



SAS[®] 9.4 Web Applications: Tuning for Performance and Scalability, Third Edition

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2016. *SAS® 9.4 Web Applications: Tuning for Performance and Scalability, Third Edition*. Cary, NC: SAS Institute Inc.

SAS® 9.4 Web Applications: Tuning for Performance and Scalability, Third Edition

Copyright © 2016, SAS Institute Inc., Cary, NC, USA

All Rights Reserved. Produced in the United States of America.

For a hard copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

November 2016

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.

9.4-P1:appsrvtuning

Contents

<i>Using This Book</i>	v
<i>Accessibility</i>	vii
Chapter 1 / Overview	1
About Performance Tuning	1
Enhancing the Performance of SAS Web Applications	1
Chapter 2 / Clustering SAS Web Application Servers	3
About Clustering SAS Web Application Servers	3
Types of Supported Clusters	4
Cluster Configurations	5
Chapter 3 / Tuning SAS Web Application Server	7
About Tuning SAS Web Application Server	7
Tuning Recommendations for SAS Web Application Server	7
Chapter 4 / Tuning SAS Web Applications	9
About Tuning SAS Web Applications	9
Tuning Recommendations for SAS Visual Analytics Transport Service	9
Chapter 5 / Tuning the Java Virtual Machine	11
About Tuning the Java Virtual Machine	11
JVM Tuning Options	11
JVM Tuning Guidelines	14
Chapter 6 / Tuning the Operating System	15
About Tuning the Operating System	15
Operating Systems	15
Chapter 7 / Tuning the PostgreSQL Data Server	21
About the Server	21
Tuning Recommendations for the Server	22
Initial Configuration Recommendations	24
Moving the Write-Ahead Log (WAL)	27
Monitoring Performance	27
Configuring Standby Database Performance	30
Chapter 8 / Monitoring and Troubleshooting	31
About Monitoring and Troubleshooting	31
Using SAS Environment Manager to Monitor and Troubleshoot Data	31
Appendix 1 / Initial Sizing and Tuning Configuration Options for Deployments	33
About the Configuration Options	33
SAS Web Application Server Single Server Settings	34
SAS Web Application Server Multiple Server Settings	35
SAS Web Server Settings	44
Recommended Reading	47

Glossary	49
Index	51

Using This Book

Audience

This document helps SAS administrators optimize the performance of the web application servers running in the middle-tier environment.

This document is organized as follows:

- A brief overview of the web applications in the middle tier
- Information about the types of SAS Web Application Server clusters
- Tuning information for the following applications and servers:
 - SAS Web Server
 - SAS Web Application Server
 - Supported Operating Systems

Requirements

This guide contains tasks that are tested and proven. SAS recommends that you properly test all planned configurations in a testing environment before applying them to a production environment. You should also back up all your data and pre-tuning configurations.

Accessibility

For information about the accessibility of any of the products mentioned in this document, see the usage documentation for that product.

1

Overview

<i>About Performance Tuning</i>	1
<i>Enhancing the Performance of SAS Web Applications</i>	1

About Performance Tuning

The SAS 9.4 middle-tier environment provides an execution environment for SAS web applications that are used to perform business analytics. The SAS 9.4 release has added functionality that enables you to automatically configure a clustered environment for the SAS web applications. As a result of this added functionality, optimizing the middle tier and SAS web applications for scalability and performance is greatly simplified.

The purpose of this document is to walk through the SAS 9.4 tuning for performance and scalability. This document includes the tuning methodology and tuning parameters for SAS web applications, SAS Web Application Server, Java Virtual Machine (JVM), and the supported operating systems.

Enhancing the Performance of SAS Web Applications

Performance requirements are usually identified in terms of transaction response time, number of transactions per second, throughput, resource utilization, total cost per transaction, availability, and more. Scalability often refers to the ability of a component 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.

In order to provide greater performance, scalability, and availability, SAS Web Application Server supports vertical clustering, horizontal clustering, and a combination of both. Vertical clusters are multiple web application instances that run on the same machine. Horizontal clusters are multiple server instances that run on a group of middle-tier server machines. The SAS Deployment Wizard can be used to automatically configure a vertical or horizontal cluster.

When clustering is configured for the SAS Web Application Server, a load-balancing process distributes requests among the server instances. Load balancing is automatically handled by the SAS Web Server. This helps ensure the availability of the SAS web applications if hardware or software failure occurs. For more information, see [“About Clustering SAS Web Application Servers”](#) on page 3.

2

Clustering SAS Web Application Servers

<i>About Clustering SAS Web Application Servers</i>	3
<i>Types of Supported Clusters</i>	4
Vertical Clusters	4
Horizontal Clusters	5
<i>Cluster Configurations</i>	5

About Clustering SAS Web Application Servers

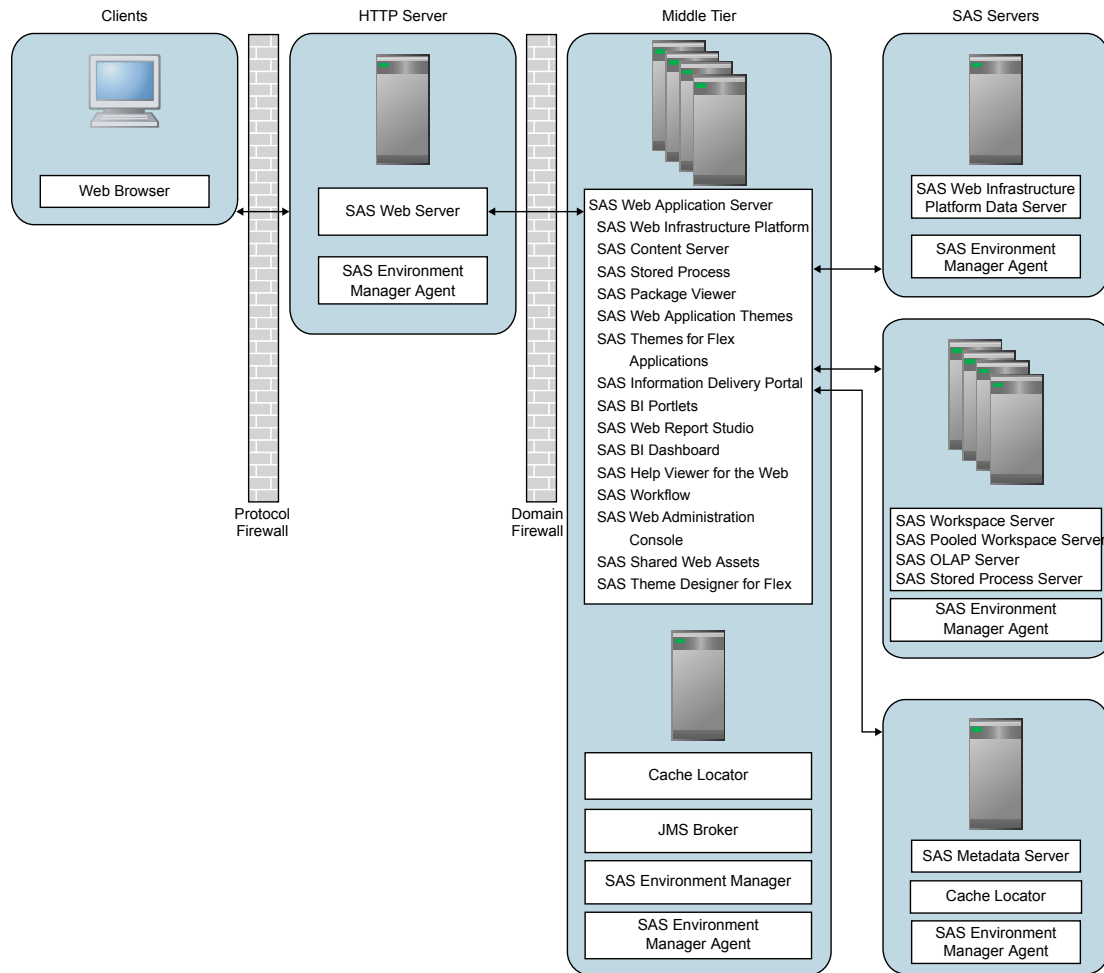
The SAS web applications can be clustered to improve performance and provide high availability. In a clustered environment, multiple server instances are used to handle client requests and transactions, thus improving performance. Performance can also be improved by deploying servers on to additional hardware. Clustering can also provide high availability by ensuring that the server will continue to operate if a server host machine fails.

The server instances in a cluster can coexist on the same machine (vertical clustering), or the server instances can run on a group of middle-tier server machines (horizontal clustering). The web applications can be deployed on vertical clusters, horizontal clusters, and a combination of both.

SAS Web Server is used to provide load balancing by distributing HTTP requests to SAS Web Application Server instances. The web server is the unique access point for customers to access all SAS web applications. The server detects when an application server in the cluster is down and routes requests to other nodes. However, SAS Web Server does not monitor the availability of individual web applications, nor does it monitor the health of an application server that is running.

The following figure illustrates multiple web applications that are deployed across a SAS Web Application Server cluster.

Figure 2.1 Clustered Web Application Servers



For more information about clustering, contact Technical Support.

Types of Supported Clusters

Vertical Clusters

Vertical clustering is the practice of deploying multiple identically configured web application server instances on a single machine. This can assist with improving performance as long as the hardware is sufficiently powerful to run additional server instances. It can also offer some improvement for availability. In the event that one web application server instance crashes (or an application on one server instance stops), the applications remain available on the other web application server instances.

Horizontal Clusters

Horizontal clustering is the practice of deploying SAS Web Application Server instances on multiple machines. This can assist with improving performance and provide greater availability to guard against hardware failure. In the event that one machine or web application server instance crashes (or an application on one server instance stops), the applications remain available on the other machines.

Cluster Configurations

There are two general deployment topologies for the middle tier of the SAS Intelligence Platform: single server (which results in a homogeneous cluster) and multiple server (which results in a heterogeneous cluster). Many deployments default to single server topology, unless the default is overridden in the corresponding SAS Deployment Wizard custom prompting level. By default, some SAS products trigger the multiple server topology when they are included in the plan.

Single server topology results in clustered nodes that all contain the same applications that can be clustered. Conversely, multiple server topology results in specific applications being deployed to different server instances. Each server instance can be clustered independently, providing the ability to allocate additional resources to the applications and application groups that are more heavily used.

Note: Whether the single or multiple server topology is selected, both vertical and horizontal clusters are still possible, as is a combination of both clustering techniques. The only difference is the manner in which the applications are distributed to the server instances.

3

Tuning SAS Web Application Server

<i>About Tuning SAS Web Application Server</i>	7
<i>Tuning Recommendations for SAS Web Application Server</i>	7

About Tuning SAS Web Application Server

You can improve the performance of SAS web applications by configuring other aspects of the behavior of the web application server. For example, to improve the performance of any web application, make sure that the server can create sufficient threads to service incoming requests and limit the frequency with which servers check for updated JavaServer Pages and servlets.

Tuning Recommendations for SAS Web Application Server

You can specify application server options to improve the performance of SAS web applications. This chapter provides a set of application server options that are relevant to the SAS 9.4 middle tier. A description is provided for each option to help you decide whether modifying a value is appropriate for your deployment.

The following options can be modified in the *SAS-configuration-directory* \Lev\Web\WebAppServer\SASServer1_n\conf\server.xml file:

maxThreads Option

Component: tomcatThreadPool Executor

Default Value: 300

Recommended Value: 300–1024

Description: Specifies the number of threads in the executor thread pool that is used to process incoming requests. If you have multiple instances of SAS Web Application Server, make the same changes in each of the files for all the servers that you want to tune.

Additional Information: The thread pool utilization can be monitored using SAS Environment Manager. For more information, see [“Using SAS Environment Manager to Monitor and Troubleshoot Data”](#) on page 31. If

the number of active threads approaches the maximum value (within 5–10 percent), the value should be increased.

maxPoolSize Option

Component: Various JDBC Resources, such as `sas/jdbc/SharedServices`

Default Value: 100

Recommended Value: 50–512

Description: Specifies the maximum number of pooled connections to the database that is associated with a given data source reference.

Additional Information: When the pool becomes exhausted, the SAS web applications log files (for example, `SASWIPServices9.4.log` and `SASPrincipalServices9.4.log`) might contain messages like the following example:

```
Could not get JDBC Connection;  
nested exception is com.atomikos.jdbc.AtomikosSQLException:  
Connection pool exhausted - try increasing 'maxPoolSize'  
and/or 'borrowConnectionTimeout' on the DataSourceBean.
```

Also, the configuration of the database server might need to be modified, along with this setting. For example, the default configuration of the SAS Web Infrastructure Platform Data Server allows a maximum of 256 connections. You can change the `max_connections` setting to be the same value as the sum of the `maxPoolSize` option value across all data sources on all server instances. If you change the value, a system restart is required. For more information about the `max_connections` setting, see [“Initial Configuration Recommendations” on page 24](#).

The following option can be added to the `SAS-configuration-directory\Lev\Web\WebAppServer\SASServer1_n\lib\jta.properties` file:

com.atomikos.icatch.checkpoint_interval Option

Component: Atomikos transaction manager

Default Value: 500

Recommended Value: 50

Description: Specifies the interval between transaction manager checkpoints. During checkpoints, completed transactions are removed from the recovery log. More frequent checkpoints can reduce the contention on the log file across threads in high volume usage scenarios.

4

Tuning SAS Web Applications

<i>About Tuning SAS Web Applications</i>	9
<i>Tuning Recommendations for SAS Visual Analytics Transport Service</i>	9

About Tuning SAS Web Applications

You can improve the speed and performance of SAS web applications by tuning them. SAS web applications can be browser-based applications or REST-based services that are invoked by native applications running on mobile or desktop clients. The required tuning varies by application.

Tuning Recommendations for SAS Visual Analytics Transport Service

Starting with SAS Visual Analytics 7.3, the following options are available to tune SAS Visual Analytics Transport Service. To tune the Transport Service application, follow these steps:

- 1 Log on to SAS Management Console.
- 2 On the **Plug-ins** tab, select **Application Management** ► **Configuration Manager** ► **SAS Application Infrastructure** ► **Visual Analytics *version*** ► **Visual Analytics Services *version***.
- 3 Right-click **Visual Analytics Transport Service *version*** and select **Properties**.
- 4 Click the **Advanced** tab and specify one or both of the following properties and values:

Table 4.1 SAS Visual Analytics Transport Service Property Descriptions

Property Name	Default Value	Recommended Value	Description
vat.subscribe.concurrentRequestLimit.guest	5	10	Adjust this property on systems with heavy guest account access from SAS Mobile BI. This property controls the number of report subscribe requests that can be executed in parallel for the guest user.
vat.xmlParserPoolSize	0	200	Use this property to enable pooling of XML parsers for reuse. Parser pooling can be helpful in optimizing performance for report viewers in high-volume systems. This property identifies the number of parser instances to be initialized in the parser pool at start-up. A value of zero indicates that parser pooling is disabled.

- 5 Click **OK** to close the Visual Analytics Transport Service Version Properties window.

To enable these properties to take effect, restart SAS Web Application Server.

5

Tuning the Java Virtual Machine

About Tuning the Java Virtual Machine	11
JVM Tuning Options	11
JVM Option Descriptions	11
JVM Tuning Options	12
JVM Tuning Options for SAS Web Applications	13
JVM Tuning Options for SAS Web Application Server	13
JVM Tuning Guidelines	14
JVM Tuning Guidelines for HP-UX, Linux, Solaris, and Windows Platforms ...	14
JVM Tuning Guidelines for AIX	14

About Tuning the Java Virtual Machine

The goal of Java Virtual Machine (JVM) tuning is to improve performance in production servers, particularly in the area of memory usage and garbage collection cycles. The goal is to also maximize the number of clients that the SAS web applications can support.

JVM Tuning Options

JVM Option Descriptions

The following table provides descriptions of the JVM options that you will tune in the SAS 9.4 middle-tier environment.

Table 5.1 JVM Tuning Descriptions

JVM Option	Description
-Xms	Specifies the minimum heap size.
-Xmx	Specifies the maximum heap size.

JVM Option	Description
<code>-XX:PermSize</code>	Specifies the initial permanent generation storage size. Applies only to the Java Virtual Machine on HP-UX, Linux, Solaris, and Windows platforms.
<code>-XX:MaxPermSize</code>	Specifies the maximum permanent generation size. Applies only to the Java Virtual Machine on HP-UX, Linux, Solaris, and Windows platforms.

JVM Tuning Options

Note: The following tuning options apply to the first maintenance release of SAS 9.4 and earlier. They also apply to servers that do not support the initial tunable parameters as documented in [Appendix 1, “Initial Sizing and Tuning Configuration Options for Deployments,”](#) on page 33.

The following table summarizes the general guidelines for tuning JVM options. These values are recommended initial settings for the tunable arguments. However, individual usage patterns vary and might benefit from additional tuning.

Table 5.2 JVM Tuning Options

Deployment Topology	Server	Platform	JVM Option and Recommended Value
Single Server	All	HP-UX, Linux, Solaris, and Windows	<code>-Xms4096m</code> <code>-Xmx4096m</code> <code>-XX:PermSize:1280m</code> <code>-XX:MaxPermSize:1280m</code>
Single Server	All	AIX	<code>-Xms2048m</code> <code>-Xmx4096m</code>
Multiple Server	SASServer1 Instances	HP-UX, Linux, Solaris, and Windows	<code>-Xms4096m</code> <code>-Xmx4096m</code> <code>-XX:PermSize:768m</code> <code>-XX:MaxPermSize:768m</code>
Multiple Server	Other Server Instances	HP-UX, Linux, Solaris, and Windows	<code>-Xms2048m</code> <code>-Xmx2048m</code> <code>-XX:PermSize:512m</code> <code>-XX:MaxPermSize:512m</code>
Multiple Server	SASServer1 Instances	AIX	<code>-Xms2048m</code> <code>-Xmx4096m</code>

Deployment Topology	Server	Platform	JVM Option and Recommended Value
Multiple Server	Other Server Instances	AIX	-Xms1024m -Xmx2048m

Monitoring the server data can be combined with the recommended settings to further fine-tune the servers. For more information, see [Chapter 8, “Monitoring and Troubleshooting,”](#) on page 31.

JVM Tuning Options for SAS Web Applications

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 5.3 JVM Tuning Parameters

JVM Option	Recommended Value	Description
-Dhttp.proxyHost	Proxy server host name. For example: -Dhttp.proxyHost=hostname	The HTTP proxy options are used by the SAS Information Delivery Portal to access document type definitions (DTDs) over the Internet. Use these options 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.
-Dhttp.proxyPort	Proxy server port. For example: -Dhttp.proxyPort=80	
-Dhttp.nonProxyHosts	Non-proxy host (exclude list). For example: -Dhttp.nonProxyHosts="*.example.com localhost"	
-Djgroups.bind_addr	Host name or IP address. For example: -Djgroups.bind_addr=hostname	Use this option on a system that has multiple network interfaces (physical or virtual). When this option is used and an IP address is specified for the bind_addr value, you can also specify: -Dresolve.dns=false
-Dmulticast_udp_ip_ttl	1 (default value is 64)	Use this option to restrict multicast traffic to a single subnet, based on your network topology.

JVM Tuning Options for SAS Web Application Server

In order to help ensure that you do not encounter problems with JVM memory for some advanced configuration procedures, you might need to change the JVM options for the server.

When SAS Web Application Server is deployed on UNIX, the JVM options are specified in the `JVM_OPTS` environment variable, which is specified in the `SAS-configuration-directory/Levn/Web/WebAppServer/SASServer1_n/bin/setenv.sh` file. If you have multiple instances of SAS Web Application Server, make the appropriate changes in each of the files.

For Windows deployments, the JVM options can be executed from a script or deployed as a service. When executed from a script, the JVM options are specified in the `JAVA_OPTS` environment variable, in the `SAS-configuration-directory\Levn\Web\WebAppServer\SASServer1_n\bin\setenv.bat` file. For Windows deployments that are deployed as a service, the JVM options are specified in the `SAS-configuration-directory\Levn\Web\WebAppServer\SASServer1_n\conf\wrapper.conf` file. Create a new entry for any new JVM options. The following example shows a few lines from the file:

```
# Java Additional Parameters
wrapper.java.additional.7=-Xmx4096m
wrapper.java.additional.8=-Xss256k
wrapper.java.additional.9=-Xms1024m
wrapper.java.additional.10=-XX:PermSize=768m
wrapper.java.additional.11=-XX:MaxPermSize=1280m
```

If you have multiple instances of SAS Web Application Server, make the same changes in each of the files.

JVM Tuning Guidelines

JVM Tuning Guidelines for HP-UX, Linux, Solaris, and Windows Platforms

The following tuning guidelines apply to HP-UX, Linux, Solaris, and Windows platforms:

- Increase the value of the `-Xms` option to equal that of the `-Xmx` option. These values can be increased if there is enough demand, but ideally they should be set to the same value to avoid the overhead of expansion and contraction. Depending on the usage patterns observed, try to ensure that there is at least 500 MB heap free.
- Identify the amount of permanent generation that is required, and set the `-XX:PermSize` and `-XX:MaxPermSize` options to equal values sufficient for the needs of the environment. Permanent Generation should be sized as small as possible, leaving approximately 200–300 MB of unused space for class loading and unloading.

Some of these metrics are available in SAS Environment Manager. For more information, see [“Using SAS Environment Manager to Monitor and Troubleshoot Data” on page 31](#).

JVM Tuning Guidelines for AIX

Generally, for the AIX platform, set the value of the `-Xms` option to be no larger than 50% of the value of the `-Xmx` option, to avoid fragmentation from pinned objects. The `-Xmx` option might need to be increased in heavily used environments.

6

Tuning the Operating System

<i>About Tuning the Operating System</i>	15
<i>Operating Systems</i>	15
AIX	15
HP-UX	16
Linux	17
Solaris SPARC and Solaris x86-64	18
Windows	18
Additional Windows System Configurations	19

About Tuning the Operating System

There are a number of configuration changes and variables that you can set to tune the SAS 9.4 middle tier for your performance and scalability needs. The following sections show how to configure the settings that are relevant to the SAS 9.4 middle tier and SAS Web Application Server.

Note: For UNIX and Linux operating environments, you must confirm that the ulimit resource limit for open file descriptors is set to at least 20480.

Operating Systems

AIX

Use the following commands to set the SAS recommended TCP/IP settings:

```
/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
```

In some environments, the maximum number of user processes might need to be increased, particularly on the system running the SAS Web Infrastructure Platform Data Server when the `max_connections` value is increased. The default value is 128.

To increase the `max_connections` value, run the following command:

```
chdev -l sys0 -a maxuproc=new_value
```

A new value of 1024 or 2048 (or higher) might be appropriate, depending on how many processes and connections are configured.

HP-UX

Use the following commands to set the SAS recommended TCP/IP settings:

```
ndd -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 following kernel parameters by using the SAM utility or the kctune utility.

Table 6.1 Recommended Kernel Parameters to Tune

Kernel Parameter	Recommended Value
swchunk	8192
shmseg	512
maxdsiz	3221225472
maxdsiz_64bit	64424509440
maxfiles_lim	10000 Note: Change this value before changing maxfiles.
maxfiles	8192
semume	512
semmsl	3072
msgssz	512 Note: Change this value before changing msgmax.
nkthread	10000
nproc	10240 Note: Change this value before changing maxuprc.
maxuprc	4096
nflocks	11585
ninode	8110
msgmap	13109

Kernel Parameter	Recommended Value
msgseg	32767 Note: Change this value before changing msgmax.
msgmnb	65535 (0x10000) Note: Change this value before changing msgmax.
msgmnb	131070 Note: Set when running multiple JVMs on the same system.
msgmax	65535 (0x10000)
msgmax	131070 Note: Set when running multiple JVMs on the same system.
msgmni	4634
semms	11586
semgni	8192
semgnu	8180
shmmax	1.85514E+11
shmmni	8192
STRMSGSZ	65535
dbc_max_pc	10
nstrpty	60
cmc_plat_poll	15
msgtql	13107

Linux

Use the following commands to temporarily set the SAS recommended TCP/IP settings:

```
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
```

Note: These settings will be lost upon rebooting your system.

Use the following commands to permanently set the SAS recommended TCP/IP settings:

```
/sbin/sysctl -w net.ipv4.tcp_fin_timeout=30
/sbin/sysctl -w net.core.netdev_max_backlog=3000
/sbin/sysctl -w net.core.somaxconn=3000
/sbin/sysctl -w net.ipv4.tcp_keepalive_intvl=15
/sbin/sysctl -w net.ipv4.tcp_keepalive_probes=5
```

In addition, set the operation system limits by accessing the `/etc/security/limits.conf` file. Add or change the values, based on the following information:

Table 6.2 Linux System Limits

Domain	Type	Item	Recommended Value
*	—	nproc	10240
*	—	stack	10240
*	—	nofile	20480

Some Linux distributions include additional settings in *.conf files in the `/etc/security/limits.d` directory. The values specified in the files in this directory override the limits specified in the `limits.conf` file. For example, some distributions set an explicit soft limit for the `nproc` `ulimit` value. This is specified in the `/etc/security/limits.d/90-nproc.conf` file. In this file, change the `nproc` value to correspond to the value specified in the preceding table.

Solaris SPARC and Solaris x86-64

Use the following commands to set the SAS recommended TCP/IP settings:

```
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
```

Windows

The Windows registry must be updated. Microsoft recommends performing a system backup before editing the registry. To set the SAS recommended parameters, use the REGEDIT command as follows:

- 1 Access the `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\Tcpip\Parameters` registry subkey. Add the DWORD value with a name of `TcpTimedWaitDelay` and a value of 30 (0x1e).

- 2 Access the `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\AFD\Parameters` registry subkey. Add the following DWORD values:

Table 6.3 AFD Service Parameters

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

The recommended values specify the number of connections that you want to be available. These values request a minimum of 20 and a maximum of 1000 available connections. The number of available connections is increased by 10 each time there are fewer than the minimum number of available connections.

- 3 In addition, the user port range should be updated. From a command prompt, run the following commands, based on the version of your Internet protocol:

```
netsh int ipvn set dynamicport tcp start=32768 num=32767
netsh int ipvn set dynamicport udp start=32768 num=32767
```

where *n* indicates the version of your Internet protocol and is either 4 or 6.

- 4 Restart Windows.

Additional Windows System Configurations

The following list includes general recommendations for configuring Windows systems:

- Disable Windows indexing on any directories that are used by SAS software.
- Set Windows performance settings so that background processes are favored.
- Set the maximum power profile in the system BIOS for all systems, except Intel Sandy Bridge.
- Disable the C1E BIOS setting on Dell systems.

7

Tuning the PostgreSQL Data Server

About the Server	21
Tuning Recommendations for the Server	22
Database Sizes	22
Tuning a Database	22
Tuning Multiple Data Server Instances	23
Recommendations for Semaphores	23
Additional Documentation for PostgreSQL Tuning	23
Initial Configuration Recommendations	24
Recommendations Based on the Database Size	24
Recommendations for Special Considerations	26
Large Database Considerations	26
Moving the Write-Ahead Log (WAL)	27
Monitoring Performance	27
About Performance Views	27
Resetting the Data	28
Displaying the Cache Hit Ratio	28
Finding Missing Indexes	28
Showing How the Database Executes a Query	28
Logging All Statements That Take More Than 100 Milliseconds	28
Seeing All Indexes in Your Tables	29
Determining the Last Time Vacuum and Analyze Were Run	29
Identifying Currently Running Queries	29
Configuring Standby Database Performance	30

About the Server

The PostgreSQL data server provides a transactional store that is used by SAS middle-tier software. You might have one or more SAS data servers installed in your environment. The data servers will vary, based on the SAS products and solutions that you have purchased.

Tuning Recommendations for the Server

Database Sizes

The suggested configurations in this chapter are based on the following database size definitions:

- A small database receives more than one concurrent query or update, and creates hundreds of records daily.
- A medium database receives several concurrent queries or updates (most of which use indexes), and creates thousands of records per hour.
- A large database receives many concurrent queries or updates, creates thousands of records per minute, receives many queries to scan tables, and processes complex queries and regular bulk loads.

Tuning a Database

To tune a database:

- 1 Tune the system and operating system that supports the database.

The PostgreSQL data server benefits from the same system tuning as SAS. A fast disk resource, sufficient memory for caching, and a sufficient number of cores with good single thread performance are crucial. Operating system tuning should favor input/output. In addition, set values for semaphores. For more information, see [“Recommendations for Semaphores” on page 23](#).

For detailed operating system configuration information, see the appropriate operating system documentation.

- 2 Tune the database configuration.

The data server automatically uses available hardware efficiently. In many cases, there is little to be gained by trying to tune the server configuration parameters. The default tuning handles a wide variety of use cases. In very few cases, configuration tuning makes a significant difference, but in most cases, configuration tuning does not change performance.

For configuration changes such as shared buffers that can help improve performance, see [“Initial Configuration Recommendations” on page 24](#).

- 3 Tune the database schema design.

This includes the following tasks: database normalizing or de-normalizing; creating artificial keys, where appropriate; creating indexes on frequent access paths, including foreign key; defining index types (expression index and multiple column versus multiple single column); and partitioning tables.

- 4 Tune the queries or access.

This includes the following tasks: configuring connection pooling, using indexed paths and prepared statements, setting fetch size, and running batch updates and analysis to update statistics.

To check and monitor performance, see “[Monitoring Performance](#)” on page 27.

Tuning Multiple Data Server Instances

If you have multiple data server instances, add the individual requirements for shared buffers to determine the total amount of memory for your shared buffer configuration.

Recommendations for Semaphores

Recent versions of Linux are shipped with appropriate semaphore values. Review the recommended values for semaphore settings that are provided in the following table and ensure that your settings match the recommended values:

Table 7.1 Recommended Semaphore Values

Setting	Recommended Value
max number of arrays	256
max semaphores per array	250
max semaphores system wide	32000
max ops per semop call	32
semaphore max value	32767

For more information about semaphore settings on different operating systems, see <http://www.postgresql.org/docs/9.1/static/kernel-resources.html>.

Additional Documentation for PostgreSQL Tuning

For detailed information about tuning PostgreSQL, see the following documentation:

- Performance Optimization: https://wiki.postgresql.org/wiki/Performance_Optimization
- Tuning Your PostgreSQL Server: https://wiki.postgresql.org/wiki/Tuning_Your_PostgreSQL_Server
- Managing Kernel Resources: <http://www.postgresql.org/docs/9.1/static/kernel-resources.html>

Initial Configuration Recommendations

Recommendations Based on the Database Size

To implement the recommended configurations, the default `postgresql.conf` file (which resides in the data directory) must be updated. For information about PostgreSQL file locations, see <http://www.postgresql.org/docs/9.1/static/storage-file-layout.html>. After updating the file, you must restart the object spawner, middle tier, and database.

The following configuration values must be updated, based on the database size. For more information about how to determine the database size, see “Database Sizes” on page 22.

Note: The `shared_buffers` setting configures the amount of memory the database server uses for shared memory buffers.

Table 7.2 Recommended Configuration Based on Database Size

Database Size	Setting	Recommended Value
Initial configuration for most databases	<code>shared_buffers</code>	256 MB
	<code>work_mem</code>	2 MB
	<code>maintenance_work_mem</code>	64 MB
	<code>wal_buffers</code>	16 MB
	<code>checkpoint_segments</code>	16
	<code>effective_cache_size</code>	2 GB
	<code>checkpoint_completion_target</code>	0.9
	<code>max_connections</code>	256

Database Size	Setting	Recommended Value
Small	shared_buffers	32 MB
	work_mem	1 MB
	maintenance_work_mem	16 MB
	wal_buffers	not applicable
	checkpoint_segments	8
	effective_cache_size	128 MB
	checkpoint_completion_target	0.5
	max_connections	100
	max_prepared_transactions	100
	Medium	shared_buffers
work_mem		2MB
maintenance_work_mem		64 MB
wal_buffers		16 MB
checkpoint_segments		16
effective_cache_size		2 GB
checkpoint_completion_target		0.9
max_connections		256
max_prepared_transactions		256

Database Size	Setting	Recommended Value
Large Note: For additional parameters to consider, see “Large Database Considerations” on page 26.	shared_buffers	4 GB
	work_mem	16 GB
	maintenance_work_mem	128 MB
	wal_buffers	16 MB
	checkpoint_segments	32
	effective_cache_size	4 GB
	checkpoint_completion_target	0.9
	max_connections	512
	max_prepared_transactions	500

Recommendations for Special Considerations

Specialized solutions or use cases might require further configuration tuning. If you need to experiment with the parameters for your optimized system performance, the most important parameters are:

shared_buffers

Specifies the amount of memory to be used for caching data. PostgreSQL also benefits from the file system cache, so `shared_buffers` should not be so large that it interferes with the file system cache. For a large database, set this parameter between 1 GB and up to 25% of the total system memory.

work_mem

Specifies the amount of memory to be used for sorts, hashing, and materialization, before writing to temporary disk files. Several running sessions can perform operations concurrently. Therefore, the total memory used might be many times the value of `work_mem`. Keep this in mind when choosing the value for this parameter. Set this parameter between 16 MB and 64 MB or more, for a specialized use case (for example, frequent very large sorts).

maintenance_work_mem

Specifies the maximum amount of memory to be used for vacuuming (reclaiming storage used by rows marked for delete) and index builds. For a large database, set this parameter to 256 MB or more.

If your application can tolerate losing a transaction if the computer or storage crashes, you can set the `synchronous_commit` parameter to **Off** for faster updates.

Large Database Considerations

For a large database, also consider the following parameters:

effective_cache_size

Specifies to the query optimizer how much memory is used by the file cache. Set this parameter to 50% of the physical memory.

checkpoint_segments

Specifies how many 16 MB write-ahead log (WAL) segments are buffered on disk before writing permanently to database tables. Set this parameter to 64 MB or more.

synchronous_commit

Specifies whether transaction commits wait for WAL records to be written to disk before the command returns a successful indication to the client. If you have an application that can tolerate losing a transaction if the computer or storage crashes, this parameter can be set to **Off** for faster updates.

Moving the Write-Ahead Log (WAL)

By default, the SAS Web Infrastructure Platform Data Server confirms each transaction is written to the WAL before a commit is complete. When you have a high volume of insert, update, or delete statements, consider moving the WAL to a separate disk resource, especially a solid-state drive (SSD) or other fast storage device.

To move the WAL directory to a separate drive or device, use symbolic links to point to the new location. The following example shows the commands to execute on a UNIX system:

```
pg_ctl stop $PGDATA
mkdir -p /mountpoint2/.../pg_xlog
mv /mountpoint1/.../pg_xlog/* /mountpoint2/.../pg_xlog/
rmdir /mountpoint1/.../pg_xlog
ln -s /mountpoint2/.../pg_xlog/ /mountpoint1/.../pg_xlog
pg_ctl start $PGDATA
```

where *mountpoint1* is the link name and *mountpoint2* is the new physical directory.

The following example shows the commands to execute on a Windows system:

```
pg_ctl stop $PGDATA
mkdir -p /mountpoint2/.../pg_xlog
mv /mountpoint1/.../pg_xlog/* /mountpoint2/.../pg_xlog/
rmdir /mountpoint1/.../pg_xlog
mklink /J C:\PostgreSQL\8.3\data\pg_xlog Z:\pg_xlog
pg_ctl start $PGDATA
```

Monitoring Performance

About Performance Views

There are several helpful performance views. The performance views display statistics that are gathered as your database is used. These views are helpful

when a characteristic load has been applied to your database for a reasonable period.

Resetting the Data

You can reset the data in the performance views. To reset all statistics in the database:

```
SELECT pg_stat_reset();
```

To reset statistics for a single table:

```
SELECT pg_stat_reset_single_table_counters(oid);
```

Displaying the Cache Hit Ratio

The buffer cache hit ratio should be about 99%. If it is not, you might need to increase its size. To see the cache hit ratio for tables:

```
SELECT sum(heap_blks_read) as heap_blks_read, sum(heap_blks_hit) as heap_blks_hit,
       (sum(heap_blks_hit) - sum(heap_blks_read)) / sum(heap_blks_hit) as hit_ratio
FROM pg_statio_user_tables;
```

To see the cache hit ration for indexes:

```
SELECT sum(idx_blks_read) as idx_blks_read, sum(idx_blks_hit) as idx_blks_hit,
       (sum(idx_blks_hit) - sum(idx_blks_read)) / sum(idx_blks_hit) as hit_ratio
FROM pg_statio_user_indexes;
```

Finding Missing Indexes

For any table over several thousand rows, the percent of time the index is used to access it should typically be near 99%. To find the missing indexes:

```
SELECT relname, 100 * idx_scan / (seq_scan + idx_scan) percent_of_times_index_used,
       n_live_tup rows_in_table
FROM pg_stat_user_tables WHERE seq_scan + idx_scan > 0 ORDER BY n_live_tup DESC;
```

Showing How the Database Executes a Query

You can see whether a particular query does a sequential scan instead of using an index:

```
EXPLAIN ANALYZE SELECT * from my_table where filer_condition;
```

For details about reading EXPLAIN output, see <http://www.postgresql.org/docs/9.2/interactive/using-explain.html>.

Logging All Statements That Take More Than 100 Milliseconds

- 1 Edit the `postgresql.conf` file.
- 2 Update the values of the following configuration settings:

```
log_min_duration_statement = 100
log_line_prefix = '%t [%p]: [%l-1] '
```

- 3 Reload the server by issuing the `pg_ctl reload` command. A server reload does not terminate existing sessions, whereas a restart does.
- 4 Run a characteristic load to generate information in the log.
 - To log fewer statements, increase the value of `log_min_duration_statement`.
 - To log all statements, set the value to `0`.
 - To turn off logging, set the value to `-1`.

You can view the log manually or use a tool like pgBadger. For more information about pgBadger, see <http://dalibo.github.com/pgbadger/>.

Seeing All Indexes in Your Tables

To see the indexes in your tables:

```
SELECT n.nspname as "Schema" , t.relname as "Table",c.relname as "Index"
FROM pg_catalog.pg_class c
      JOIN pg_catalog.pg_namespace n on n.oid = c.relnamespace
      JOIN pg_catalog.pg_index i on i.indexrelid = c.oid
      JOIN pg_catalog.pg_class t on i.indrelid = t.oid
WHERE c.relkind = 'i' and n.nspname not in ('pg_catalog', 'pg_toast') and
      pg_catalog.pg_table_is_visible(c.oid)
ORDER BY n.nspname,t.relname,c.relname;
```

Determining the Last Time Vacuum and Analyze Were Run

Vacuuming removes rows marked for delete and analyze updates the statistics on the table used by the query optimizer. These operations should be run automatically at appropriate times. If any index is not being used, or you observe unusual slowness, you should check that they have been run recently.

```
SELECT relname, last_vacuum, last_autovacuum, last_analyze, last_autoanalyze
FROM pg_stat_user_tables ORDER BY relname;
```

Identifying Currently Running Queries

To see how long a query has been running:

```
SELECT procpid, current_timestamp - xact_start AS xact_runtime, current_query
FROM pg_stat_activity where xact_start is not null
ORDER BY xact_start;
```

To see the most common currently running queries:

```
SELECT count(*) as cnt, current_query FROM pg_stat_activity
GROUP BY current_query
ORDER BY cnt DESC;
```

Configuring Standby Database Performance

For reliability, the SAS Web Infrastructure Platform Data Server supports several types of database replication. The simplest form of replication is a standby database, and it has the least impact on database performance. A standby database is an exact copy of the Data Server database and of the supported types of replication. To support standby, the Data Server database functions by writing to the transaction log. Sending the logs might consume resources on the same host as the database, but normal database operations should not be slowed if hardware resources are sufficient.

You can have either a warm standby instance or a hot standby instance of the SAS Web Infrastructure Platform Data Server. A warm standby instance can quickly be enabled as a production database. A hot standby instance allows read-only queries even while in standby mode. It is simple to set up log shipping in a warm standby. A hot standby can use log shipping, but is kept more up-to-date with streaming replication. In either case, you should set the size of the `wal_keep_segments` parameter sufficiently high so that data is not lost before changes can be sent to the standby.

The following values are recommended for the `wal_keep_segments` parameter, which is defined in the `postgresql.conf` file:

- medium database: 32
- large database: 128

For more information about warm standby and hot standby with streaming replication, see <http://www.postgresql.org/docs/>.

8

Monitoring and Troubleshooting

<i>About Monitoring and Troubleshooting</i>	31
<i>Using SAS Environment Manager to Monitor and Troubleshoot Data</i>	31

About Monitoring and Troubleshooting

Monitoring data can be obtained from SAS Environment Manager. The data can then be evaluated for tuning purposes.

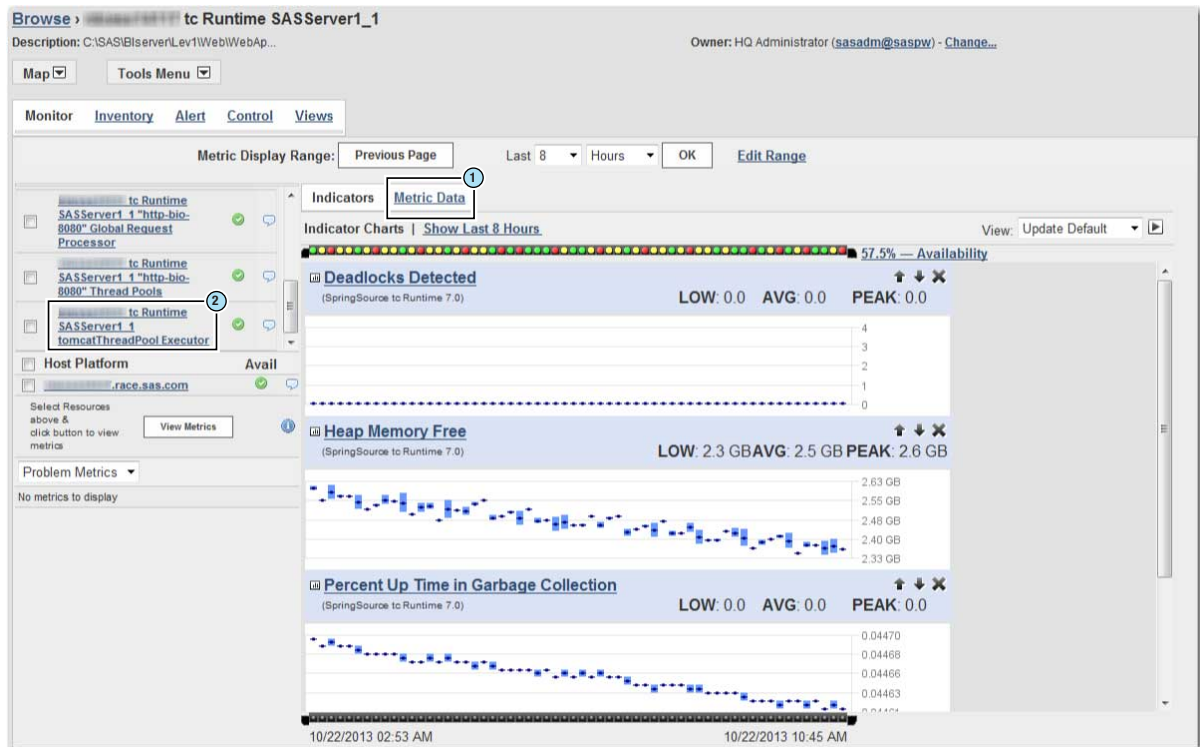
To monitor the metrics for a server, log on to SAS Environment Manager and click **Resources** ▶ **Browse** ▶ **Servers** . From the list of servers, select the SAS Web Application Server that you want to monitor. The server type should be displayed as **SpringSource tc Runtime 7.0**.

Using SAS Environment Manager to Monitor and Troubleshoot Data

The following display shows a sample of the window and monitoring data that is available. For each SAS Web Application Server instance, metrics can be

analyzed and used to tune the server instance. The metrics can be obtained by selecting the highlighted links:

Figure 8.1 SAS Environment Manager Resource Monitoring



- 1 Metric Data** shows current committed heap size, free heap, max heap, and heap in use.
- 2 server-name tc Runtime SAS Server1_n tomcatThreadPool Executor** shows current committed heap size, free heap, max heap, and heap in use.

For more information, see *SAS Environment Manager: User's Guide*.

Appendix 1

Initial Sizing and Tuning Configuration Options for Deployments

<i>About the Configuration Options</i>	33
<i>SAS Web Application Server Single Server Settings</i>	34
<i>SAS Web Application Server Multiple Server Settings</i>	35
SAS Web Application Server SASServer1 Settings	35
SAS Web Application Server SASServer2 Settings	37
SAS Web Application Server SASServer6 Settings	38
SAS Web Application Server SASServer7 Settings	41
SAS Web Application Server SASServer12 Settings	42
<i>SAS Web Server Settings</i>	44

About the Configuration Options

The following sections provide information about the initial tunable parameters for SAS Web Application Server and SAS Web Server, based on the deployment size. The parameters include Java Virtual Machine (JVM), database, and web server configuration options.

Note: The initial configuration options that are documented in this appendix are supported in single server mode and in multiple server mode for SASServers 1, 2, 6, 7, and 12. For the second maintenance release of SAS 9.4 and later, this tuning profile feature is available for SASServers 1, 2, 6, and 12. Beginning with the third maintenance release of SAS 9.4, tuning profile support is added for SASServer7. During the installation process, you can choose a small, medium, or large deployment. The installation process automatically configures the JVM settings shown in the following tables. For the first maintenance release of SAS 9.4 and earlier, you must manually update your configuration with these settings.

SAS Web Application Server Single Server Settings

The following table contains the SAS Web Application Server single server general concepts and settings. These initial configuration options are available for the second maintenance release of SAS 9.4 and later.

Table A1.1 Single Server Settings

Platform	Server Size	Option	Recommended Value
HP-UX, Linux, Solaris, and Windows	Small	-Xms	1024m
		-Xmx	3072m
		-XX:PermSize	768m
		-XX:MaxPermSize	1536m
	Medium	-Xms	1024m
		-Xmx	4096m
		-XX:PermSize	768m
		-XX:MaxPermSize	1536m
	Large	-Xms	8192m
		-Xmx	8192m
		-XX:PermSize	1664m
		-XX:MaxPermSize	1664m
AIX	Small	-Xms	1024m
		-Xmx	3072m
	Medium	-Xms	1024m
		-Xmx	4096m
	Large	-Xms	4096m
		-Xmx	8192m
Additional JVM Args (all platforms)	Small	not applicable	not applicable
	Medium	not applicable	not applicable
	Large	-Dsas.svcs.http.max.total.connections	512
-Dsas.svcs.http.max.connections		512	

Platform	Server Size	Option	Recommended Value
SharedServices JDBC Resource	Small	maxPoolSize	35
	Medium	maxPoolSize	100
	Large	maxPoolSize	256
Tomcat Executor Thread Pool	Small	maxThreads	150
	Medium	maxThreads	300
	Large	maxThreads	1024

SAS Web Application Server Multiple Server Settings

SAS Web Application Server SAServer1 Settings

The following table contains the SAS Web Application Server general concepts and settings for SAServer1. These initial configuration options are available for the second maintenance release of SAS 9.4 and later.

Table A1.2 SAServer1 Settings

Platform	Server Size	Option	Recommended Value
HP-UX, Linux, Solaris, and Windows	Small	-Xms	1024m
		-Xmx	3072m
		-XX:PermSize	384m
		-XX:MaxPermSize	1024m
	Medium	-Xms	1024m
		-Xmx	4096m
		-XX:PermSize	512m
		-XX:MaxPermSize	1024m
	Large	-Xms	4096m
		-Xmx	4096m
		-XX:PermSize	1024m
		-XX:MaxPermSize	1024m

Platform	Server Size	Option	Recommended Value
AIX	Small	-Xms	1024m
		-Xmx	3072m
	Medium	-Xms	1024m
		-Xmx	4096m
	Large	-Xms	2048m
		-Xmx	4096m
Additional JVM Args (all platforms)	Small	not applicable	not applicable
	Medium	not applicable	not applicable
	Large	-Dsas.svcs.http.max.total.connections	512
		-Dsas.svcs.http.max.connections	512
SharedServices JDBC Resource	Small	maxPoolSize	35
	Medium	maxPoolSize	100
	Large	maxPoolSize	256
Tomcat Executor Thread Pool	Small	maxThreads	150
	Medium	maxThreads	300
	Large	maxThreads	1024

SAS Web Application Server SASServer2 Settings

The following table contains the SAS Web Application Server general concepts and settings for SASServer2. These initial configuration options are available for the second maintenance release of SAS 9.4 and later.

Table A1.3 SASServer2 Settings

Platform	Server Size	Option	Recommended Value
HP-UX, Linux, Solaris, and Windows	Small	-Xms	768m
		-Xmx	1536m
		-XX:PermSize	384m
		-XX:MaxPermSize	768m
	Medium	-Xms	1024m
		-Xmx	2048m
		-XX:PermSize	512m
		-XX:MaxPermSize	768m
	Large	-Xms	2048m
		-Xmx	2048m
		-XX:PermSize	768m
		-XX:MaxPermSize	768m
AIX	Small	-Xms	768m
		-Xmx	1536m
	Medium	-Xms	1024m
		-Xmx	2048m
	Large	-Xms	1024m
		-Xmx	2048m
Additional JVM Args (all platforms)	Small	not applicable	not applicable
	Medium	not applicable	not applicable
	Large	-Dsas.svcs.http.max.total.connections	512
		-Dsas.svcs.http.max.connections	512
SharedServices JDBC Resource	Small	maxPoolSize	35
	Medium	maxPoolSize	35
	Large	maxPoolSize	100

Platform	Server Size	Option	Recommended Value
Tomcat Executor Thread Pool	Small	maxThreads	150
	Medium	maxThreads	300
	Large	maxThreads	1024

SAS Web Application Server SASServer6 Settings

Settings for the Second Maintenance Release

The following table contains the SAS Web Application Server general concepts and settings for SASServer6. These initial configuration options are available for the second maintenance release of SAS 9.4.

Table A1.4 SASServer6 Settings for the Second Maintenance Release

Platform	Server Size	Option	Recommended Value
HP-UX, Linux, Solaris, and Windows	Small	-Xms	768m
		-Xmx	1536m
		-XX:PermSize	384m
		-XX:MaxPermSize	768m
	Medium	-Xms	1024m
		-Xmx	2048m
		-XX:PermSize	512m
		-XX:MaxPermSize	768m
	Large	-Xms	2048m
		-Xmx	2048m
		-XX:PermSize	768m
		-XX:MaxPermSize	768m
AIX	Small	-Xms	768m
		-Xmx	1536m
	Medium	-Xms	1024m
		-Xmx	2048m
	Large	-Xms	1024m
		-Xmx	2048m

Platform	Server Size	Option	Recommended Value
Additional JVM Args (all platforms)	Small	not applicable	not applicable
	Medium	not applicable	not applicable
	Large	-Dsas.svcs.http.max.total.connections	512
		-Dsas.svcs.http.max.connections	512
SharedServices JDBC Resource	Small	maxPoolSize	35
	Medium	maxPoolSize	35
	Large	maxPoolSize	100
Tomcat Executor Thread Pool	Small	maxThreads	150
	Medium	maxThreads	300
	Large	maxThreads	1024

Settings for the Third Maintenance Release

The following table contains the SAS Web Application Server general concepts and settings for SAS Server6. These initial configuration options are available for the third maintenance release of SAS 9.4 and later.

Table A1.5 SAS Server6 Settings for the Third Maintenance Release

Platform	Server Size	Option	Recommended Value
HP-UX, Linux, Solaris, and Windows	Small	-Xms	768m
		-Xmx	1536m
		-XX:PermSize	384m
		-XX:MaxPermSize	768m
	Medium	-Xms	2048m
		-Xmx	4096m
		-XX:PermSize	512m
		-XX:MaxPermSize	768m
	Large	-Xms	8192m
		-Xmx	8192m
		-XX:PermSize	768m
		-XX:MaxPermSize	768m

Platform	Server Size	Option	Recommended Value
AIX	Small	-Xms	768m
		-Xmx	1536m
	Medium	-Xms	1024m
		-Xmx	2048m
	Large	-Xms	1024m
		-Xmx	2048m
Additional JVM Args (all platforms)	Small	not applicable	not applicable
	Medium	not applicable	not applicable
	Large	-Dsas.svcs.http.max.total.connections	512
		-Dsas.svcs.http.max.connections	512
SharedServices JDBC Resource	Small	maxPoolSize	35
	Medium	maxPoolSize	35
	Large	maxPoolSize	100
Tomcat Executor Thread Pool	Small	maxThreads	150
	Medium	maxThreads	300
	Large	maxThreads	1024

SAS Web Application Server SASServer7 Settings

The following table contains the SAS Web Application Server general concepts and settings for SASServer7. These initial configuration options are available for the third maintenance release of SAS 9.4 and later.

Table A1.6 SASServer7 Settings

Platform	Server Size	Option	Recommended Value
HP-UX, Linux, Solaris, and Windows	Small	-Xms	768m
		-Xmx	1536m
		-XX:PermSize	384m
		-XX:MaxPermSize	768m
	Medium	-Xms	1024m
		-Xmx	2048m
		-XX:PermSize	512m
		-XX:MaxPermSize	768m
	Large	-Xms	4096m
		-Xmx	4096m
		-XX:PermSize	768m
		-XX:MaxPermSize	768m
AIX	Small	-Xms	768m
		-Xmx	1536m
	Medium	-Xms	1024m
		-Xmx	2048m
	Large	-Xms	1024m
		-Xmx	2048m
Additional JVM Args (all platforms)	Small	not applicable	not applicable
	Medium	not applicable	not applicable
	Large	-Dsas.svcs.http.max.total.connections	512
		-Dsas.svcs.http.max.connections	512
SharedServices JDBC Resource	Small	maxPoolSize	35
	Medium	maxPoolSize	35
	Large	maxPoolSize	100

Platform	Server Size	Option	Recommended Value
Tomcat Executor Thread Pool	Small	maxThreads	150
	Medium	maxThreads	300
	Large	maxThreads	1024

SAS Web Application Server SASServer12 Settings

The following table contains the SAS Web Application Server general concepts and settings for SASServer12. These initial configuration options are available for the second maintenance release of SAS 9.4 and later.

Table A1.7 SASServer12 Settings

Platform	Server Size	Option	Recommended Value
HP-UX, Linux, Solaris, and Windows	Small	-Xms	1024m
		-Xmx	1536m
		-XX:PermSize	384m
		-XX:MaxPermSize	768m
	Medium	-Xms	1024m
		-Xmx	2048m
		-XX:PermSize	512m
		-XX:MaxPermSize	768m
	Large	-Xms	8192m
		-Xmx	8192m
		-XX:PermSize	768m
		-XX:MaxPermSize	768m
AIX	Small	-Xms	1024m
		-Xmx	1536m
	Medium	-Xms	1024m
		-Xmx	2048m
	Large	-Xms	4096m
		-Xmx	8192m

Platform	Server Size	Option	Recommended Value
Additional JVM Args (all platforms)	Small	not applicable	not applicable
	Medium	not applicable	not applicable
	Large	-Dsas.svcs.http.max.total.connections	512
		-Dsas.svcs.http.max.connections	512
SharedServices JDBC Resource	Small	maxPoolSize	35
	Medium	maxPoolSize	35
	Large	maxPoolSize	100
Tomcat Executor Thread Pool	Small	maxThreads	150
	Medium	maxThreads	300
	Large	maxThreads	1024

SAS Web Server Settings

The following table contains the SAS Web Server settings. These initial configuration options are available for the second maintenance release of SAS 9.4 and later.

Table A1.8 SAS Web Server Settings

Platform	Server Size	Option	Recommended Value	Filename
Windows (mpm_winnt_module)	Small	ThreadsPerChild	512	conf\extra\httpd-mpm.conf
	Medium	ThreadsPerChild	1024	conf\extra\httpd-mpm.conf
	Large	ThreadLimit	3072	conf\extra\httpd-mpm.conf
		ThreadsPerChild	3072	conf\extra\httpd-mpm.conf
		MaxKeepAliveRequests	0	conf\extra\httpd-default.conf

Platform	Server Size	Option	Recommended Value	Filename
UNIX and Linux (mpm_worker_module)	Small	ServerLimit	16	conf/extra/httpd-mpm.conf
		MaxClients	512	conf/extra/httpd-mpm.conf
		ThreadsPerChild	32	conf/extra/httpd-mpm.conf
	Medium	ServerLimit	32	conf/extra/httpd-mpm.conf
		MaxClients	1024	conf/extra/httpd-mpm.conf
		ThreadsPerChild	32	conf/extra/httpd-mpm.conf
	Large	ServerLimit	96	conf/extra/httpd-mpm.conf
		MaxClients	3072	conf/extra/httpd-mpm.conf
		ThreadsPerChild	32	conf/extra/httpd-mpm.conf
		MaxKeepAliveRequests	0	conf/extra/httpd-default.conf

Recommended Reading

Here is the recommended reading list for this document:

- *SAS Intelligence Platform: Overview*
- *SAS Intelligence Platform: System Administration Guide*
- *SAS Intelligence Platform: Security Administration Guide*
- *SAS Intelligence Platform: Middle-Tier Administration Guide*
- SAS offers instructor-led training and self-paced e-learning courses to help you administer the SAS Intelligence Platform. For more information about the courses available, see support.sas.com/admintraining.

For a complete list of SAS publications, go to sas.com/store/books. If you have questions about which titles you need, please contact a SAS Representative:

SAS Books
SAS Campus Drive
Cary, NC 27513-2414
Phone: 1-800-727-0025
Fax: 1-919-677-4444
Email: sasbook@sas.com
Web address: sas.com/store/books

Glossary

cluster

See [computer cluster](#).

computer cluster (cluster)

a set of connected nodes (computers that are used as servers) in a centralized, cohesive system that shares computing tasks across the system for fast, reliable processing. A computer cluster can be as simple as two machines connected in a network, but more often refers to a large network of computers. A cluster can be established to achieve higher levels of performance and load distribution, or to increase reliability through redundancy.

deploy

to install an instance of operational SAS software and related components. The deployment process often includes configuration and testing as well.

performance tuning (tuning)

one or more modifications to a computer system that enables the system to handle a higher processing load. Performance tuning generally follows a series of steps to analyze system behavior, modify areas that cause bottlenecks, and test for improvement.

pool

a group of server connections that can be shared and reused by multiple client applications. A client-side pool consists of one or more puddles. See *also* [puddle](#).

portal

a web application that enables users to access websites, data, documents, applications, and other digital content from a single, easily accessible user interface. A portal's personalization features enable each user to configure and organize the interface to meet individual or role-based needs. See *also* [portlet](#).

portlet

a web component that is managed by a web application and that is aggregated with other portlets to form a page within the application. Portlets can process requests from the user and generate dynamic content.

puddle

a group of servers that are started and run using the same login credentials. Each puddle can also allow a group of clients to access the servers. See *also* [pool](#).

SAS Application Server

a logical entity that represents the SAS server tier, which in turn comprises servers that execute code for particular tasks and metadata objects.

SAS Management Console

a Java application that provides a single user interface for performing SAS administrative tasks.

SAS Metadata Repository

a container for metadata that is managed by the SAS Metadata Server.

SAS Web Infrastructure Platform

a collection of middle-tier services and applications that provide infrastructure and integration features that are shared by SAS web applications and other HTTP clients.

SAS Workspace Server

a SAS server that provides access to SAS Foundation features such as the SAS programming language and SAS libraries.

service

one or more application components that an authorized user or application can call at any time to provide results that conform to a published specification. For example, network services transmit data or provide conversion of data in a network, database services provide for the storage and retrieval of data in a database, and web services interact with each other on the World Wide Web.

tuning

See [performance tuning](#).

Index

A

AIX tuning 15

C

cluster of web application servers 3
 clustering
 horizontal 5
 vertical 4
 configuring
 standby database performance 30
 configuring standby database
 performance 30

H

horizontal clustering 5
 HP-UX tuning 16

I

initial configuration options 33
 SAS Web Application Server 34
 SAS Web Server 44

J

Java Virtual Machine (JVM) 11
 option descriptions 11
 SAS Web Application options 13
 SAS Web Application Server options
 13
 tuning 11
 tuning guidelines 14
 Java Virtual Machine (JVM) tuning
 options 12
 JVM
 See [Java Virtual Machine \(JVM\)](#)

L

Linux tuning 17

M

monitoring
 using SAS Environment Manager
 31
 monitoring performance
 determining last vacuum 29
 displaying cache hit ratio 28
 executing a server query 28
 finding missing indexes 28
 identifying running queries 29
 logging all statements 28
 resetting data 28
 seeing table indexes 29

P

performance
 clustering web application servers 3
 postgresql.conf file
 initial configuration values 24

S

SAS Visual Analytics Transport
 Service
 tuning 9
 SAS web application
 tuning 9
 SAS Web Application JVM options 13
 SAS Web Application Server JVM
 options 13
 SAS Web Application Server tuning 7
 SAS web application tuning 9
 SAS Web Infrastructure Platform Data
 Server
 postgresql.conf file 24
 tuning recommendations 22
 tuning steps 22

Solaris tuning 18

T

troubleshooting

using SAS Environment Manager 31

tuning

AIX 15

HP-UX 16

Java Virtual Machine (JVM) 11

Linux 17

SAS Visual Analytics Transport Service 9

SAS Web Application Server 7

SAS web applications 9

Solaris 18

Windows 18

tuning guidelines

AIX 14

HP-UX 14

Linux 14

Solaris 14

Windows 14

tuning Java Virtual Machine (JVM) guidelines 14

recommended values 12

SAS Web Application Server 13

SAS Web Applications 13

tuning recommendations

SAS Web Application Server 7

SAS Web Infrastructure Platform Data Server 22

tuning steps

SAS Web Infrastructure Platform Data Server 22

V

vertical clustering 4

W

Web Application Server

tuning 7

Windows

system configurations 19

tuning 18

Windows system configurations 19

Write-Ahead Log (WAL) 27



Gain Greater Insight into Your SAS[®] Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.

 support.sas.com/bookstore
for additional books and resources.


THE POWER TO KNOW.

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. © 2013 SAS Institute Inc. All rights reserved. S107969US.0613

