SAS 9.2 Web Applications: Tuning for Performance and Scalability
# 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 and SAS Shared Services ......................................................... 7  
JVM #5: SAS Web Infrastructure Platform ......................................................................... 7  
JVM #6: SAS Content Server ............................................................................................... 7  
JVM #7: SAS Remote Services ............................................................................................. 8  

**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 and SAS Shared Services ....................................................... 8  
JVM #5: SAS Web Infrastructure Platform ....................................................................... 8  
JVM #6: SAS Content Server .......................................................................................... 9  
JVM #7: SAS Remote Services .......................................................................................... 9  

Chapter 3 — Scaling SAS Information Delivery Portal ....................................................... 10  
**Pattern 2: Tuning Parameters for HP and Sun JDKs** .................................................. 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  

**Pattern 2: Tuning Parameters for an IBM JDK** ......................................................... 11  
JVM #1 SAS Information Delivery Portal and SAS Web Infrastructure Platform Applications .................................................................................................................. 11  
JVM #2 SAS Web Infrastructure Platform Services and SAS Package Viewer ............... 11  
JVM #3 SAS Content Server ........................................................................................... 12  
JVM #4: SAS Remote Services ......................................................................................... 12  

Chapter 4 — Scaling SAS Web Report Studio .................................................................... 13  
**Pattern 3: Tuning Parameters for Sun and HP JDKs** .................................................. 13  
JVM #1 SAS Web Report Studio ........................................................................................ 13  
JVM #2 SAS Web Infrastructure Platform ...................................................................... 14  
JVM #3 SAS Content Server ........................................................................................... 14  
JVM #4: SAS Remote Services ......................................................................................... 14  

**Pattern 3: Tuning Parameters for an IBM JDK** ......................................................... 14  
JVM #1 SAS Web Report Studio ........................................................................................ 14  
JVM #2 SAS Content Server and SAS Web Infrastructure Platform .............................. 14  
JVM #3: SAS Remote Services ......................................................................................... 15  

Chapter 5 – Scaling Web OLAP Viewer for Java ............................................................. 16  
**Pattern 4: Tuning Parameters for HP and Sun JDKs** .................................................. 16
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.
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.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Scaling the entire SAS middle tier</td>
<td>Moderate to Aggressive</td>
<td>Moderate to Aggressive</td>
<td>Moderate to Aggressive</td>
<td>Moderate to Aggressive</td>
</tr>
<tr>
<td>Scaling SAS Information Delivery Portal</td>
<td>Aggressive</td>
<td>None or Low</td>
<td>None or Low</td>
<td>None or Low</td>
</tr>
<tr>
<td>Scaling SAS Web Report Studio</td>
<td>None, Low, or Moderate</td>
<td>Aggressive</td>
<td>None, Low, or Moderate</td>
<td>None, Low, or Moderate</td>
</tr>
<tr>
<td>Scaling SAS BI Dashboard</td>
<td>None, Low, or Moderate</td>
<td>None, Low, or Moderate</td>
<td>Aggressive</td>
<td>None, Low, or Moderate</td>
</tr>
<tr>
<td>Scaling SAS Web OLAP Viewer for Java</td>
<td>None, Low, or Moderate</td>
<td>None, Low, or Moderate</td>
<td>None, Low, or Moderate</td>
<td>Aggressive</td>
</tr>
</tbody>
</table>
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=20
**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 and IBM JDK respective platforms, see the Third Party Software for SAS 9.2 Foundation Web site. 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:

**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`

**IBM JDK**

- `-Xms512m -Xmx512m -Xss120k -Xms120k -Xgcpolicy:gencon`
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 -Xms1152m -Xmx1280m -XX:PermSize=256m
-XX:MaxPermSize=256m -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=512m
-XX:MaxPermSize=512m -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=512m
-XX:MaxPermSize=512m -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 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:+CMSClassUnloadingEnabled
-XX:+CMSPermGenSweepingEnabled
-Dcom.sas.iom.orb.SelectionHub.clientThreadCount=3

**64-bit:**
- server -d64 -Xms1024m -Xmx2048m -XX:PermSize=512m
-XX:MaxPermSize=512m -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 #5: SAS Web Infrastructure Platform

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

**32-bit:**
- server -Xms1280m -Xmx1280m -XX:PermSize=256m -XX:MaxPermSize=256m
-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=512m
-XX:MaxPermSize=512m -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 and IBM JDK in respective platforms. The tuning options for

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

IBM JDK
-Xms512m -Xmx512m -Xss120k -Xms120k -Xgcpolicy:gencon

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 -Xms256k -Xgcpolicy:gencon
64-bit: -Xms1280m -Xmx2048m -Xss256k -Xms256k -Xgcpolicy:gencon

JVM #2: SAS Information Delivery Portal

32-bit: -Xms1280m -Xmx1280m -Xss256k -Xms256k -Xgcpolicy:gencon
64-bit: -Xms1280m -Xmx2048m -Xss256k -Xms256k -Xgcpolicy:gencon

JVM #3: SAS Web OLAP Viewer for Java

32-bit: -Xms1280m -Xmx1280m -Xss256k -Xms256k -Xgcpolicy:gencon
64-bit: -Xms1280m -Xmx2048m -Xss256k -Xms256k -Xgcpolicy:gencon

JVM #4: SAS BI Dashboard and SAS Shared Services

32-bit: -Xms1024m -Xmx1024m -Xss256k -Xms256k -Xgcpolicy:gencon
64-bit: -Xms1024m -Xmx2048m -Xss256k -Xms256k -Xgcpolicy:gencon

JVM #5: SAS Web Infrastructure Platform

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

32-bit: -Xms1280m -Xmx1280m -Xss256k -Xms256k -Xgcpolicy:gencon
64-bit: -Xms1280m -Xmx2048m -Xss256k -Xms256k -Xgcpolicy:gencon
**JVM #6: SAS Content Server**

32-bit: -Xms1280m -Xmx1280m -Xss256k -Xms256k -Xgcpolicy:gencon

64-bit: -Xms1280m -Xmx2048m -Xss256k -Xms256k -Xgcpolicy:gencon

**JVM #7: SAS Remote Services**

SAS Remote Services runs in its own JVM that is provided by a 32-bit Sun JDK and IBM JDK in respective platforms. The tuning options for

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

IBM JDK
- -Xms512m -Xmx512m -Xss120k -Xms120k -Xgcpolicy:gencon
### 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**

![Application Topology](image)

**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**

-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:+CMSGenSweepingEnabled`
JVM #2 SAS Web Infrastructure Platform Services and SAS Package Viewer

32-bit:  
-server -Xms1152m -Xmx1152m -XX:PermSize=256m -XX:MaxPermSize=256m  
-Xss160k -XX:+UseTLAB -XX:+UseConcMarkSweepGC  
-XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=40  
-XX:+DisableExplicitGC -XX:NewSize=128m -XX:MaxNewSize=128m

64-bit:  
-server -d64 -Xms2048m -Xmx2048m -XX:PermSize=512m  
-XX:MaxPermSize=512m  
-Xss256k -XX:-UseTLAB -XX:+UseConcMarkSweepGC  
-XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=40  
-XX:+DisableExplicitGC -XX:NewSize=256m -XX:MaxNewSize=256m

JVM #3 SAS Content Server

32-bit:  
-server -Xms1280m -Xmx1280m -XX:PermSize=128m -XX:MaxPermSize=128m  
-Xss128k -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:+UseConcMarkSweepGC  
-XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=40  
-XX:+DisabledExplicitGC  
-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 -Xms256k -Xgcpolicy:gencon

64-bit: -Xms1024m -Xmx4096m -Xss256k -Xms256k -Xgcpolicy:gencon

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

32-bit: -Xms1280m -Xmx1280m -Xss256k -Xms256k -Xgcpolicy:gencon

64-bit: -Xms1024m -Xmx3072m -Xss256k -Xms256k -Xgcpolicy:gencon
**JVM #3 SAS Content Server**

**32-bit:** `-Xms1280m -Xmx1280m -Xss256k -Xms256k -Xgcpolicy:gencon`

**64-bit:** `-Xms1280m -Xmx1280m -Xss256k -Xms256k -Xgcpolicy:gencon`

**JVM #4: SAS Remote Services**

SAS Remote Services runs in its own JVM that is provided by a 32-bit Sun JDK and IBM JDK in respective platforms. The tuning options for

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`

IBM JDK

- `-Xms512m -Xmx512m -Xss120k -Xms120k -Xgcpolicy:gencon`
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


64-bit: -server -d64 -Xms2048m -Xmx2048m -XX:PermSize=256m -XX:MaxPermSize=256m -XX:MaxNewSize=256m -XX:MaxNewSize=256m -XX:+UseConcMarkSweepGC -XX:+UseTLAB -XX:+DisableExplicitGC
JVM #2 SAS Web Infrastructure Platform

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


64-bit: -server -d64 -Xms2048m -Xmx2048m -XX:PermSize=320m -XX:MaxPermSize=320m -Xss256k -XX:NewSize=256m -XX:MaxNewSize=256m -XX:+UseConcMarkSweepGC -XX:+UseTLAB -XX:+DisableExplicitGC

JVM #3 SAS Content Server


64-bit: -server -d64 -Xms1024m -Xmx2048m -XX:PermSize=512m -XX:MaxPermSize=512m -Xss256k -XX:NewSize=256m -XX:MaxNewSize=256m -XX:+UseConcMarkSweepGC -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.

-XX:+CMSIncrementalMode -XX:+CMSClassUnloadingEnabled

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 -Xms256k -Xgcpolicy:gencon

64-bit: -Xms2048m -Xmx2048m -Xss256k -Xms256k -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 -Xms256k -Xgcpolicy:gencon

64-bit: -Xms1024m -Xmx3072m -Xss256k -Xms256k -Xgcpolicy:gencon
JVM #3: SAS Remote Services

SAS Remote Services runs in its own JVM that is provided by a 32-bit Sun JDK and IBM JDK in respective platforms. The tuning options for

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

IBM JDK
- Xms512m -Xmx512m -Xss120k -Xms120k -Xgcpolicy:gencon
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
-XX:s160k -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**

64-bit: `server -d64 -Xms2048m -Xmx2048m -XX:PermSize=512m -XX:MaxPermSize=512m -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:+UseConcMarkSweepGC -XX:+UseTLAB -XX:+DisableExplicitGC -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: `server -Xms1280m -Xmx1280m -Xss256k -Xms256k -Xgcpolicy:gencon`  
64-bit: `server -Xms2048m -Xmx2048m -Xss256k -Xms256k -Xgcpolicy:gencon`  

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

32-bit: `server -Xms1280m -Xmx1280m -Xss256k -Xms256k -Xgcpolicy:gencon`  
64-bit: `server -Xms2048m -Xmx2048m -Xss256k -Xms256k -Xgcpolicy:gencon`  

**JVM #3: SAS Remote Services**

SAS Remote Services runs in its own JVM that is provided by a 32-bit Sun JDK and IBM JDK in respective platforms. The tuning options for

**Sun JDK**

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

**IBM JDK**

-Xms512m -Xmx512m -Xss120k -Xms120k -Xgcpolicy:gencon
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. 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

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 -Xms1152m -Xmx1152m -XX:PermSize=256m -XX:MaxPermSize=256m
-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 -Xms1024m -Xmx2048m -XX:PermSize=512m
-XX:MaxPermSize=512m -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 -Xms1152m -Xmx1152m -XX:PermSize=256m -XX:MaxPermSize=256m
-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=512m
-XX:MaxPermSize=512m -Xss256k -XX:-UseTLAB -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:+CMSClassUnloadingEnabled
-XX:+CMSPermGenSweepingEnabled
-Dcom.sas.iom.orb.SelectionHub.clientThreadCount=3

64-bit: -server -d64 -Xms1024m -Xmx2048m -XX:PermSize=512m
-XX:MaxPermSize=512m -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 #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 and IBM JDK in respective platforms. The tuning options for

**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

**IBM JDK**
- Xms512m -Xmx512m -Xss120k -Xms120k -Xgcpolicy:gencon

---

**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 -Xms256k -Xgcpolicy:gencon

**64-bit:**
- Xms1280m -Xmx2048m -Xss256k -Xms256k -Xgcpolicy:gencon

---

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

**32-bit:**
- Xms1280m -Xmx1280m -Xss256k -Xms256k -Xgcpolicy:gencon

**64-bit:**
- Xms1280m -Xmx2048m -Xss256k -Xms256k -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 -Xms256k -Xgcpolicy:gencon

**64-bit:**
- Xms1280m -Xmx2048m -Xss256k -Xms256k -Xgcpolicy:gencon
JVM #4: SAS BI Dashboard and SAS Shared Services
32-bit: -Xms1024m -Xmx1024m -Xss256k -Xms256k -Xgcpolicy:gencon
64-bit: -Xms1024m -Xmx2048m -Xss256k -Xms256k -Xgcpolicy:gencon

JVM #5: SAS Content Server
32-bit: -Xms1280m -Xmx1280m -Xss256k -Xms256k -Xgcpolicy:gencon
64-bit: -Xms1280m -Xmx2048m -Xss256k -Xms256k -Xgcpolicy:gencon

JVM #6: SAS Remote Services
SAS Remote Services runs in its own JVM that is provided by a 32-bit Sun JDK and IBM JDK in respective platforms. The tuning options for

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

IBM JDK
-Xms512m -Xmx512m -Xss120k -Xms120k -Xgcpolicy:gencon
Chapter 7 – SAS Workflow Tuning

SAS Workflow is a suite of services that provides a platform for business process augmentation. Use SAS Workflow to model, automate, integrate, monitor, and streamline business processes. SAS Workflow is used primarily by SAS solutions such as SAS Enterprise Case Management. When SAS Workflow is used directly through custom application development or through integration with standalone or enterprise level business process, it must be tuned based on the usage volume. This section provides tuning recommendations that apply to a high-usage SAS Workflow environment. SAS Workflow is packaged with the SAS Shared Services Web application as Web archive (WAR). For high-usage environments, SAS recommends deploying SAS Workflow in a separate JVM and to use the following JVM tuning options.

Sun JDK

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

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

IBM JDK

32-bit: 
- Xms1280m -Xmx1280m -Xss256k -Xms256k -Xgcpolicy:gencon -Xmn294m
-XX:DisableExplicitGC

64-bit: 
- Xms2048m -Xmx2048m -Xss256k -Xms256k -Xgcpolicy:gencon -Xmn384m
-XX:DisableExplicitGC

SAS Workflow Cache Tuning

SAS Workflow uses a cache. The cache is populated based on the SAS Workflow instance creation and loading. By default, the maximum number of cache elements is 100000. This default cache size can process up to 10000 workflow instances. If the deployment requires more than 10000 workflow instances, then the cache size must be increased. The cache size is set in the ehcache.xml configuration file. This file is stored in the sas.workflow.engine.server.jar under WEB-INF/lib directory of SAS Workflow WAR. To change the cache size, you must extract the contents of the sas.workflow.war file, edit the ehcache.xml file, and then re-create the sas.workflow.war file.

If you want to reduce the memory requirement and possibly the need to have a different JVM for Workflow, you can reduce the default cache size by editing the ehcache.xml file and re-creating the sas.workflow.war file.

When using a 32-bit platform the address space is limited to 2GB. For a high end workload on a 32-bit platform, reduce the cache size and utilize the clustering (either vertical or horizontal) based on the performance requirements.
Chapter 8 – 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.

<table>
<thead>
<tr>
<th>Server number</th>
<th>SAS 9.2 Middle Tier Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server1</td>
<td>SAS WIP Applications (sas.wip.apps9.2.ear)</td>
</tr>
<tr>
<td></td>
<td>SAS WIP Content Server (sas.wip.scs9.2.ear)</td>
</tr>
<tr>
<td></td>
<td>SAS WIP Services (sas.wip.services9.2.ear)</td>
</tr>
<tr>
<td>Server2</td>
<td>SAS BI Dashboard (sas.bidashboard4.2.ear)</td>
</tr>
<tr>
<td></td>
<td>SAS Shared Services (sas.shared9.2.ear)</td>
</tr>
<tr>
<td></td>
<td>SAS Portal (sas.portal4.2.ear)</td>
</tr>
<tr>
<td></td>
<td>SAS Package Viewer (sas.packageviewer4.2.ear)</td>
</tr>
<tr>
<td>Server3</td>
<td>SAS Web Report Studio (sas.webreportstudio4.2.ear)</td>
</tr>
<tr>
<td></td>
<td>SAS Web OLAP Viewer (sas.webolapviewer4.2.ear)</td>
</tr>
<tr>
<td></td>
<td>SAS Stored Process (sas.storedprocess9.2.ear)</td>
</tr>
</tbody>
</table>

Example 2 – Four JVMs

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

<table>
<thead>
<tr>
<th>Server number</th>
<th>SAS 9.2 Middle Tier Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server1</td>
<td>SAS WIP Applications (sas.wip.apps9.2.ear)</td>
</tr>
<tr>
<td></td>
<td>SAS WIP Content Server (sas.wip.scs9.2.ear)</td>
</tr>
<tr>
<td></td>
<td>SAS WIP Services (sas.wip.services9.2.ear)</td>
</tr>
<tr>
<td>Server2</td>
<td>SAS Portal (sas.portal4.2.ear)</td>
</tr>
<tr>
<td></td>
<td>SAS Package Viewer (sas.packageviewer4.2.ear)</td>
</tr>
<tr>
<td>Server3</td>
<td>SAS Web Report Studio (sas.webreportstudio4.2.ear)</td>
</tr>
<tr>
<td></td>
<td>SAS Stored Process (sas.storedprocess9.2.ear)</td>
</tr>
<tr>
<td>Server4</td>
<td>SAS BI Dashboard (sas.bidashboard4.2.ear)</td>
</tr>
<tr>
<td></td>
<td>SAS Shared Services (sas.shared9.2.ear)</td>
</tr>
<tr>
<td></td>
<td>SAS Web OLAP Viewer (sas.webolapviewer4.2.ear)</td>
</tr>
</tbody>
</table>
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 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. Configure the JAAS Login Module for Application Logins – PFS and SCS.
8. 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/Lev1/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/Lev1/Web/Staging/sasthemes.ear:
   ```
   jar xf /opt/SAS/Config/Lev1/Web/Staging/sasthemes.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.
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
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 9 – 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.

<table>
<thead>
<tr>
<th>JVM Option</th>
<th>Recommended Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Dhttp.proxyHost</td>
<td>proxy server host name</td>
<td>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=&lt;host name&gt; -Dhttp.proxyPort=80 -Dhttp.nonProxyHosts= &quot;*.example.com</td>
</tr>
<tr>
<td>-Dhttp.proxyPort</td>
<td>proxy server port</td>
<td></td>
</tr>
<tr>
<td>-Dhttp.nonProxyHosts</td>
<td>non-proxy host (exclude list)</td>
<td></td>
</tr>
<tr>
<td>-Dsas.svcs.http.max.connections</td>
<td>20 (default value is 2)</td>
<td>The recommended value is based on an environment with a large number of concurrent users.</td>
</tr>
<tr>
<td>-Djgroups.bind_addr</td>
<td>host name or IP address</td>
<td>Use this option on a system that has multiple network interfaces (physical or virtual). When this option is used, also use -Dresolve.dns=false.</td>
</tr>
<tr>
<td>-Dmulticast_udp_ip_ttl</td>
<td>30 (default value is 64)</td>
<td>This option sets the time to live on UDP packets that are used for multicast calls.</td>
</tr>
<tr>
<td>-Dlogoff.corePoolSize</td>
<td>50</td>
<td>Use this option in an environment where</td>
</tr>
<tr>
<td>Parameter</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>-Dlogoff.maxPoolSize</td>
<td>50</td>
<td>Use this option with the -Dlogoff.corePoolSize parameter if the concurrent user connections and the frequency of user logoffs is significantly high (for example, over 500 connections and logoffs).</td>
</tr>
<tr>
<td>-Dcom.sas.iom.orb.SelectionHub.clientThreadCount</td>
<td>3</td>
<td>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.</td>
</tr>
</tbody>
</table>
Chapter 10 – 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.

<table>
<thead>
<tr>
<th>Web Application Server</th>
<th>Parameter</th>
<th>Recommended Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JBoss</td>
<td>Thread Pool</td>
<td>25 min and 75 max</td>
<td>Controls the number of threads in the Web container thread pool.</td>
</tr>
<tr>
<td>JBoss</td>
<td>Disable deployment scanner</td>
<td></td>
<td>Disable the deployment scanner that is used for hot deployment and undeployment.</td>
</tr>
<tr>
<td>WebLogic</td>
<td>-Dweblogic.ChunkSize</td>
<td>a multiple of the network maximum transfer unit (MTU), minus the TCP header size</td>
<td>Set this value as the number of bytes and provide it with the JVM options for the server start up.</td>
</tr>
<tr>
<td>WebLogic</td>
<td>Accept Backlog</td>
<td></td>
<td>Increase the default value by 25% when you see “connection refused or dropped” message at the client, but no errors on the server.</td>
</tr>
<tr>
<td>WebSphere</td>
<td>Thread Pool</td>
<td>25 min and 75 max</td>
<td>Controls the number of threads in the Web container thread pool.</td>
</tr>
<tr>
<td>WebSphere</td>
<td>HttpSessionIdReuse</td>
<td>true</td>
<td>Add this custom property to the JVM options for the application server.</td>
</tr>
<tr>
<td>WebSphere</td>
<td>prependSlashToResource</td>
<td>true</td>
<td>Add this custom property to the application server.</td>
</tr>
<tr>
<td>WebSphere</td>
<td>com.ibm.ws.webcontainer.channelwritetype</td>
<td>true</td>
<td>Add this custom property to the Web container.</td>
</tr>
<tr>
<td>WebSphere</td>
<td>JSP enable class reloading</td>
<td>disable</td>
<td>Disable this property for each of the SAS Web applications.</td>
</tr>
</tbody>
</table>
Chapter 11 – 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:


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.

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

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

<table>
<thead>
<tr>
<th>JVM Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Xms</td>
<td>Minimum heap</td>
</tr>
<tr>
<td>-Xmx</td>
<td>Maximum heap</td>
</tr>
</tbody>
</table>
### 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:

```bash
# 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 Oracle WebLogic Server, 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/Levnen/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 version 6.1 and version 7.0. For more information, see the IBM WebSphere Application Server documentation.

1. On the left panel, expand the **Servers** node For WebSphere Application Server 6.1. For WebSphere Application Server 7.0, there is an additional expansion needed called **Server Types**.
2. Click **Application Servers** and then click the name of the server that you want to configure.
3. For WebSphere 6.1, select **Process Definition > Java Virtual Machine**. For WebSphere 7.0, select **Java and Process Management > 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 12 – 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
ndd -set /dev/tcp tcp_time_wait_interval 60000
ndd -set /dev/tcp tcp_conn_req_max_q 256
ndd -set /dev/tcp tcp_conn_req_max_q0 2500
ndd -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:

```
ndd -set /dev/tcp tcp_conn_request_max 8192
ndd -set /dev/tcp tcp_keepalive_interval 7200000
ndd -set /dev/tcp tcp_keepalives_kills 1
```

In addition, SAS recommends tuning the kernel parameters by using the SAM utility or the kctune utility.
<table>
<thead>
<tr>
<th>Kernel Parameter</th>
<th>Recommended Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>swchunk</td>
<td>8192</td>
</tr>
<tr>
<td>shmsseg</td>
<td>512</td>
</tr>
<tr>
<td>maxdsize</td>
<td>3221225472</td>
</tr>
<tr>
<td>maxdsize_64bit</td>
<td>64424509440</td>
</tr>
<tr>
<td>maxfiles_lim</td>
<td>10000 (Change this one before maxfiles.)</td>
</tr>
<tr>
<td>maxfiles</td>
<td>8192</td>
</tr>
<tr>
<td>semume</td>
<td>512</td>
</tr>
<tr>
<td>semmsl</td>
<td>3072</td>
</tr>
<tr>
<td>msgssz</td>
<td>512 (Change this one before msgmax)</td>
</tr>
<tr>
<td>nkthread</td>
<td>10000</td>
</tr>
<tr>
<td>max_thread_proc</td>
<td>4096</td>
</tr>
<tr>
<td>nproc</td>
<td>8192 (Change this one before maxuprc)</td>
</tr>
<tr>
<td>maxuprc</td>
<td>4096</td>
</tr>
<tr>
<td>nflocks</td>
<td>11585</td>
</tr>
<tr>
<td>ninode</td>
<td>8110</td>
</tr>
<tr>
<td>msgmap</td>
<td>13109</td>
</tr>
<tr>
<td>msgseg</td>
<td>32767 (Change this one before msgmax)</td>
</tr>
<tr>
<td>msgmnb</td>
<td>65535 (0x10000) (Change this one before msgmax)</td>
</tr>
<tr>
<td>msgmnb</td>
<td>131070 (when running multiple JVMs on the same system)</td>
</tr>
<tr>
<td>msgmax</td>
<td>65535 (0x10000)</td>
</tr>
<tr>
<td>msgmax</td>
<td>131070 (when running multiple JVMs on the same system)</td>
</tr>
<tr>
<td>msgmni</td>
<td>4634</td>
</tr>
<tr>
<td>semmns</td>
<td>11586</td>
</tr>
<tr>
<td>semmni</td>
<td>8192</td>
</tr>
<tr>
<td>semmnu</td>
<td>8180</td>
</tr>
<tr>
<td>shmmx</td>
<td>1.85514E+11</td>
</tr>
<tr>
<td>shmmni</td>
<td>8192</td>
</tr>
<tr>
<td>STRMSGSZ</td>
<td>65535</td>
</tr>
<tr>
<td>dbc_max_pc</td>
<td>10</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TcpTimedWaitDelay</td>
<td>0x1e (30 seconds)</td>
</tr>
<tr>
<td>MaxUserPort</td>
<td>greater than 32768</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnableDynamicBacklog</td>
<td>1 decimal</td>
</tr>
<tr>
<td>MinimumDynamicBacklog</td>
<td>20 decimal</td>
</tr>
<tr>
<td>MaximumDynamicBacklog</td>
<td>1000 decimal</td>
</tr>
<tr>
<td>DynamicBacklogGrowthDelta</td>
<td>10 decimal</td>
</tr>
</tbody>
</table>

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.
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.