

SAS 9.2 Web Applications: Tuning for Performance and Scalability



Copyright Notice

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS 9.2 Web Applications: Tuning for Performance and Scalability*, Cary, NC: SAS Institute Inc., 2009.

SAS 9.2 Web Applications: Tuning for Performance and Scalability

Copyright © 2009, SAS Institute Inc., Cary, NC, USA.

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, by any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc. Limited permission is granted to store the copyrighted material in your system and display it on terminals, print only the number of copies required for use by those persons responsible for installing and supporting the SAS programming and licensed programs for which this material has been provided, and to modify the material to meet specific installation requirements. The SAS Institute copyright notice must appear on all printed versions of this material or extracts thereof and on the display medium when the material is displayed. Permission is not granted to reproduce or distribute the material except as stated above.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of the software by the government is subject to restrictions as set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.

® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

Table of Contents

Chapter 1 — Overview	1
Establish Your Performance Objective	1
Common JVM Options	3
JBoss JVM Options	4
SAS Remote Services JVM Options	4
Chapter 2 — Scaling the Entire SAS Middle Tier	5
Pattern 1: Tuning Parameters for HP and Sun JDKs	6
JVM #1: SAS Web Report Studio	6
JVM #2: SAS Information Delivery Portal and SAS Package Viewer	6
JVM #3: SAS Web OLAP Viewer for Java	6
JVM #4: SAS BI Dashboard	7
JVM #5: SAS Web Infrastructure Platform	7
JVM #6: SAS Content Server	7
JVM #7: SAS Remote Services	7
Pattern 1: Tuning Parameters for 32-bit Operating Systems and an IBM JDK	8
JVM #1: SAS Web Report Studio	8
JVM #2: SAS Information Delivery Portal	8
JVM #3: SAS Web OLAP Viewer for Java	8
JVM #4: SAS BI Dashboard	8
JVM #5: SAS Web Infrastructure Platform	8
JVM #6: SAS Content Server	8
JVM #7: SAS Remote Services	8
Chapter 3 — Scaling SAS Information Delivery Portal	9
Pattern 2: Tuning Parameters for HP and Sun JDKs	9
JVM #1: SAS Information Delivery Portal and SAS Web Infrastructure Platform Applications	9
JVM #2 SAS Web Infrastructure Platform Services and SAS Package Viewer	10
JVM #3 SAS Content Server	10
JVM #4: SAS Remote Services	10
Pattern 2 Tuning Parameters for an IBM JDK	10
JVM #1 SAS Information Delivery Portal and SAS Web Infrastructure Platform Applications.	10
JVM #2 SAS Web Infrastructure Platform Services and SAS Package Viewer	11
JVM #3 SAS Content Server	11
JVM #4: SAS Remote Services	11
Chapter 4 — Scaling SAS Web Report Studio	12
Pattern 3: Tuning Parameters for Sun and HP JDKs	12
JVM #1 SAS Web Report Studio	12
JVM #2 SAS Content Server and SAS Web Infrastructure Platform	13
JVM #3: SAS Remote Services	13
Pattern 3: Tuning Parameters for an IBM JDK	13
JVM #1 SAS Web Report Studio	13
JVM #2 SAS Content Server and SAS Web Infrastructure Platform	13
JVM #3: SAS Remote Services	13
Chapter 5 – Scaling Web OLAP Viewer for Java	14
Pattern 4: Tuning Parameters for HP and Sun JDKs	14
JVM #1 SAS Web OLAP Viewer for Java and SAS Web Infrastructure Platform Applications..	14

JVM #2 SAS Content Server, SAS Web Infrastructure Platform Services, and SAS Shared Services.....	15
JVM #3: SAS Remote Services	15
Pattern 4: Tuning Parameters for an IBM JDK	15
JVM #1 SAS Web OLAP Viewer for Java and SAS Web Infrastructure Platform Applications..	15
JVM #2 SAS Content Server, SAS Web Infrastructure Platform Services, and SAS Shared Services.....	15
JVM #3: SAS Remote Services	15
Chapter 6 – Scaling SAS BI Dashboard	16
Pattern 5: Tuning Parameters for an HP or Sun JDK	16
JVM #1: SAS Web Report Studio	16
JVM #2: SAS Information Delivery Portal and SAS Package Viewer	17
JVM #3: SAS Web Infrastructure Platform.....	17
JVM #4: SAS BI Dashboard and SAS Shared Services.....	17
JVM #5: SAS Content Server	18
JVM #6: SAS Remote Services	18
Pattern 5: Tuning Parameters for an IBM JDK	18
JVM #1: SAS Web Report Studio	18
JVM #2: SAS Information Delivery Portal and SAS Package Viewer	18
JVM #3: SAS Web Infrastructure Platform.....	18
JVM #4: SAS BI Dashboard and SAS Shared Services.....	18
JVM #5: SAS Content Server	19
JVM #6: SAS Remote Services	19
Chapter 7 – Reconfiguring the SAS 9.2 Middle Tier.....	20
Stage 1 – Plan the SAS 9.2 Middle Tier Application and JVM Distribution	20
Example 1 – Three JVMs	20
Example 2 – Four JVMs	21
Stage 2 – Prepare and Configure the Web Application Server Environment .	21
Stage 3 – Deploy SAS Themes to an HTTP Server	22
Stage 4 – Change the Connections for the SAS Web Applications	22
Stage 5 – Change the Connection for SAS Content Server	22
Stage 6 – Change the WebDAV Repository URL	23
Chapter 8 – SAS Application Tuning Parameters	24
Chapter 9 – Web Application Server Tuning Parameters	26
Chapter 10 – Java Virtual Machine Tuning Parameters.....	27
Setting JVM Options for JBoss	28
Setting JVM Options for WebLogic Server.....	28
Setting JVM Options for WebSphere Application Server	29
Chapter 11 – Operating System Tuning Parameters	30
Solaris SPARC and Solaris x86-64	30
Linux	30
AIX.....	30
HP-UX.....	30
Windows	32

Chapter 1 — Overview

The SAS 9.2 middle tier environment provides an execution environment for SAS Web applications that are used to perform business analytics. Optimizing the middle tier and SAS Web applications for scalability and performance requires careful planning and effort to carry out the plan. The planning and execution can be complex as they balance the demands for availability, reliability, security and performance. The purpose of this document is to walk through the SAS 9.2 tuning for performance and scalability. This document discusses the tuning methodology and tuning parameters that are involved in SAS Web applications, the Web application server, Java Virtual Machines (JVM), and operating systems.

Establish Your Performance Objective

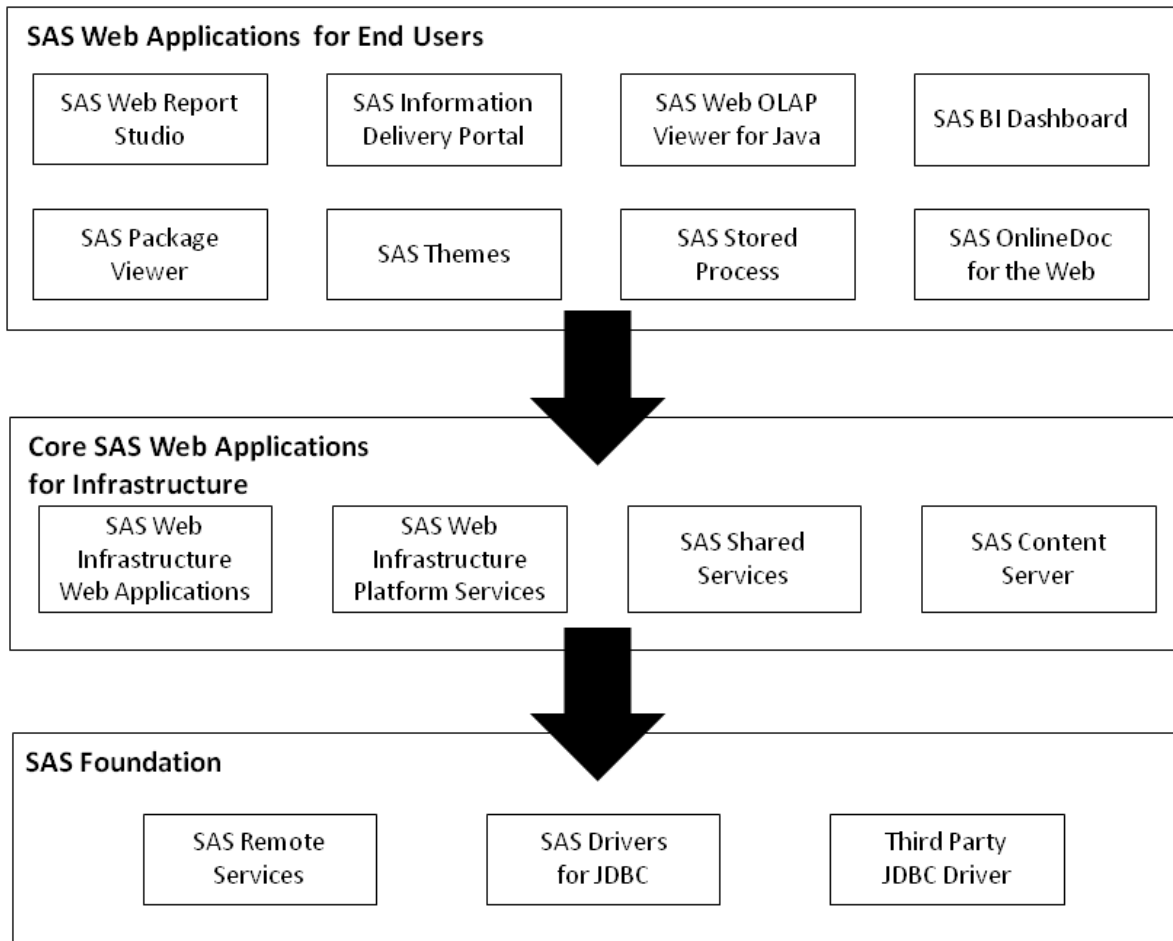
Establishing your performance objective is the most important step of the entire process for tuning the performance and scalability of the middle tier. Performance requirements are usually identified in terms of transaction response time, number of transactions per second, throughput time, resource utilization, total cost per transaction, availability, and more. Scalability often refers to a component's ability to adapt readily to a greater or lesser intensity of use, volume, or demand while meeting integral business objectives. The common objective of scaling a component or system is to increase the capacity for growth, increase the speed of the component, improve the efficiency, or shift or reduce the load on the component.

Tuning for performance and scalability is approached from the perspective of enhancing the end-user experience with the following SAS Web applications:

- SAS Web Report Studio
- SAS Information Delivery Portal
- SAS Web OLAP Viewer
- SAS BI Dashboard

This document provides an approach for improving performance and scalability based on common patterns. The end-user Web applications make use of Web infrastructure software, and the Web infrastructure software makes use of SAS Foundation software. Each of these layers has an impact on performance. Each pattern in this document provides guidelines on how to tune the software in the lower layers to enhance the user experience with the selected end-user Web application. The following figure shows the software layering described in this paragraph. However, this figure is not a detailed architecture diagram of SAS software; it is a visual aid to assist with tuning the SAS Web applications.

Figure 1 SAS Middle Tier Software Layers



The following table categorizes the scalability and performance objectives for the end-user SAS Web applications. The first column identifies the tuning patterns that are provided in this document.

Usage Pattern	SAS Information Delivery Portal	SAS Web Report Studio	SAS BI Dashboard	SAS Web OLAP Viewer for Java
Scaling the entire SAS middle tier	Moderate to Aggressive	Moderate to Aggressive	Moderate to Aggressive	Moderate to Aggressive
Scaling SAS Information Delivery Portal	Aggressive	None or Low	None or Low	None or Low
Scaling SAS Web Report Studio	None, Low, or Moderate	Aggressive	None, Low, or Moderate	None, Low, or Moderate
Scaling SAS BI Dashboard	None, Low, or Moderate	None, Low, or Moderate	Aggressive	None, Low, or Moderate
Scaling SAS Web OLAP Viewer for Java	None, Low, or Moderate	None, Low, or Moderate	None, Low, or Moderate	Aggressive

The default configuration of SAS 9.2 Web applications in a Web application server environment and its operating system is sufficient in most cases to meet moderate and low classifications of performance and scalability requirements.

Much of the performance and scalability improvements for each pattern are accomplished by adjusting the JVM options for the Web application server. For each pattern, the tuning options are provided according to the JDK vendor:

- Sun and HP JDKs – These JDKs use the same JVM options.
- IBM JDK – An IBM JDK is used for all Web application servers deployed on AIX and is also used for IBM WebSphere Application Server, unless WebSphere Application Server is deployed on a Sun Solaris or HP-UX Itanium platform.

For more information about the supported Web application servers and the supported operating systems and JDKs, see the [Third Party Software for SAS 9.2 Foundation](#) Web site. This document provides suggested JVM options that will assist you in meeting your scalability and performance objectives. Once you have applied the JVM options to your topology, perform testing to confirm that your objectives have been met, and adjust the JVM options accordingly.

Common JVM Options

The following list identifies the common JVM options that apply to all Web application servers that host the SAS 9.2 Web applications. In the later sections of this document, JVM options that are specific to a JDK or application are provided. Make sure that the common options provided here are also included when you reconfigure the Web application server:

```
-Dsas.container.identifier=jboss/weblogic/websphere
-Djava.awt.headless=true
-Dsun.rmi.transport.tcp.connectionPool=true
-Dsun.rmi.transport.connectionTimeout=2000
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000
-Dsas.auto.publish.port=web_container_port_of_wip
-Dcom.sas.services.logging.disableRemoteList=true
-Dcom.sas.services.logging.disableRemoteLogging=true
-Djava.net.preferIPv4Stack=true
-Djava.net.preferIPv6Addresses=false
-Dmulticast_udp_ip_ttl=30
-Dmulticast.address=multicast_host_address
-Dmulticast.port=multicast_port
-Dcom.sas.log.config.url="file:///SAS-config-dir/Config/Lev1/Web/
Common/LogConfig/"
-Dsas.svcs.http.max.connections=50
```

JBoss JVM Options

The following list identifies the JVM options that apply only to JBoss and must be added to the other JVM tuning options for any deployments that use JBoss:

```
-Djboss.platform.mbeanserver  
-Djavax.management.builder.initial=  
org.jboss.system.server.jmx.MBeanServerBuilderImpl  
-Djboss.home.dir="JBOSS_HOME"  
-Djboss.server.home.dir="JBOSS_HOME\server\SASServer1"  
-Dsas.deploy.dir="JBOSS_HOME\server\SASServer1\deploy_sas"
```

SAS Remote Services JVM Options

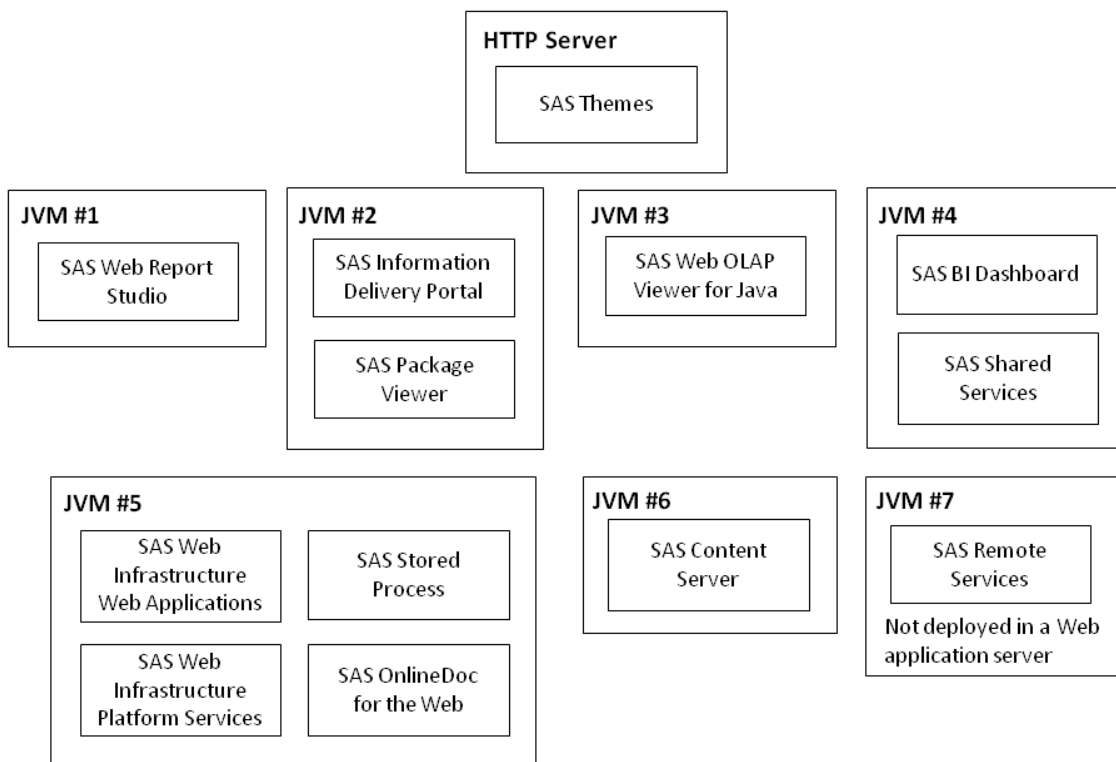
SAS Remote Services runs in its own JVM that is provided by a 32-bit Sun JDK. The common JVM options mentioned earlier in this section do not apply for SAS Remote Services. SAS Remote Services is used across all patterns, and the following example shows the JVM tuning options that should be used in all cases, regardless of the pattern:

```
-Xms512m -Xmx512m -XX:PermSize=64m -XX:MaxPermSize=64m -Xss120k  
-XX:-UseTLAB -XX:+UseConcMarkSweepGC -XX:NewSize=64m -XX:MaxNewSize=64m  
-Dcom.sun.management.jmxremote -Djava.net.preferIPv4Stack=true
```

Chapter 2 — Scaling the Entire SAS Middle Tier

This is pattern 1 and it describes how to tune the SAS middle tier to achieve moderate to aggressive performance and scalability objectives for the four key end-user SAS Web applications. In this pattern, each of the four Web applications is run in its own JVM. In this configuration, you can have all the servers on a single physical machine, or multiple physical machines. The decision to use more than one machine depends on the capacity of the machine. Some contributing factors include the number of CPUs on the machine, RAM, CPU clock speed, and L2 cache size. The following figure shows the application topology that scales the entire SAS 9.2 middle tier.

Figure 2 Application Topology for Scaling the Entire SAS Middle Tier



Pattern 1: Tuning Parameters for HP and Sun JDKs

Note: The Common JVM Options on page 3 must be added along with the following application-specific JVM options.

JVM #1: SAS Web Report Studio

32-bit: -server -Xms1280m -Xmx1280m -XX:PermSize=128m
-XX:MaxPermSize=128m -Xss128k -XX:NewSize=128m -XX:MaxNewSize=128m
-XX:+UseConcMarkSweepGC -XX:+UseCMSInitiatingOccupancyOnly
-XX:CMSInitiatingOccupancyFraction=40 -XX:MaxTenuringThreshold=4
-XX:SurvivorRatio=4 -XX:-UseTLAB -XX:+DisableExplicitGC

64-bit: -server -d64 -Xms1280m -Xmx2048m -XX:PermSize=256m
-XX:MaxPermSize=256m -Xss256k -XX:NewSize=128m -XX:MaxNewSize=128m
-XX:+UseConcMarkSweepGC -XX:+UseCMSInitiatingOccupancyOnly
-XX:CMSInitiatingOccupancyFraction=40 -XX:MaxTenuringThreshold=4
-XX:SurvivorRatio=4 -XX:-UseTLAB -XX:+DisableExplicitGC

JVM #2: SAS Information Delivery Portal and SAS Package Viewer

32-bit: -server -Xms1280m -Xmx1280m -XX:PermSize=128m
-XX:MaxPermSize=128m -Xss160k -XX:-UseTLAB -XX:+UseConcMarkSweepGC
-XX:+DisableExplicitGC -XX:NewSize=128m -XX:MaxNewSize=128m
-XX:+UseCMSInitiatingOccupancyOnly
-XX:CMSInitiatingOccupancyFraction=30 -XX:-UseBiasedLocking
-XX:MaxTenuringThreshold=4 -XX:SurvivorRatio=4

64-bit: -server -d64 -Xms1280m -Xmx2048m -XX:PermSize=256m
-XX:MaxPermSize=256m -Xss256k -XX:-UseTLAB -XX:+UseConcMarkSweepGC
-XX:+DisableExplicitGC -XX:NewSize=128m -XX:MaxNewSize=128m
-XX:+UseCMSInitiatingOccupancyOnly
-XX:CMSInitiatingOccupancyFraction=30 -XX:-UseBiasedLocking
-XX:MaxTenuringThreshold=4 -XX:SurvivorRatio=4

JVM #3: SAS Web OLAP Viewer for Java

32-bit: -server -Xms1024m -Xmx1024m -XX:PermSize=320m
-XX:MaxPermSize=320m -XX:NewSize=128m -XX:MaxNewSize=128m -Xss128k
-XX:+UseConcMarkSweepGC -XX:-UseTLAB -XX:+DisableExplicitGC
-XX:+CMSClassUnloadingEnabled -XX:+CMSPermGenSweepingEnabled
-Dcom.sas.iom.orb.SelectionHub.clientThreadCount=3

64-bit: -server -d64 -Xms1024m -Xmx2048m -XX:PermSize=320m
-XX:MaxPermSize=320m -XX:NewSize=256m -XX:MaxNewSize=256m -Xss256k
-XX:+UseConcMarkSweepGC -XX:-UseTLAB -XX:+DisableExplicitGC
-XX:+CMSClassUnloadingEnabled -XX:+CMSPermGenSweepingEnabled
-Dcom.sas.iom.orb.SelectionHub.clientThreadCount=3

JVM #4: SAS BI Dashboard

32-bit: `-server -Xms1024m -Xmx1024m -XX:PermSize=320m -XX:MaxPermSize=320m
-XX:NewSize=128m -XX:MaxNewSize=128m -Xss128k -XX:+UseConcMarkSweepGC
-XX:-UseTLAB -XX:+DisableExplicitGC -XX:+CMSCClassUnloadingEnabled
-XX:+CMSPermGenSweepingEnabled
-Dcom.sas.iom.orb.SelectionHub.clientThreadCount=3`

64-bit: `-server -d64 -Xms1024m -Xmx2048m -XX:PermSize=320m
-XX:MaxPermSize=320m -XX:NewSize=256m -XX:MaxNewSize=256m -Xss256k
-XX:+UseConcMarkSweepGC -XX:-UseTLAB -XX:+DisableExplicitGC
-XX:+CMSCClassUnloadingEnabled -XX:+CMSPermGenSweepingEnabled
-Dcom.sas.iom.orb.SelectionHub.clientThreadCount=3`

JVM #5: SAS Web Infrastructure Platform

This JVM hosts the SAS Web Infrastructure Platform Applications, SAS Web Infrastructure Platform Services, SAS Shared Services, SAS Stored Process, and SAS OnlineDoc for the Web applications.

32-bit: `-server -Xms1280m -Xmx1280m -XX:PermSize=128m -XX:MaxPermSize=128m
-Xss128k -XX:NewSize=128m -XX:MaxNewSize=128m -XX:+UseConcMarkSweepGC
-XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=40
-XX:MaxTenuringThreshold=4 -XX:SurvivorRatio=4 -XX:-UseTLAB
-XX:+DisableExplicitGC`

64-bit: `-server -d64 -Xms1280m -Xmx2048m -XX:PermSize=128m
-XX:MaxPermSize=128m -Xss256k -XX:NewSize=256m -XX:MaxNewSize=256m
-XX:+UseConcMarkSweepGC -XX:+UseCMSInitiatingOccupancyOnly
-XX:CMSInitiatingOccupancyFraction=40 -XX:MaxTenuringThreshold=4
-XX:SurvivorRatio=4 -XX:-UseTLAB -XX:+DisableExplicitGC`

JVM #6: SAS Content Server

32-bit: `-server -Xms1280m -Xmx1280m -XX:PermSize=96m -XX:MaxPermSize=96m
-Xss160k -XX:-UseTLAB -XX:+UseConcMarkSweepGC
-XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=40
-XX:+DisableExplicitGC -XX:NewSize=128m -XX:MaxNewSize=128m`

64-bit: `-server -Xms1280m -Xmx2048m -XX:PermSize=96m -XX:MaxPermSize=96m
-Xss256k -XX:-UseTLAB -XX:+UseConcMarkSweepGC
-XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=40
-XX:+DisableExplicitGC -XX:NewSize=256m -XX:MaxNewSize=256m`

JVM #7: SAS Remote Services

SAS Remote Services runs in its own JVM that is provided by a 32-bit Sun JDK.

`-Xms512m -Xmx512m -XX:PermSize=64m -XX:MaxPermSize=64m -Xss120k
-XX:-UseTLAB -XX:+UseConcMarkSweepGC -XX:NewSize=64m -XX:MaxNewSize=64m
-Dcom.sun.management.jmxremote -Djava.net.preferIPv4Stack=true`

Pattern 1: Tuning Parameters for 32-bit Operating Systems and an IBM JDK

This series of tuning parameters uses the topology identified in pattern #1. The parameters are suited to a Web application server that is using an IBM JDK.

Note: The Common JVM Options on page 3 must be added along with the following application-specific JVM options.

JVM #1: SAS Web Report Studio

32-bit: -Xms1280m -Xmx1280m -Xss256k -Xmso256k -Xgcpolicy:gencon

64-bit: -Xms1280m -Xmx2048m -Xss256k -Xmso256k -Xgcpolicy:gencon

JVM #2: SAS Information Delivery Portal

32-bit: -Xms1280m -Xmx1280m -Xss256k -Xmso256k -Xgcpolicy:gencon

64-bit: -Xms1280m -Xmx2048m -Xss256k -Xmso256k -Xgcpolicy:gencon

JVM #3: SAS Web OLAP Viewer for Java

32-bit: -Xms1280m -Xmx1280m -Xss256k -Xmso256k -Xgcpolicy:gencon

64-bit: -Xms1280m -Xmx2048m -Xss256k -Xmso256k -Xgcpolicy:gencon

JVM #4: SAS BI Dashboard

32-bit: -Xms1024m -Xmx1024m -Xss256k -Xmso256k -Xgcpolicy:gencon

64-bit: -Xms1024m -Xmx2048m -Xss256k -Xmso256k -Xgcpolicy:gencon

JVM #5: SAS Web Infrastructure Platform

This JVM hosts the SAS Web Infrastructure Platform Applications, SAS Web Infrastructure Platform Services, SAS Shared Services, SAS Stored Process, and SAS OnlineDoc for the Web applications.

32-bit: -Xms1280m -Xmx1280m -Xss256k -Xmso256k -Xgcpolicy:gencon

64-bit: -Xms1280m -Xmx2048m -Xss256k -Xmso256k -Xgcpolicy:gencon

JVM #6: SAS Content Server

32-bit: -Xms1280m -Xmx1280m -Xss256k -Xmso256k -Xgcpolicy:gencon

64-bit: -Xms1280m -Xmx2048m -Xss256k -Xmso256k -Xgcpolicy:gencon

JVM #7: SAS Remote Services

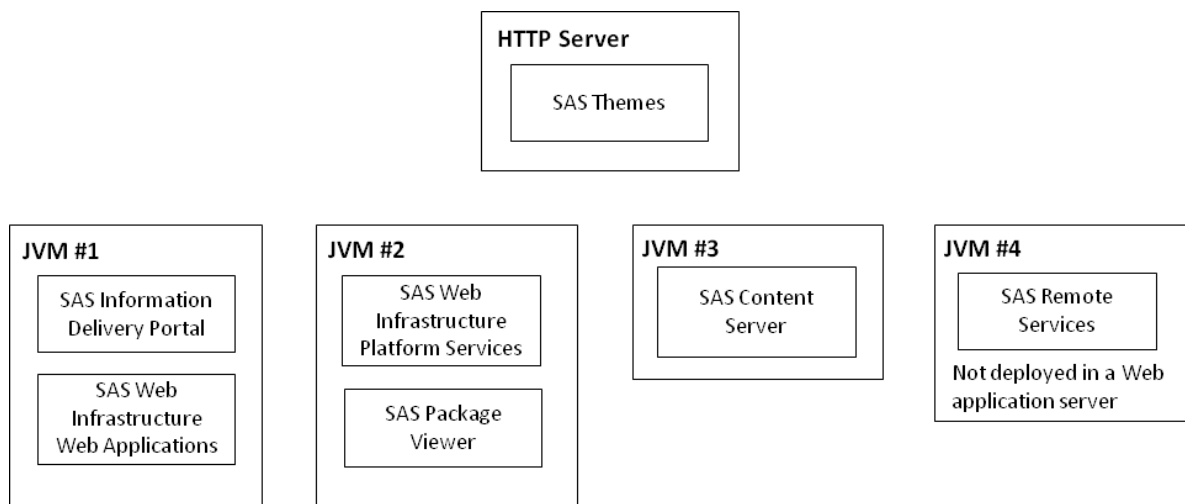
SAS Remote Services runs in its own JVM that is provided by a 32-bit Sun JDK.

-Xms512m -Xmx512m -XX:PermSize=64m -XX:MaxPermSize=64m -Xss120k
-XX:-UseTLAB -XX:+UseConcMarkSweepGC -XX:NewSize=64m -XX:MaxNewSize=64m
-Dcom.sun.management.jmxremote -Djava.net.preferIPv4Stack=true

Chapter 3 — Scaling SAS Information Delivery Portal

This is pattern 2 and describes how to configure the SAS middle tier to focus on the SAS Information Delivery Portal Web application. The application is scaled for high performance requirements such as a large number of users, low response time, and high throughput. In this topology, the SAS Information Delivery Portal and the SAS Web Infrastructure Platform Applications are deployed together in the same JVM. SAS Web Infrastructure Platform Services and SAS Package Viewer are deployed in another JVM. SAS Remote Services uses its own JVM. SAS Themes is deployed in an HTTP Server. The following figure shows the Web application topology that scales the SAS Information Delivery Portal application.

Figure 3 Application Topology for Scaling SAS Information Delivery Portal



Pattern 2: Tuning Parameters for HP and Sun JDKs

Note: The Common JVM Options on page 3 must be added along with the following application-specific JVM options.

JVM #1: SAS Information Delivery Portal and SAS Web Infrastructure Platform Applications

```
32-bit: -server -Xms1280m -Xmx1280m -XX:PermSize=128m -XX:MaxPermSize=128m
-Xss160k -XX:-UseTLAB -XX:+UseConcMarkSweepGC -XX:+DisableExplicitGC
-XX:NewSize=128m -XX:MaxNewSize=128m -XX:CMSInitiatingOccupancyFraction=30
-XX:-UseBiasedLocking -XX:MaxTenuringThreshold=4 -XX:SurvivorRatio=4
-XX:+UseCMSInitiatingOccupancyOnly
```

```
64-bit: -server -d64 -Xms1024m -Xmx2048m -XX:PermSize=512m
-XX:MaxPermSize=512m -Xss256k -XX:NewSize=128m -XX:MaxNewSize=256m
-XX:+UseConcMarkSweepGC -XX:+UseTLAB -XX:+DisableExplicitGC
-XX:+CMSIncrementalMode -XX:+CMSClassUnloadingEnabled
-XX:+CMSPermGenSweepingEnabled
```

JVM #2 SAS Web Infrastructure Platform Services and SAS Package Viewer

32 bit: -server -Xms1280m -Xmx1280m -XX:PermSize=96m -XX:MaxPermSize=96m
-Xss160k -XX:-UseTLAB -XX:+UseConcMarkSweepGC
-XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=40
-XX:+DisableExplicitGC -XX:NewSize=128m -XX:MaxNewSize=128m

64 bit: -server -d64 -Xms1280m -Xmx1280m -XX:PermSize=96m -XX:MaxPermSize=96m
-Xss160k -XX:-UseTLAB -XX:+UseConcMarkSweepGC
-XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=40
-XX:+DisableExplicitGC -XX:NewSize=128m -XX:MaxNewSize=128m

JVM #3 SAS Content Server

32 bit: -server -Xms1280m -Xmx1280m -XX:PermSize=128m -XX:MaxPermSize=128m
-Xss128k -XX:NewSize=128m -XX:MaxNewSize=128m -XX:+UseConcMarkSweepGC
-XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=40
-XX:MaxTenuringThreshold=4 -XX:SurvivorRatio=4 -XX:-UseTLAB
-XX:+DisableExplicitGC

64 bit: -server -d64 -Xms1024m -Xmx2048m -XX:PermSize=512m
-XX:MaxPermSize=512m -Xss256k -XX:NewSize=128m -XX:MaxNewSize=256m
-XX:+UseConcMarkSweepGC -XX:+UseTLAB -XX:+DisableExplicitGC
-XX:+CMSIncrementalMode -XX:+CMSClassUnloadingEnabled
-XX:+CMSPermGenSweepingEnabled

JVM #4: SAS Remote Services

SAS Remote Services runs in its own JVM that is provided by a 32-bit Sun JDK.

-Xms512m -Xmx512m -XX:PermSize=64m -XX:MaxPermSize=64m -Xss120k
-XX:-UseTLAB -XX:+UseConcMarkSweepGC -XX:NewSize=64m -XX:MaxNewSize=64m
-Dcom.sun.management.jmxremote -Djava.net.preferIPv4Stack=true

Pattern 2 Tuning Parameters for an IBM JDK

Note: The Common JVM Options on page 3 must be added along with the following application-specific JVM options.

JVM #1 SAS Information Delivery Portal and SAS Web Infrastructure Platform Applications

32 bit: -Xms1280m -Xmx1280m -Xss256k -Xmso256k -Xgcpolicy:gencon

64 bit: -Xms1024m -Xmx4096m -Xss256k -Xmso256k -Xgcpolicy:gencon

JVM #2 SAS Web Infrastructure Platform Services and SAS Package Viewer

32 bit: -Xms1280m -Xmx1280m -Xss256k -Xmso256k -Xgcpolicy:gencon

64 bit: -Xms1024m -Xmx3072m -Xss256k -Xmso256k -Xgcpolicy:gencon

JVM #3 SAS Content Server

32 bit: -Xms1280m -Xmx1280m -Xss256k -Xmso256k -Xgcpolicy:gencon

64 bit: -Xms1280m -Xmx1280m -Xss256k -Xmso256k -Xgcpolicy:gencon

JVM #4: SAS Remote Services

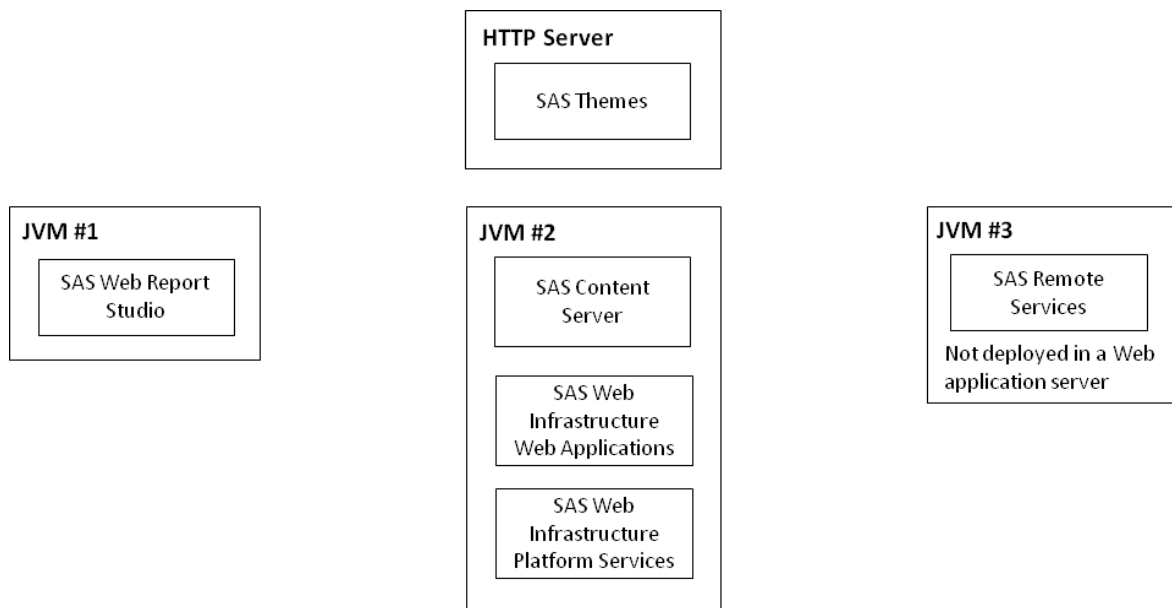
SAS Remote Services runs in its own JVM that is provided by a 32-bit Sun JDK.

-Xms512m -Xmx512m -XX:PermSize=64m -XX:MaxPermSize=64m -Xss120k
-XX:-UseTLAB -XX:+UseConcMarkSweepGC -XX:NewSize=64m -XX:MaxNewSize=64m
-Dcom.sun.management.jmxremote -Djava.net.preferIPv4Stack=true

Chapter 4 — Scaling SAS Web Report Studio

This is pattern 3 and describes how to configure the SAS middle tier to focus on the SAS Web Report Studio application. This application is the focus for high performance requirements such as a large number of users, fast response and high throughput. The SAS Content Server, SAS Web Infrastructure Applications, and SAS Web Infrastructure Services are deployed in one JVM. SAS Web Report Studio is deployed in another JVM. SAS Remote Services is also in its own JVM, and SAS Themes is deployed in the HTTP server. The following figure shows the Web application topology that scales the SAS Web Report Studio application

Figure 4 Application Topology for Scaling SAS Web Report Studio



Pattern 3: Tuning Parameters for Sun and HP JDKs

Note: The Common JVM Options on page 3 must be added along with the following application-specific JVM options.

JVM #1 SAS Web Report Studio

32 bit: `-server -Xms1280m -Xmx1280m -XX:PermSize=128m -XX:MaxPermSize=128m -Xss128k -XX:NewSize=128m -XX:MaxNewSize=128m -XX:+UseConcMarkSweepGC -XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=40 -XX:MaxTenuringThreshold=4 -XX:SurvivorRatio=4 -XX:-UseTLAB -XX:+DisableExplicitGC`

64 bit: `-server -d64 -Xms2048m -Xmx2048m -XX:PermSize=256m -XX:MaxPermSize=256m -Xss256k -XX:NewSize=256m -XX:MaxNewSize=256m -XX:+UseConcMarkSweepGC -XX:+UseTLAB -XX:+DisableExplicitGC`

JVM #2 SAS Content Server and SAS Web Infrastructure Platform

This JVM hosts the SAS Content Server, SAS Web Infrastructure Platform Applications, and SAS Web Infrastructure Platform Services.

32 bit: `-server -Xms1280m -Xmx1280m -XX:PermSize=128m -XX:MaxPermSize=128m
-Xss128k -XX:NewSize=128m -XX:MaxNewSize=128m -XX:+UseConcMarkSweepGC
-XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=40
-XX:MaxTenuringThreshold=4 -XX:SurvivorRatio=4 -XX:-UseTLAB
-XX:+DisableExplicitGC`

64 bit: `-server -d64 -Xms2048m -Xmx2048m -XX:PermSize=256m
-XX:MaxPermSize=256m -Xss256k -XX:NewSize=256m -XX:MaxNewSize=256m
-XX:+UseConcMarkSweepGC -XX:+UseTLAB -XX:+DisableExplicitGC`

JVM #3: SAS Remote Services

SAS Remote Services runs in its own JVM that is provided by a 32-bit Sun JDK.

`-Xms512m -Xmx512m -XX:PermSize=64m -XX:MaxPermSize=64m -Xss120k
-XX:-UseTLAB -XX:+UseConcMarkSweepGC -XX:NewSize=64m -XX:MaxNewSize=64m
-Dcom.sun.management.jmxremote -Djava.net.preferIPv4Stack=true`

Pattern 3: Tuning Parameters for an IBM JDK

Note: The Common JVM Options on page 3 must be added along with the following application-specific JVM options.

JVM #1 SAS Web Report Studio

32 bit: `-Xms1280m -Xmx1280m -Xss256k -Xmso256k -Xgcpolicy:gencon`

64 bit: `-Xms2048m -Xmx2048m -Xss256k -Xmso256k -Xgcpolicy:gencon`

JVM #2 SAS Content Server and SAS Web Infrastructure Platform

This JVM hosts the SAS Content Server, SAS Web Infrastructure Platform Applications, and SAS Web Infrastructure Platform Services.

32 bit: `-Xms1280m -Xmx1280m -Xss256k -Xmso256k -Xgcpolicy:gencon`

64 bit: `-Xms1024m -Xmx3072m -Xss256k -Xmso256k -Xgcpolicy:gencon`

JVM #3: SAS Remote Services

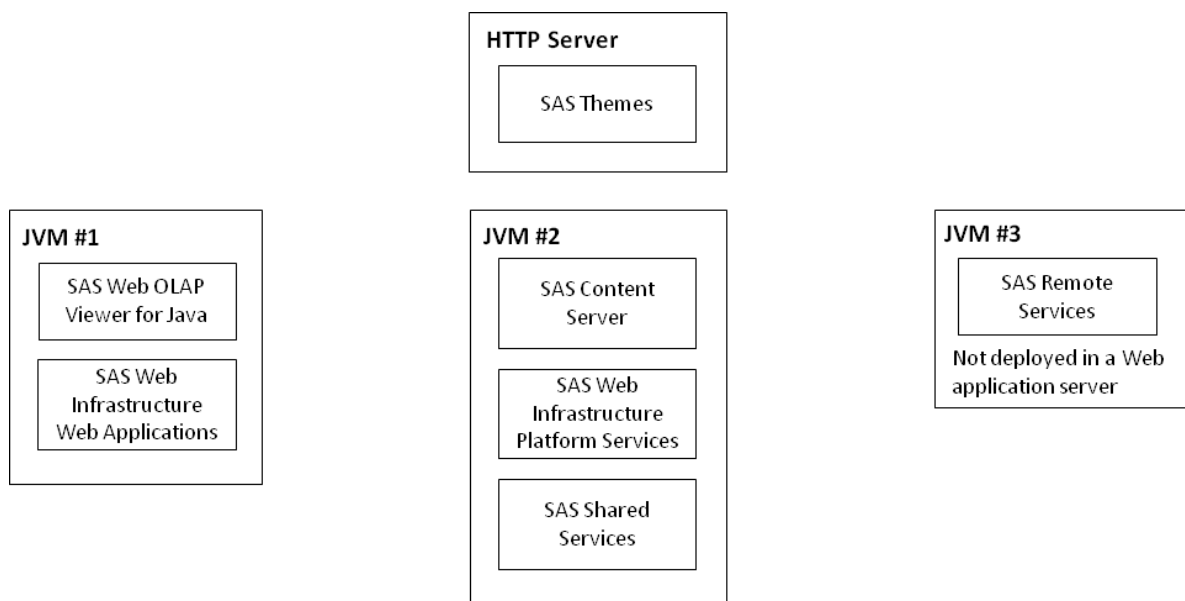
SAS Remote Services runs in its own JVM that is provided by a 32-bit Sun JDK.

`-Xms512m -Xmx512m -XX:PermSize=64m -XX:MaxPermSize=64m -Xss120k
-XX:-UseTLAB -XX:+UseConcMarkSweepGC -XX:NewSize=64m -XX:MaxNewSize=64m
-Dcom.sun.management.jmxremote -Djava.net.preferIPv4Stack=true`

Chapter 5 – Scaling Web OLAP Viewer for Java

This is pattern 4 and describes how to configure the SAS middle tier to focus on the SAS Web OLAP Viewer for Java application. This application is the focus for high performance requirements such as large number of users, fast response, and high throughput. SAS Web OLAP Viewer for Java and the SAS Web Infrastructure Platform Applications are deployed in one JVM. SAS Content Server, SAS Web Infrastructure Platform Services, and SAS Shared Services are deployed in another JVM. SAS Remote Services is deployed in its own JVM and SAS Themes is deployed to an HTTP server. The following figure shows the Web application topology that scales the SAS Web OLAP Viewer for Java application.

Figure 5 Application Topology for Scaling SAS Web OLAP Viewer for Java



Pattern 4: Tuning Parameters for HP and Sun JDKs

Note: The Common JVM Options on page 3 must be added along with the following application-specific JVM options.

JVM #1 SAS Web OLAP Viewer for Java and SAS Web Infrastructure Platform Applications

32 bit: `-server -Xms1280m -Xmx1280m -XX:PermSize=224m -XX:MaxPermSize=224m -Xss160k -XX:NewSize=256m -XX:MaxNewSize=256m -XX:+UseConcMarkSweepGC -XX:-UseTLAB -XX:+DisableExplicitGC`

64 bit: `-server -d64 -Xms2048m -Xmx2048m -XX:PermSize=256m -XX:MaxPermSize=256m -Xss256k -XX:NewSize=256m -XX:MaxNewSize=256m -XX:+UseConcMarkSweepGC -XX:+UseTLAB -XX:+DisableExplicitGC`

JVM #2 SAS Content Server, SAS Web Infrastructure Platform Services, and SAS Shared Services

32 bit: `-server -Xms1280m -Xmx1280m -XX:PermSize=224m -XX:MaxPermSize=224m
-Xss160k -XX:NewSize=256m -XX:MaxNewSize=256m -XX:+UseConcMarkSweepGC
-XX:-UseTLAB -XX:+DisableExplicitGC`

64 bit: `-server -d64 -Xms2048m -Xmx2048m -XX:PermSize=256m
-XX:MaxPermSize=256m -Xss256k -XX:NewSize=256m -XX:MaxNewSize=256m
-XX:+UseConcMarkSweepGC -XX:+UseTLAB -XX:+DisableExplicitGC`

JVM #3: SAS Remote Services

SAS Remote Services runs in its own JVM that is provided by a 32-bit Sun JDK.

`-Xms512m -Xmx512m -XX:PermSize=64m -XX:MaxPermSize=64m -Xss120k
-XX:-UseTLAB -XX:+UseConcMarkSweepGC -XX:NewSize=64m -XX:MaxNewSize=64m
-Dcom.sun.management.jmxremote -Djava.net.preferIPv4Stack=true`

Pattern 4: Tuning Parameters for an IBM JDK

Note: The Common JVM Options on page 3 must be added along with the following application-specific JVM options.

JVM #1 SAS Web OLAP Viewer for Java and SAS Web Infrastructure Platform Applications

32 bit: `-Xms1280m -Xmx1280m -Xss256k -Xmso256k -Xgcpolicy:gencon`

64 bit: `-Xms2048m -Xmx2048m -Xss256k -Xmso256k -Xgcpolicy:gencon`

JVM #2 SAS Content Server, SAS Web Infrastructure Platform Services, and SAS Shared Services

32 bit: `-Xms1280m -Xmx1280m -Xss256k -Xmso256k -Xgcpolicy:gencon`

64 bit: `-Xms2048m -Xmx2048m -Xss256k -Xmso256k -Xgcpolicy:gencon`

JVM #3: SAS Remote Services

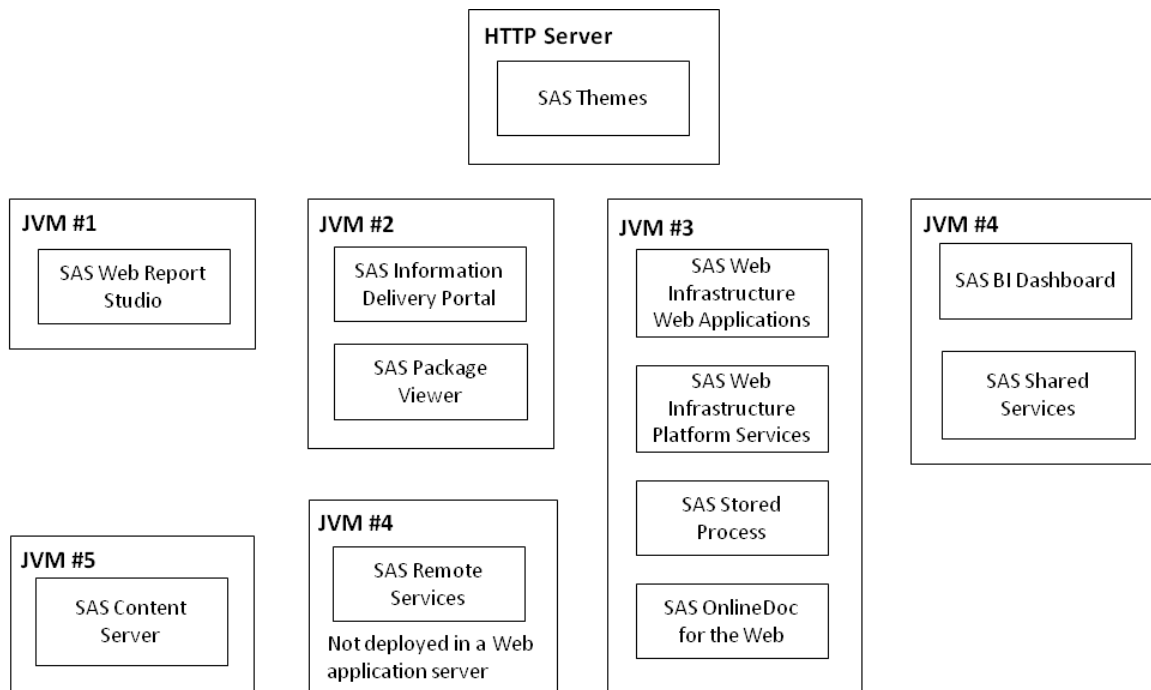
SAS Remote Services runs in its own JVM that is provided by a 32-bit Sun JDK.

`-Xms512m -Xmx512m -XX:PermSize=64m -XX:MaxPermSize=64m -Xss120k
-XX:-UseTLAB -XX:+UseConcMarkSweepGC -XX:NewSize=64m -XX:MaxNewSize=64m
-Dcom.sun.management.jmxremote -Djava.net.preferIPv4Stack=true`

Chapter 6 – Scaling SAS BI Dashboard

This is pattern 5 and describes how to configure the SAS middle tier to focus on the SAS BI Dashboard application. This application is the focus for high performance requirements such as a large number of users, fast response, and high throughput. This pattern is very similar to Pattern 1 that is used for scaling the entire SAS middle tier, except that SAS Web OLAP Viewer for Java is not included in this pattern and SAS Shared Services is deployed with SAS BI Dashboard. SAS Remote Services is deployed in its own JVM and SAS Themes is deployed to an HTTP server. The following figure shows the Web application topology that scales the SAS BI Dashboard application.

Figure 6 Application Topology for Scaling SAS BI Dashboard



Pattern 5: Tuning Parameters for an HP or Sun JDK

Note: The Common JVM Options on page 3 must be added along with the following application-specific JVM options.

JVM #1: SAS Web Report Studio

```

32-bit: -server -Xms1280m -Xmx1280m -XX:PermSize=128m -XX:MaxPermSize=128m
-Xss128k -XX:NewSize=128m -XX:MaxNewSize=128m -XX:+UseConcMarkSweepGC
-XX:+UseCMSInitiatingOccupancyOnly
-XX:CMSInitiatingOccupancyFraction=40 -XX:MaxTenuringThreshold=4
-XX:SurvivorRatio=4 -XX:-UseTLAB -XX:+DisableExplicitGC
  
```

64-bit: -server -d64 -Xms1280m -Xmx2048m -XX:PermSize=128m
 -XX:MaxPermSize=128m -Xss256k -XX:NewSize=256m -XX:MaxNewSize=256m
 -XX:+UseConcMarkSweepGC -XX:+UseCMSInitiatingOccupancyOnly
 -XX:CMSInitiatingOccupancyFraction=40 -XX:MaxTenuringThreshold=4
 -XX:SurvivorRatio=4 -XX:-UseTLAB -XX:+DisableExplicitGC

JVM #2: SAS Information Delivery Portal and SAS Package Viewer

32-bit: -server -Xms1280m -Xmx1280m -XX:PermSize=128m -XX:MaxPermSize=128m
 -Xss160k -XX:-UseTLAB -XX:+UseConcMarkSweepGC -XX:+DisableExplicitGC
 -XX:NewSize=128m -XX:MaxNewSize=128m -XX:+UseCMSInitiatingOccupancyOnly -
 XX:CMSInitiatingOccupancyFraction=30 -XX:-UseBiasedLocking
 -XX:MaxTenuringThreshold=4 -XX:SurvivorRatio=4

64-bit: -server -d64 -Xms1280m -Xmx2048m -XX:PermSize=128m
 -XX:MaxPermSize=128m -Xss256k -XX:-UseTLAB -XX:+UseConcMarkSweepGC
 -XX:+DisableExplicitGC -XX:NewSize=256m -XX:MaxNewSize=256m
 -XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=30
 -XX:-UseBiasedLocking -XX:MaxTenuringThreshold=4 -XX:SurvivorRatio=4

JVM #3: SAS Web Infrastructure Platform

This JVM hosts the SAS Web Infrastructure Platform Applications and SAS Web Infrastructure Platform Services.

32-bit: -server -Xms1280m -Xmx1280m -XX:PermSize=128m -XX:MaxPermSize=128m
 -Xss128k -XX:NewSize=128m -XX:MaxNewSize=128m -XX:+UseConcMarkSweepGC
 -XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=40
 -XX:MaxTenuringThreshold=4 -XX:SurvivorRatio=4 -XX:-UseTLAB
 -XX:+DisableExplicitGC

64-bit: -server -d64 -Xms1280m -Xmx2048m -XX:PermSize=128m
 -XX:MaxPermSize=128m -Xss256k -XX:NewSize=256m -XX:MaxNewSize=256m
 -XX:+UseConcMarkSweepGC -XX:+UseCMSInitiatingOccupancyOnly
 -XX:CMSInitiatingOccupancyFraction=40 -XX:MaxTenuringThreshold=4
 -XX:SurvivorRatio=4 -XX:-UseTLAB -XX:+DisableExplicitGC

JVM #4: SAS BI Dashboard and SAS Shared Services

32-bit: -server -Xms1024m -Xmx1024m -XX:PermSize=320m -XX:MaxPermSize=320m
 -XX:NewSize=128m -XX:MaxNewSize=128m -Xss128k -XX:+UseConcMarkSweepGC
 -XX:-UseTLAB -XX:+DisableExplicitGC -XX:+CMSCClassUnloadingEnabled
 -XX:+CMSPermGenSweepingEnabled
 -Dcom.sas.iom.orb.SelectionHub.clientThreadCount=3

64-bit: -server -d64 -Xms1024m -Xmx2048m -XX:PermSize=320m
 -XX:MaxPermSize=320m -XX:NewSize=256m -XX:MaxNewSize=256m -Xss256k
 -XX:+UseConcMarkSweepGC -XX:-UseTLAB -XX:+DisableExplicitGC
 -XX:+CMSCClassUnloadingEnabled -XX:+CMSPermGenSweepingEnabled
 -Dcom.sas.iom.orb.SelectionHub.clientThreadCount=3

JVM #5: SAS Content Server

32-bit: `-server -Xms1280m -Xmx1280m -XX:PermSize=96m -XX:MaxPermSize=96m
-Xss160k -XX:-UseTLAB -XX:+UseConcMarkSweepGC
-XX:+UseCMSInitiatingOccupancyOnly
-XX:CMSInitiatingOccupancyFraction=40 -XX:+DisableExplicitGC
-XX:NewSize=128m -XX:MaxNewSize=128m`

64-bit: `-server -d64 -Xms1280m -Xmx2048m -XX:PermSize=96m -XX:MaxPermSize=96m
-Xss256k -XX:-UseTLAB -XX:+UseConcMarkSweepGC
-XX:+UseCMSInitiatingOccupancyOnly
-XX:CMSInitiatingOccupancyFraction=40 -XX:+DisableExplicitGC
-XX:NewSize=256m -XX:MaxNewSize=256m`

JVM #6: SAS Remote Services

SAS Remote Services runs in its own JVM that is provided by a 32-bit Sun JDK.

`-Xms512m -Xmx512m -XX:PermSize=64m -XX:MaxPermSize=64m -Xss120k
-XX:-UseTLAB -XX:+UseConcMarkSweepGC -XX:NewSize=64m -XX:MaxNewSize=64m
-Dcom.sun.management.jmxremote -Djava.net.preferIPv4Stack=true`

Pattern 5: Tuning Parameters for an IBM JDK

Note: The Common JVM Options on page 3 must be added along with the following application-specific JVM options.

JVM #1: SAS Web Report Studio

32-bit: `-Xms1280m -Xmx1280m -Xss256k -Xmso256k -Xgcpolicy:gencon`

64-bit: `-Xms1280m -Xmx2048m -Xss256k -Xmso256k -Xgcpolicy:gencon`

JVM #2: SAS Information Delivery Portal and SAS Package Viewer

32-bit: `-Xms1280m -Xmx1280m -Xss256k -Xmso256k -Xgcpolicy:gencon`

64-bit: `-Xms1280m -Xmx2048m -Xss256k -Xmso256k -Xgcpolicy:gencon`

JVM #3: SAS Web Infrastructure Platform

This JVM hosts the SAS Web Infrastructure Platform Applications and SAS Web Infrastructure Platform Services.

32-bit: `-Xms1280m -Xmx2048m -Xss256k -Xmso256k -Xgcpolicy:gencon`

64-bit: `-Xms1280m -Xmx2048m -Xss256k -Xmso256k -Xgcpolicy:gencon`

JVM #4: SAS BI Dashboard and SAS Shared Services

32-bit: `-Xms1024m -Xmx1024m -Xss256k -Xmso256k -Xgcpolicy:gencon`

64-bit: `-Xms1024m -Xmx2048m -Xss256k -Xmso256k -Xgcpolicy:gencon`

JVM #5: SAS Content Server

32-bit: -Xms1280m -Xmx1280m -Xss256k -Xmso256k -Xgcpolicy:gencon

64-bit: -Xms1280m -Xmx2048m -Xss256k -Xmso256k -Xgcpolicy:gencon

JVM #6: SAS Remote Services

SAS Remote Services runs in its own JVM that is provided by a 32-bit Sun JDK.

-Xms512m -Xmx512m -XX:PermSize=64m -XX:MaxPermSize=64m -Xss120k
-XX:-UseTLAB -XX:+UseConcMarkSweepGC -XX:NewSize=64m -XX:MaxNewSize=64m
-Dcom.sun.management.jmxremote -Djava.net.preferIPv4Stack=true

Chapter 7 – Reconfiguring the SAS 9.2 Middle Tier

In the previous chapters of this document, several patterns are presented that describe how to achieve improved performance and scalability from the SAS 9.2 middle tier. As demonstrated by the variety of patterns, the SAS 9.2 Web applications have the flexibility to be distributed amongst several JVMs. The SAS Deployment Wizard is used to install and configure SAS software, and it supports creating a topology that uses two JVMs. This chapter describes how to distribute the SAS Web applications amongst three and four JVMs using a manual deployment process. The information in this chapter assumes that SAS 9.2 was already installed with the SAS Deployment Wizard.

Stage 1 – Plan the SAS 9.2 Middle Tier Application and JVM Distribution

As you plan SAS 9.2 environment based on the performance, functional and nonfunctional requirements, decide how many JVMs you plan to choose. The following list identifies the EAR files that are available with SAS 9.2:

- SAS BI Dashboard (sas.bidashboard4.2.ear and sas.bidashboardjsr1684.2.ear)
- SAS Package Viewer (sas.packageviewer4.2.ear)
- SAS Portal (sas.portal4.2.ear)
- SAS Shared Services (sas.shared9.2.ear)
- SAS Stored Process (sas.storedprocess9.2.ear)
- SAS Themes (sas.themes.ear)
- SAS Web OLAP Viewer (sas.webolapviewer4.2.ear)
- SAS Web Report Studio (sas.webreportstudio4.2.ear)
- SAS WIP Applications (sas.wip.apps9.2.ear)
- SAS WIP Content Server (sas.wip.scs9.2.ear)
- SAS WIP Services (sas.wip.services9.2.ear)

Note: The SAS Themes application is actually a WAR file with static content such as images and cascading style sheets. For the topologies described in this document, SAS recommends that you serve the static content from SAS Themes through an HTTP server. Information about performing this reconfiguration is provided in this chapter. In addition, SAS Remote Services is not identified in the following examples, but it must be present in the topology and it runs in its own JVM.

Example 1 – Three JVMs

The following table shows a recommended distribution of SAS 9.2 middle tier applications in a three JVM (Servers) topology.

Server number	SAS 9.2 Middle Tier Applications
Server1	SAS WIP Applications (sas.wip.apps9.2.ear) SAS WIP Content Server (sas.wip.scs9.2.ear) SAS WIP Services (sas.wip.services9.2.ear) SAS Shared Services (sas.shared9.2.ear)

Server2	SAS BI Dashboard (sas.bidashboard4.2.ear) SAS Portal (sas.portal4.2.ear) SAS Package Viewer (sas.packageviewer4.2.ear)
Server3	SAS Web Report Studio (sas.webreportstudio4.2.ear) SAS Web OLAP Viewer (sas.webolapviewer4.2.ear) SAS Stored Process (sas.storedprocess9.2.ear)

Example 2 – Four JVMs

The following table shows a recommended distribution of SAS 9.2 middle tier applications in a four JVM (Servers) topology.

Server number	SAS 9.2 Middle Tier Applications
Server1	SAS WIP Applications (sas.wip.apps9.2.ear) SAS WIP Content Server (sas.wip.scs9.2.ear) SAS WIP Services (sas.wip.services9.2.ear) SAS Shared Services (sas.shared9.2.ear)
Server2	SAS Portal (sas.portal4.2.ear) SAS Package Viewer (sas.packageviewer4.2.ear)
Server3	SAS Web Report Studio (sas.webreportstudio4.2.ear) SAS Stored Process (sas.storedprocess9.2.ear)
Server4	SAS BI Dashboard (sas.bidashboard4.2.ear and SAS Web OLAP Viewer (sas.webolapviewer4.2.ear)

Stage 2 – Prepare and Configure the Web Application Server Environment

This section assumes that you are familiar with configuring your Web application server (JBoss, WebLogic Server, or WebSphere Application Server). The following list identifies the high-level steps that you must perform on each Web application server:

1. Create the number servers that you decided in stage 1. For each of the servers, perform the following steps.
2. Create and Configure a Java Mail Session.
3. Create and Configure a JMS Server and resources.
4. Configure JDBC Driver - SAS Table Server.
5. Configure Data Source – Shared Service.
6. Deploy the respective SAS 9.2 middle tier applications (EAR files) to the server.
7. Load content to SAS Content Server.

8. Configure the JAAS Login Module for Application Logins – PFS and SCS.
9. Set the `-Dsas.auto.publish.port=JVM` option to the Web application server's own port number.

For detailed instructions about how to perform the steps in the previous list, see the `SAS-config-dir/Levn/Documents/Instructions.html` file. In addition, sample `Instructions.html` files are available on the [Third Party Software for SAS 9.2 Foundation](#) Web site.

Stage 3 – Deploy SAS Themes to an HTTP Server

In all the patterns provided in this document, SAS recommends deploying the SAS Themes content to an HTTP server. By serving the SAS Themes content from an HTTP server, you shift the processing load of serving static HTML files from the Web application server to the HTTP server. The following instructions describe how to perform this deployment on Apache HTTP Server. Performing this reconfiguration on other HTTP servers is similar. To configure Apache HTTP Server to serve the static HTML files for SAS Themes, perform the following steps:

1. In a temporary directory, extract the contents of `SAS-config-dir/Levn/Web/Staging/sas.themes.ear`:


```
jar xf /opt/SAS/Config/Lev1/Web/Staging/sas.themes.ear
```

The WAR file `sas.theme.default.war` is extracted.
2. Change directory to `APACHE_HOME/htdocs` and make a new directory named `SASTheme_default`.
3. Extract the `sas.theme.default.war` file into `APACHE_HOME/htdocs/SASTheme_default`.

Stage 4 – Change the Connections for the SAS Web Applications

After SAS Themes is deployed to the HTTP server and the SAS Web applications are distributed to different servers, information about to access the applications such as host and port must be updated in SAS metadata. Change the connection information to a URL that includes the Web application server host name and port number for the Web application server that is hosting the SAS Web application. Information about the SAS Themes application must be changed to the HTTP server host name and port.

To change the connection access point, follow these steps in SAS Management Console:

1. Select **Application Management > Configuration Manager**.
2. Right-click on the SAS Web application you want to reconfigure, and select **Properties**.
3. Click the **Connection** tab, set **Host Name** and **Port Number** to the host name and port number of the Web application server that is hosting the SAS Web application (use the HTTP server host name and port for `SASTheme_default`), and then click **OK**.

Stage 5 – Change the Connection for SAS Content Server

If the SAS Content Server application (`sas.wip.scs9.2.ear`) is not deployed on the original Web application server and port, then you must reconfigure SAS metadata with the connection information. This change is similar to the change required for each of the SAS Web applications. SAS

metadata must be changed to identify the host name and port of the Web application server that is hosting the SAS Content Server application. To reconfigure the host name and port of the SAS Content Server in SAS metadata, perform the following steps in SAS Management Console:

1. Select **Server Manager > SAS Content Server**.
2. Right-click the **Connection: SAS Content Server** icon in the right panel and select **Properties**.
3. Click the **Options** tab and set the **Host name** and **Port number** fields to the host name and port number of the Web application server that is hosting the `sas.wip.scs9.2.ear` EAR file or the HTTP Server that is performing a reverse proxy for the SAS Content Server.
4. Click **OK**.

Stage 6 – Change the WebDAV Repository URL

Just as in Stage 5, if the SAS Content Server application (`sas.wip.scs9.2.ear`) is not deployed on the original Web application server and port, then you must reconfigure SAS metadata with the connection information for the WebDAV repository. There are five applications that use SAS metadata to identify the connection information for the SAS Content Server. These applications are identified in the following list:

- Remote Services
- SASPackageViewer4.2 Local Services
- SASPortal4.2 Local Services
- SASStoredProcess9.2 Local Services
- SASWebReportStudio4.2 Local Services

To reconfigure the WebDAV URL for the applications, perform the following steps in SAS Management Console:

1. Select **Environment Management > Foundation Services Manager**.
2. Select the application and then select **Core > Information Service**.
3. Right-click **Information Service** and select **Properties**.
4. On the **Information Service Properties** dialog box, click the **Service Configuration** tab and then click **Configuration**.
5. On the **Information Service Configuration** dialog box, click the **Repositories** tab.
6. Select **WebDAV** and then click **Edit**.
7. Change the **Host** and **Port** values to the host name and port of the Web application server that is hosting the `sas.wip.scs9.2.ear` EAR file.
8. Click **OK** to close the **Information Service Configuration** dialog box.
9. Click **OK** to close the **Information Service Properties** dialog box.

Chapter 8 – SAS Application Tuning Parameters

The following table provides JVM options that are used to configure the SAS Web applications. A description is provided for each option to help you decide whether modifying a value is appropriate for your deployment.

JVM Option	Recommended Value	Description
-Dhttp.proxyHost	proxy server host name	The HTTP proxy options are used by the SAS Information Delivery Portal to access DTDs over the Internet. These options are used only when a proxy server for Internet connections exists. If you use a proxy server, make sure that the local domain, within the network boundary, is included in the exclude list. For example: -Dhttp.proxyHost=<host name> -Dhttp.proxyPort=80 -Dhttp.nonProxyHosts=
-Dhttp.proxyPort	proxy server port	
-Dhttp.nonProxyHosts	non-proxy host (exclude list)	
-Dsas.svcs.http.max.connections	50 (default value is 2)	The recommended value is based on an environment with a large number of concurrent users.
-Djgroups.bind_addr	host name or IP address	Use this option on a system that has multiple network interfaces (physical or virtual). When this option is used, also use -Dresolve.dns=false.
-Dmulticast_udp_ip_ttl	30 (default value is 64)	This option sets the time to live on UDP packets that are used for multicast calls.
-Dlogoff.corePoolSize	50	Use this option in an environment where

		concurrent user connections are significantly high (for example, over 500 connections).
<code>-Dlogoff.maxPoolSize</code>	50	Use this option with the <code>-Dlogoff.corePoolSize</code> parameter if the concurrent user connections and the frequency of user logoffs is significantly high (for example, over 500 connections and logoffs).
<code>-Dcom.sas.iom.orb.SelectionHub. clientThreadCount</code>	3	This option controls the number of threads waiting for packets from IOM servers. In an environment with significant concurrent user connections (for example, over 500 connections), set the value to 3.

Chapter 9 – Web Application Server Tuning Parameters

The following table provides a set of JVM options that are identified by the Web application server vendor, and that are relevant to the SAS 9.2 middle tier.

Web Application Server	Parameter	Recommended Value	Description
JBoss	Thread Pool	25 min and 75 max	Controls the number of threads in the Web container thread pool.
JBoss	Disable deployment scanner		Disable the deployment scanner that is used for hot deployment and undeployment.
WebLogic	<code>-Dweblogic.ChunkSize</code>	a multiple of the network maximum transfer unit (MTU), minus the TCP header size	Set this value as the number of bytes and provide it with the JVM options for the server start up.
WebLogic	Accept Backlog		Increase the default value by 25% when you see “connection refused or dropped” message at the client, but no errors on the server.
WebSphere	Thread Pool	25 min and 75 max	Controls the number of threads in the Web container thread pool.
WebSphere	<code>HttpSessionIdReuse</code>	true	Add this custom property to the JVM options for the application server.
WebSphere	<code>prependSlashToResource</code>	true	Add this custom property to the application server.
WebSphere	<code>com.ibm.ws.webcontainer.channelwritetype</code>	true	Add this custom property to the Web container.
WebSphere	JSP enable class reloading	disable	Disable this property for each of the SAS Web applications.

Chapter 10 – Java Virtual Machine Tuning Parameters

SAS 9.2 supports both 32-bit and 64-bit platforms. Tuning the JVM heap, PermGen space size, garbage collection, and stack size all differ between 32-bit and 64-bit platforms. On 32-bit Windows platforms there is an address space restriction. The total heap, Java heap plus the native heap, cannot exceed a 2GB limit in a Windows environment. However, you can increase the 2GB limit to a 3GB limit for 32-bit Windows platforms. For information about how to increase the limit, see the following URL:

<http://technet.microsoft.com/en-us/library/bb124810.aspx>

SAS recommends increasing the limit to 3GB for any aggressive performance and scalability requirements. On 32-bit platforms, avoid exceeding 1280MB as the maximum JVM heap size. The goal of the JVM tuning is to minimize the garbage collection cycles while maximizing the performance and number of clients that the SAS Web applications can support.

The following table summarizes general guidelines for JVM tuning in the SAS 9.2 middle tier when the applications are distributed to multiple JVMs. If you plan to deploy the SAS 9.2 middle tier to one or two JVMs, then use the default JVM options that were provided by the SAS Deployment Wizard during the installation of SAS software.

JVM Option	32-Bit Platform Value	64-Bit Platform Value
Heap size, min and max	-Xms1280m -Xmx1280m	-Xms2048m -Xmx2048m
Stack size	-Xss128k	-Xss256k
Permanent generation size (Sun and HP JDKs)	-XX:PermSize=128m	-XX:PermSize=256m
Young generation size (Sun and HP JDKs)	-XX:NewSize=128m -XX:MaxNewSize=128m	-XX:NewSize=256m -XX:MaxNewSize=256m
Garbage collection (Sun and HP JDKs) for 32-bit and 64-bit	-XX:+UseConcMarkSweepGC -XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=40 -XX:MaxTenuringThreshold=4 -XX:SurvivorRatio=4	
Garbage collection (IBM JDK) for 32-bit and 64-bit	-Xgcpolicy:gencon	
Distributed garbage collection intervals for 32-bit and 64-bit	-Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000	

The following table describes the JVM options that are used in the SAS 9.2 middle tier.

JVM Option	Description
-Xms	Minimum heap
-Xmx	Maximum heap

-Xss	Stack size
-XX:PermSize	For Sun and HP JDKs this option sets the initial permanent generation storage size.
-XX:MaxPermSize	For Sun and HP JDKs this option sets the maximum permanent storage size.
-XX:NewSize -XX:MaxNewSize	For Sun and HP JDKs this option sets the young generation sizes.
-XX:UseConcMarkSweepGC -XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction	For Sun and HP JDKs these options modify garbage collection to use the concurrent low pause collector.
-XX:MaxTenuringThreshold	For Sun and HP JDKs, objects that survive a young generation collection have a copying cost. This value is used to set the copying costs.

Setting JVM Options for JBoss

This section describes how to set JVM options for JBoss.

When JBoss is deployed on UNIX or Linux, the JVM options are specified in the `JAVA_OPTS` variable that is found in `JBOSS_HOME/bin/SASServer1.sh`.

When JBoss is deployed on Windows as a service, the JVM options are specified in the `JBOSS_HOME\server\SASServer1\wrapper.conf` file. Create a new entry for any new JVM options. The following example shows a few of the lines from the file:

```
# Java Additional Parameters
wrapper.java.additional.1=-Xms768m
wrapper.java.additional.2=-Xmx768m
wrapper.java.additional.3=-XX:PermSize=512m
wrapper.java.additional.4=-XX:MaxPermSize=512m
wrapper.java.additional.5=-Xss160k
wrapper.java.additional.6=-XX:NewSize=128m
```

When JBoss is deployed on Windows, but run from scripts rather than as a service, the JVM options are specified in the `JAVA_OPTS` variable that is found in the `JBOSS_HOME\bin\SASServer1.bat` file.

Setting JVM Options for WebLogic Server

For BEA WebLogic Server 9.2, you can set JVM options by editing a start script or by using the Administration Console. Use whichever approach you have used to set other server options. The paragraphs below explain how to use each approach.

Note: If you are not running the Node Manager, then you must specify JVM options in a start script.

To set the JVM options in a start script, follow these steps:

1. Change directories to your WebLogic Server domain directory that is configured for SAS 9.2. For a SAS Deployment Wizard configured domain, this directory is `SAS-config-dir/Levn/Web/SASDomain/bin`.

2. Open the server start script in an editor. The script is named `startManagedWebLogic` with a `.sh` or `.cmd` suffix.
3. Uncomment the line reserved for setting JVM options and add them. For a SAS Deployment Wizard configured domain, use the `JAVA_OPTIONS` variable.

If the same script is used to start more than one managed server and you want each managed server to start with different JVM options, then you need to perform additional editing so that the preferred JVM options are applied to each managed server.

4. Save your changes, and close the file.
5. Restart the server so that the new settings are used.

To set the JVM options using the Administration Console, follow these steps:

1. In the **Domain Structure** panel, expand the **Servers** node.
2. Click the name of the server that you want to configure.
3. In the right pane, click **Server Start**.
4. Select **Lock & Edit**.
5. In the **Arguments** text field, provide the JVM options. After inserting your options, click **Save** and then click **Activate Changes**.
6. Restart the server so that the new settings are used.

Setting JVM Options for WebSphere Application Server

Use the WebSphere Application Server administration console to set JVM options. The following procedure describes how to set the parameters for WebSphere Application Server 6.1. For more information, see the IBM WebSphere Application Server documentation.

1. On the left panel, expand the **Servers** node.
2. Click **Application Servers** and then click the name of the server that you want to configure.
3. Select **Process Definition > Java Virtual Machine**.
4. Provide the JVM options in the **Generic JVM Arguments** text field.
5. Click **Apply** and then **Save** the changes to the master configuration.
6. Restart the server so that the new settings are used.

Note: Some parameters can be specified in other boxes on the Java Virtual Machine page. Avoid placing the same settings in different text fields on this page; this can have unpredictable results.

Chapter 11 – Operating System Tuning Parameters

There are number of configuration changes and variables that you can set to tune the SAS 9.2 middle tier for your performance and scalability needs. The following sections show how to configure the settings that are relevant to the SAS 9.2 middle tier and the Web application servers that SAS 9.2 supports.

Note: For UNIX and Linux operating environments, you must also confirm (or configure) that the ulimit resource limit for open file descriptors is at least 10240. Information about this resource limit is provided at the [Pre-Installation Steps for JBoss, WebLogic, and WebSphere Application Servers with SAS 9.2](#) Web site. This step is especially important for deployments that use WebLogic Server.

Solaris SPARC and Solaris x86-64

Use the following commands to set the SAS recommended parameters:

```
ndd -set /dev/tcp tcp_keepalive_interval 300000
nnd -set /dev/tcp tcp_time_wait_interval 60000
nnd -set /dev/tcp tcp_conn_req_max_q 256
nnd -set /dev/tcp tcp_conn_req_max_q0 2500
nnd -set /dev/tcp tcp_max_buf 4000000
```

Linux

Use the following commands to set the SAS recommended parameters:

```
echo 30 > /proc/sys/net/ipv4/tcp_fin_timeout
echo 3000 > /proc/sys/net/core/netdev_max_backlog
echo 3000 > /proc/sys/net/core/somaxconn
echo 15 > /proc/sys/net/ipv4/tcp_keepalive_intvl
echo 5 > /proc/sys/net/ipv4/tcp_keepalive_probes
```

AIX

Use the following commands to set the SAS recommended parameters:

```
/usr/sbin/no -o tcp_timewait =1
/usr/sbin/no -o tcp_keepidle=600
/usr/sbin/no -o tcp_keepintvl=10
/usr/sbin/no -o tcp_keepinit=40
```

HP-UX

Use the following commands to set the SAS recommended parameters:

```
nnd -set /dev/tcp tcp_conn_request_max 8192
nnd -set /dev/tcp tcp_keepalive_interval 7200000
nnd -set /dev/tcp tcp_keepalives_kill 1
```

In addition, SAS recommends tuning the kernel parameters by using the SAM utility or the kctune utility.

Kernel Parameter	Recommended Value
swchunk	8192
shmseg	512
maxdsiz	3221225472
maxdsiz_64bit	64424509440
maxfiles_lim	10000 (Change this one before maxfiles.)
maxfiles	8192
semume	512
semmsl	3072
msgssz	512 (Change this one before msgmax)
nkthread	10000
max_thread_proc	4096
nproc	8192 (Change this one before maxuprc)
maxuprc	4096
nflocks	11585
ninode	8110
msgmap	13109
msgseg	32767 (Change this one before msgmax)
msgmnb	65535 (0x10000) (Change this one before msgmax)
msgmnb	131070 (when running multiple JVMs on the same system)
msgmax	65535 (0x10000)
msgmax	131070 (when running multiple JVMs on the same system)
msgmni	4634
semmns	11586
semmni	8192
semmnu	8180
shmmax	1.85514E+11
shmmni	8192
STRMSGSZ	65535
dbc_max_pc	10

nstrpty	60
cmc_plat_poll	15
msgtql	13107

Windows

Microsoft recommends performing a system backup before editing the registry. Use **regedit** to set the SAS recommended parameters:

1. Access HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TCPIP\Parameters. Add the following **DWORD Values**:

Name	Value
TcpTimedWaitDelay	0x1e (30 seconds)
MaxUserPort	greater than 32768

2. Set values for the number of available connections. Request a minimum of 20 and a maximum of 1000 available connections. Increase the number of available connections by 10 each time that there are fewer than the minimum numbers of available connections. Access HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AFD\Parameters. Add the following **DWORD Values**:

Name	Value
EnableDynamicBacklog	1 decimal
MinimumDynamicBacklog	20 decimal
MaximumDynamicBacklog	1000 decimal
DynamicBacklogGrowthDelta	10 decimal

3. For Windows 2000 hosts, access HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\. Add a **DWORD Value** with a name of TcpDelAckTicks and a value of 0.
4. For Windows 2003 hosts, access HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\. Add a **DWORD Value** with a name of TcpAckFrequency and a value of 1.
5. Restart Windows.



THE
POWER
TO KNOW.

support.sas.com

SAS is the world leader in providing software and services that enable customers to transform data from all areas of their business into intelligence. SAS solutions help organizations make better, more informed decisions and maximize customer, supplier, and organizational relationships. For more than 30 years, SAS has been giving customers around the world The Power to Know®. Visit us at **www.sas.com**.