================================================ #
jdbcconnpool.default_jdbcpool.waitforconnectiontimeoutmilliseconds=30000
jdbcconnpool.default_jdbcpool.idleconnectiontimeoutmilliseconds=120000
jdbcconnpool.default_jdbcpool.inuseconnectiontimeoutmilliseconds=120000
jdbcconnpool.default_jdbcpool.jdbcdriver=ibm.sql.DB2Driver
jdbcconnpool.default_jdbcpool.databaseurl=
jdbcconnpool.default_jdbcpool.datasourcename=default_pool_datasource
#
#----------------------------------------------------------------------#
jdbcconnpool.SessionJDBCConnectionPool.minconnections=10
jdbcconnpool.SessionJDBCConnectionPool.maxconnections=40
jdbcconnpool.SessionJDBCConnectionPool.waitforconnectiontimeoutmilliseconds=300
00
jdbcconnpool.SessionJDBCConnectionPool.idleconnectiontimeoutmilliseconds=120000
jdbcconnpool.SessionJDBCConnectionPool.inuseconnectiontimeoutmilliseconds=-1
jdbcconnpool.SessionJDBCConnectionPool.jdbcdriver=ibm.sql.DB2Driver
jdbcconnpool.SessionJDBCConnectionPool.databaseurl=
jdbcconnpool.SessionJDBCConnectionPool.connectionidentity=server
#
# ================================================================ #
#   Virtual Host settings
```

```
# ================================================================ #
host.default_host.alias=localhost
host.default_host.mimetypefile=
#
# ================================================================ #
#   Deployed Web Application Settings
# ================================================================ #
#   deployedwebapp.<appname>.host=
#   deployedwebapp.<appname>.rooturi=
#   deployedwebapp.<appname>.classpath=
#   deployedwebapp.<appname>.documentroot=
#   deployedwebapp.<appname>.autoreloadinterval=
#   deployedwebapp.<appname>.authresource=
#   deployedwebapp.<appname>.filter.<MIME_type>=<servletname>
#
# ================================================================ #
#   Web Application Settings
# ================================================================ #
#    webapp.<appname>.description=
#    webapp.<appname>.servletmapping=
#    webapp.<appname>.jspmapping=
#    webapp.<appname>.jsplevel=
#    webapp.<appname>.filemapping=
#    webapp.<appname>.attributes=
#    webapp.<appname>.errorpagemapping=
#
# ================================================================ #
#   Servlet Settings
# ================================================================ #
#    webapp.<appname>.servlet.<servletname>.autostart=
#    webapp.<appname>.servlet.<servletname>.code=
#    webapp.<appname>.servlet.<servletname>.initargs=
#    webapp.<appname>.servlet.<servletname>.servletmapping=
#
# ================================================================ #
#   The following defines the examples web application used for
#   installation verification.
# ================================================================ #

deployedwebapp.examples.host=default_host
deployedwebapp.examples.rooturi=/webapp/examples
deployedwebapp.examples.classpath=$was_install_root$/WebServerPlugIn/hosts/defa
ult_host/examples/servlets
deployedwebapp.examples.documentroot=$was_install_root$/WebServerPlugIn/hosts/d
efault_host/examples/web
webapp.examples.jspmapping=*.jsp
webapp.examples.filemapping=/
webapp.examples.jsplevel=1.1
webapp.examples.servlet.simpleJSP.code=SimpleJSPServlet
```

```
webapp.examples.servlet.simpleJSP.servletmapping=/simpleJSP
webapp.examples.servlet.simpleJSP.servletmapping=/simpleJSP.servlet
webapp.examples.servlet.showCfg.code=com.ibm.servlet.engine.config.ServletEngin
eConfigServlet
webapp.examples.servlet.showCfg.servletmapping=/showCfg
jdbcconnpool.SessionJDBCConnectionPool.minconnections=10
jdbcconnpool.SessionJDBCConnectionPool.maxconnections=40
jdbcconnpool.SessionJDBCConnectionPool.waitforconnectiontimeoutmilliseconds=300
00
jdbcconnpool.SessionJDBCConnectionPool.idleconnectiontimeoutmilliseconds=120000
jdbcconnpool.SessionJDBCConnectionPool.inuseconnectiontimeoutmilliseconds=-1
jdbcconnpool.SessionJDBCConnectionPool.jdbcdriver=ibm.sql.DB2Driver
jdbcconnpool.SessionJDBCConnectionPool.databaseurl=
jdbcconnpool.SessionJDBCConnectionPool.connectionidentity=server
```

G

current.env

Locate the current.env file

The WebSphere Application Server runtime server start procedures point to environment file current.env for JVM and other configuration information.

Each of the WebSphere Application Server start procedures uses a BBOENV DD statement with a PATH parameter that points to the HFS file:

```
//BBOASR2 PROC SRVNAME='BBOASR2A',
//       PARMS=''
//       SET RELPATH='controlinfo/envfile'
//       SET CBCONFIG='/WebSphere390/CB390'
//BBOASR2 EXEC PGM=BBOCTL,REGION=0M,TIME=NOLIMIT,
//        PARM='/ -ORBsrvname &SRVNAME &PARMS'
//BBOENV DD PATH='&CBCONFIG/&RELPATH/&SYSPLEX/&SRVNAME/current.env'
//CEEDUMP   DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//SYSOUT    DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//SYSPRINT  DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//
//* ================================================================ */
//* PROPRIETARY-STATEMENT:                                           */
//* Licensed Material - Property of IBM                              */
//*                                                                  */
//* 5655-F31 (C) Copyright IBM Corp. 2001                            */
```

```
//* All Rights Reserved.                                               */
//* U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or */
//* Disclosure restricted by GSA-ADP schedule contract with IBM Corp.*/
//* Status = H28W400                                                   */
//* ================================================================= */
//*
```

Where:

&CBCONFIG	The WebSphere Application server installation directory; the default is: WebSphere390/CB390.
&RELPATH	This must be: controlinfo/envfile.
&SYSPLEX	The name of your sysplex.
&SRVNAME	The server name (for example, BBOASR2A).

```
File  Directory  Special_file  Commands  Help
_____________________________________________________________________
                            Directory List

Select one or more files with / or action codes. If / is used also select an
action from the action bar otherwise your default action will be used.
Select with S to use your default action.  Cursor select can also be used
for quick  navigation.  See help for details.
EUID=287   /WebSphere390/CB390/controlinfo/envfile/OPPLEX/BBOASR2A/
  Type  Filename                                          Row 1 of 9
 _ Dir   .
 _ Dir   ..
 _ Dir   backup
 _ File  current.env
 _ File  current.env.nodebug
 _ File  current.env.orig
 _ File  jvm.properties
 _ File  trace.dat
 _ File  webcontainer.conf


 Command ===>
_____________________________________________________________________
F1=Help      F3=Exit     F4=Name      F5=Retrieve  F6=Keyshelp  F7=Backward
F8=Forward  F11=Command  F12=Cancel
```

Figure G-1 The current.env file

After the initial bootstrap process, during installation, the WebSphere Application Server writes the environmental data (current.env) into the System Management Database.

Once this happens, you must make all your changes using the Systems Management User Interface (SMUI).

Example current.env file

```
******************************** Top of Data ***************************
# ENVIRONMENT FILE FROM CONVERSATION Change Acords Server identities
#-----------------------------------------------
CLASSPATH=/usr/lpp/WebSphere/lib/ws390srt.jar:/usr/lpp/WebSphere/lib/xerces.jar
:
JVM_LOGFILE=/tmp/jvm.was40.log
DB2SQLJPROPERTIES=/usr/lpp/db2/db2710/classes/db2sqljjdbc.properties
Check ===>
JVM_HEAPSIZE=512
JVM_MINHEAPSIZE=512
TRACEBUFFLOC=SYSPRINT
BBOLANG=ENUS
CBCONFIG=/WebSphere390/CB390
DAEMON_PORT=5555
DAEMON_SSL_PORT=5556
DATASHARING=1
DM_GENERIC_SERVER_NAME=CBDAEMON
DM_SPECIFIC_SERVER_NAME=DAEMON01
ICU_DATA=/usr/lpp/WebSphere/bin
IR_GENERIC_SERVER_NAME=CBINTFRP
IR_SPECIFIC_SERVER_NAME=INTFRP01
IRPROC=BBOIR
IVB_DRIVER_PATH=/usr/lpp/WebSphere
LDAPCONF=/WebSphere390/CB390/OPPLEX/etc/ldap/SC59.bboslapd.conf
LDAPIRCONF=/WebSphere390/CB390/OPPLEX/etc/ldap/SC59.bboslapd.conf
LDAPIRROOT=o=BOSS,c=US
LDAPROOT=o=BOSS,c=US
LIBPATH=/usr/lpp/db2/db2710/lib:/usr/lpp/java/IBM/J1.3/bin:/usr/lpp/java/IBM/J1
LOGSTREAMNAME=WAS.ERROR.LOG
NM_GENERIC_SERVER_NAME=CBNAMING
NM_SPECIFIC_SERVER_NAME=NAMING01
NMPROC=BBONM
OTS_DEFAULT_TIMEOUT=300
OTS_MAXIMUM_TIMEOUT=300
RAS_MINORCODEDEFAULT=NODIAGNOSTICDATA
RESOLVE_IPNAME=wtsc59oe.itso.ibm.com
RESOLVE_PORT=900
SM_DEFAULT_ADMIN=CBADMIN
SM_GENERIC_SERVER_NAME=CBSYSMGT
SM_SPECIFIC_SERVER_NAME=SYSMGT01
SMPROC=BBOSMS
SOMOOSQL=1 SRVIPADDR=None
SYS_DB2_SUB_SYSTEM_NAME=DB7A
```

```
TRACEALL=0
TRACEPARM=00
DAEMON_IPNAME=wtsc59oe.itso.ibm.com
CONFIGURED_SYSTEM=SC59
***************************** Bottom of Data ****************************
```

H

webcontainer.conf

Locate the webcontainer.conf file

```
File  Directory  Special_file  Commands  Help
_____________________________________________________________________
                          Directory List

Select one or more files with / or action codes. If / is used also select an
action from the action bar otherwise your default action will be used.
Select with S to use your default action.  Cursor select can also be used
for quick  navigation.  See help for details.
EUID=287   /WebSphere390/CB390/controlinfo/envfile/OPPLEX/BBOASR2A/
  Type  Filename                                              Row 4 of 9
_ Dir   backup
_ File  current.env
_ File  current.env.nodebug
_ File  current.env.orig
_ File  jvm.properties
_ File  trace.dat
_ File  webcontainer.conf
```

Figure H-1 Locating the webcontainer.conf file

After the initial bootstrap process, during installation, the WebSphere Application Server writes the environmental data (current.env) into the System Management Database. Once this happens, you must make all your changes using the Systems Management User Interface (SMUI).

Example webcontainer.conf file

```
appserver.version=4.01
session.enable=true
session.urlrewriting.enable=false
session.cookies.enable=true
session.protocolswitchrewriting.enable=false
session.cookie.name=sesessionid

_____________________________________________________________________________
session.cookie.comment=<comment>
The default is "WebSphere Session Support".
session.cookie.comment=servlet Session Support
session.cookie.maxage=-1
session.cookie.path=<path>
session.cookie.path=/
session.cookie.secure=false
session.tablesize=<integer>
session.tableoverflowenable=true,
session.tablesize=1000
session.invalidationtime=180000
session.tableoverflowenable=true
session.dbenable=false

_____________________________________________________________________________
session.datasourcename=jdbc/SessionDataSource
session.dbtablename=
session.domain=
Virtual Host settings
host.default_host.alias=localhost
<ahost.default_host.contextroots=/*
host.default_host.mimetypefile=
```

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 on ordering these publications, see “How to get IBM Redbooks” on page 165.

- *Writing Optimized Java Applications for z/OS,* SG24-6541
- e-*business Cookbook for z/OS Volume I: Technology Introduction,* SG24-5664
- *e-business Enablement Cookbook for z/OS Vol. II: Infrastructure,* SG24-5981
- *IBM Host Integration: A Practical Approach to Performance Planning*, SG24-5960
- *Hierarchical File System Usage Guide*, SG24-5482
- *Understanding LDAP*, SG24-4986

Other resources

These publications are also relevant as further information sources:

- *z/OS UNIX System Services User's Guide,* SA22-7801
- *z/OS UNIX System Services Planning,* GA22-7800
- *z/OS UNIX System Services Command Reference*, SA22-7802
- *z/OS HTTP Server Planning, Installing and Using, Version 5.3,* SC34-4826
- *z/OS Language Environment Programming Guide*, SA22-7561
- *z/OS Language Environment Programming Reference*, SA22-7562
- *z/OS Language Environment Customization,* SA22-7564
- *z/OS Communications Server: IP Configuration Guide,* SC31-8775
- *z/OS Communications Server: IP Configuration Reference,* SC31-8776
- *z/OS V1R2.0 MVS Programming: Resource Recovery*, SA22-7616
- *z/OS MVS Planning: Workload Management*, SA22-7602

- *z/OS V1R3.0 MVS System Management Facilities (SMF)*, SA22-7630
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization,* GA22-7834
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Operations and Administration*, SA22-7835
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications,* SA22-7836
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling CORBA Applications,* SA22-7848
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management User Interface,* SA22-7838
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Migration,* GA22-7860
- *IBM TCP/IP Performance Tuning Guide*, SC31-7188
- *MVS Initialization and Tuning Guide*, SC28-1751
- *MVS Initialization and Tuning Reference*, SC28-1752
- *MVS System Management Facilities*, GC28-1763
- *OS/390 RMF Report Analysis*, SC28-1950
- *OS/390 MVS System Management Facilities*, GC28-1783
- *OS/390 eNetwork Communication Server: IP User's Guide and IP Configuration Guide*, SC31-8513
- *HTTP Server Planning, Installing, and Using for OS/390*, SC31-8690
- *MVS TCP/IP User's Guide*, SC31-7136
- *Resource Measurement Facility User 's Guide,* SC33-7990
- *z/OS V1R2 SecureWay Security Server LDAP Server Administration*, SC24-5923

Referenced Web sites

These Web sites are also relevant as further information sources:

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

  http://www.ibm.com/software/webservers/appserv/zos_os390/
- z/OS USS Performance

  http://www.ibm.com/servers/eserver/zseries/zos/unix/bpxa1tun.html

- WebSphere for z/OS and OS/390 hints and tips

  http://www.ibm.com/software/webservers/httpservers/doc/v51/ttunc.htm

- S/390 eBusiness Performance

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

How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

ibm.com/redbooks

You can also download additional materials (code samples or diskette/CD-ROM images) from that site.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.

Index

IBM Redbooks

z/OS for e-business: An Introduction to System Tuning

(0.2"spine)
0.17"<->0.473"
90<->249 pages

z/OS for e-business: An Introduction to System Tuning

Set up the z/OS infrastructure for e-business and Java applications

Tune the HTTP Server

Tune the WebSphere Application Server

This IBM Redbook provides a high-level overview of what is needed to set up a z/OS configuration in an e-business environment. It consolidates some of the tuning information for the WebSphere Application Server and z/OS operating system environments in one place.

In addition, since system environments change over time and you need to proactively prevent performance problems in today's e-business world, it covers aspects of monitoring your system's performance.

This publication represents an introductory work on this subject, so to guide your further understanding we reference other IBM Redbooks, Web sites, and IBM documentation that contain more detailed pieces of the overall picture of what is needed to successfully set up and tune the WebSphere Application Server and supporting z/OS operating system.

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

SG24-6542-00 ISBN 0738424536