

Configuring IBM WebSphere Application Server 6.1 to Support SAS 9.2 Web Applications

This document is for SAS installers who want to configure IBM WebSphere Application Server for use with SAS instead of using IBM WebSphere Application Server Network Deployment. When you run SAS Deployment Wizard to install and configure SAS 9.2 Enterprise Business Intelligence software, you have the option to allow the deployment wizard to automatically configure a WebSphere Application Server Network Deployment application server for you and to deploy SAS Web applications to that server. You can also use WebSphere Application Server to run SAS Web applications in SAS 9.2, but the deployment wizard cannot perform the automatic configuration that it can for WebSphere Application Server Network Deployment. For WebSphere Application Server, you must perform the configuration manually.

The deployment wizard prompts for the installation path to WebSphere Application Server Network Deployment. The path to the WebSphere Application Server Network Deployment or WebSphere Application Server product must be provided. However, because the deployment wizard validates a WebSphere Application Server Network Deployment installation only for automatic configuration, a change must be made to a text file under the WebSphere Application Server installation to allow the deployment wizard to continue when the installation location contains WebSphere Application Server instead. You must obtain the `WAS.product` file from SAS Technical Support and replace the corresponding file in the WebSphere Application Server installation location after backing up the original copy of the `WAS.product` file. This change permits the deployment wizard to continue executing, ultimately creating the SAS EAR files that must be deployed manually for the SAS Web applications. After the deployment wizard runs, configure WebSphere Application Server and deploy the SAS Web applications as described in this document.

From a security standpoint, when you follow these instructions, the SAS Web applications perform SAS authentication by default. SAS authentication in most environments is set to perform authentication against the host operating system. So, after you complete the configuration steps in this document, consider enabling Web authentication as described in *Configuring IBM WebSphere Application Server 6.1 for Web Authentication with SAS 9.2 Web Applications*.

Note: The example options and code shown throughout this document are for Windows. However, you can follow the same manual configuration steps on other platforms that SAS supports.

Prerequisites

You must have installed WebSphere Application Server, Version 6.1. An application server named `server1` was configured for you when WebSphere Application Server was installed. This is the application server where you will run SAS Web applications.

Substitute the WAS.product File and Run the SAS Deployment Wizard

Before running the SAS Deployment Wizard, the existing WebSphere Application Server `WAS.product` file must be backed up and then replaced with a substitute file that is provided by SAS Technical Support. It is very important that this change is reversed at the end of this procedure. To get the substitute file and apply it, perform the following steps:

1. Navigate to the IBM WebSphere Application Server home directory (`WAS_INSTALL_ROOT`) and copy the entire IBM WebSphere Application Server home directory to a backup directory.
2. Navigate to the `WAS_INSTALL_ROOT\properties\version` directory. The default locations are shown in the following examples:

```
C:\Program Files\IBM\WebSphere\AppServer\properties\version (Windows)
/usr/IBM/WebSphere/Appserver/properties/version (UNIX)
```

3. Locate the `WAS.product` file.
4. Rename the existing `WAS.product` file to `WAS.product.original`. This is a temporary step that you must immediately reverse after SAS Deployment Wizard finishes.
5. Proceed to the SAS FTP server, `ftp://ftp.sas.com/techsup/download/web`.
6. Copy the `WAS.product` file from the SAS FTP server to your `WAS_INSTALL_ROOT\properties\version` directory. This substitute `WAS.product` file is temporary and is used only until the SAS Deployment Wizard finishes.
7. Proceed with the SAS Deployment Wizard.

Important: When the SAS Deployment Wizard finishes, perform the following steps:

8. Navigate to the `WAS_INSTALL_ROOT\properties\version` directory.
9. Remove the substitute `WAS.product` file that was copied from the SAS FTP server.
10. Rename `WAS.product.original` back to `WAS.product`.

Configure WebSphere Application Server and Deploy SAS Web Applications

Follow the directions in this section to configure WebSphere Application Server and deploy your SAS Web applications.

Note: As you perform tasks for using the WebSphere Integrated Solutions Console (known as the *administrative console*), this document does not specify when you need to save your changes. Generally, you should save changes each time that you finish making changes on a page. This document also does not specify when you should perform **System administration > Save Changes to the Master Repository**. You should do this periodically and, in particular, just before you stop server1.

When you ran the SAS Deployment Wizard, it created a set of EAR files that contain the SAS Web applications. The SAS Deployment Wizard builds these EAR files and stores them in the `SAS-config-dir\Lev1\Web\Staging` directory.

Customize server1

To configure a WebSphere Application Server to host the SAS Web applications, perform the following steps:

1. Make sure that port numbers for server1 are set correctly:
 - a. Select **Servers > Application Servers > server1**, and expand the plus sign next to the **Ports** link. A list of the ports that are defined for this server displays.
 - b. Compare the value of **WC_defaulthost** with the port number you provided in SAS Deployment Wizard when you were prompted for a WebSphere listen port
 - c. If the port number values are not the same, click the **Ports** link and then the **WC_defaulthost** link. On the **General Properties** page, change the value of **Port** to match the value that you entered in SAS Deployment Wizard.
 - d. Make sure that your **WC_defaulthost** port value matches the value that you provided for the **default_host** virtual host, which you can find at **Environment > Virtual Hosts > default_host > Host Aliases** in the administrative console.

Note: In the future, you must also check the **BOOTSTRAP_ADDRESS** (RMI) port.

2. Set the JVM options for server1 to those that the SAS Web applications require:
 - a. Select **Servers > Application Servers > server1 > Java and Process Management > Process Definition > Java Virtual Machine**.
 - b. Set the **Initial Heap Size** and **Maximum Heap Size** to values, as specified in [Appendix A – JVM Options for Supported Platforms](#). Enter the **-Xms** value in the **Initial Heap Size** field and the **-Xmx** value in the **Maximum Heap Size** field. For example, for a 32-bit Windows system, enter a value of 768 in both fields.
 - c. In the **Generic JVM arguments** field, enter a list of JVM options and separated them with spaces. First, consult Appendix A, and then enter the options that are listed for your operating system and processor architecture all on one line. You must substitute values for these options:

```
-Dmulticast.address=IP-address  
-Dmulticast.port=port-number  
-Dcom.sas.log.config.url=URL-for-log-file
```

The *URL-for-log-file* value can point to any location. The SAS Deployment Wizard convention is to use a file URL similar to the following example:

```
"file:///C:/SAS/Config/Lev1/Web/Common/LogConfig/"
```

3. Specify two custom properties for the web container:
 - a. Select **Servers > Application servers > server1 > Web Container Settings > Web container > Custom Properties**.
 - b. Click **New**.
 - c. On the **General Properties** page, enter these values for the Name and Value and click **OK**:
Name: `com.ibm.ws.webcontainer.channelwritetype`
Value: `sync`

- d. Define a second customer property using these values:
Name: `prependSlashToResource`
Value: `true`
4. Create a service integration bus:
 - a. Select **Service Integration > Buses**.
 - b. Click **New**.
 - c. On the **Create a new bus** page, set the name of the bus to `SAS Messaging Bus` and make sure that bus security is not enabled. Click **Next**.
 - d. On the **Confirm create of new bus** page, click **Finish**.
5. Add server1 as a member of the SAS Messaging Bus:
 - a. Select **Service Integration > Buses > SAS Messaging Bus > Bus Members**.
 - b. Click **Add**.
 - c. On the **Select server, cluster or WebSphere MQ server** page, select the **Server** radio button, select `sdw01Node01:server1` (or the equivalent) from the **Server** list and click **Next**.
 - d. On the **Select the type of message store and Provide the message store properties** pages, click **Next**.
 - e. Click **Finish**.

Configure JMS Resources

To configure the JMS resources that are needed by the SAS Web applications, perform the following steps:

1. Create a JMS queue connection factory:
 - a. Select **Resources > JMS > Queue connection factories**.
 - b. On the **Queue connection factories** page, select server scope from the list (such as `Node=sdw01Node01, Server=server1`) and click **New**.
 - c. On the **Select JMS resource provider** page, select **Default messaging provider** and click **OK**.
 - d. On the **New** page, provide this information and click **OK**.
Name: `SASQueueConnectionFactory`
JNDI Name: `sas/jms/QueueConnectionFactory`
Bus name: `SAS Messaging Bus`
 - e. Click **Apply**.
2. Create a SAS Alert Queue:
 - a. Select **Resources > JMS > Queues**.
 - b. On the **Queues** page, select server scope from the list and click **New**.
 - c. On the **Select JMS resource provider** page, select **Default messaging provider** and click **OK**.

- d. On the **New** page, provide the following information and click **OK**:
 - Name: SAS Alert Queue
 - JNDI Name: sas/jms/AlertQueue
 - Bus name: SAS Messaging Bus
 - Queue name: other, please specify. When you choose this value, a dialog box displays. Enter SAS Alert Queue in the dialog box.
- e. Click **Apply**.
3. Create a SAS workflow queue by using the same procedure that you used to create the alert queue but substitute the string “Workflow” for “Alert” everywhere. You should now have a queue named SAS Workflow Queue, and it should have a JNDI named sas/jms/WorkflowQueue.
4. Create a bus destination for the SAS Alert Queue:
 - a. Select **Service integration > Buses > SAS Messaging Bus > Destinations**.
 - b. Click **New**.
 - c. On the **Create new destination** page, select **Queue** and click **Next**.
 - d. On the **Set Queue attributes** page, enter SAS Alert Queue in the **Identifier** field and click **Next**.
 - e. On the **Assign the queue to a bus member** page, select the Node=sdw01Node01:Server=server1 entry from the **Bus member** list and click **Next**. (Your node name can be different.)
 - f. Click **Finish**.
5. Create a bus destination for the SAS Workflow Queue by following the same procedure that you used to create the SAS Alert Queue destination but change the identifier to SAS Workflow Queue.

Configure a Mail Session and a Mail Provider

To configure the mail resources needed by the SAS Web applications, perform the following steps:

1. Select **Resources > Mail > Mail sessions**.
2. On the **Mail Sessions** page, select server scope from the list and click **New**.
3. On the **New** page, enter the following information and click **OK**.
 - Name: SAS Mail Session
 - JNDI Name: sas/mail/Session
 - Mail transport host: smtp.example.com (the fully qualified name of your mail-server host)
 - Mail transport protocol: smtp
4. Select **Resources > Mail > Mail Providers**.
5. Click the **Built-in Mail Provider** link that is associated with server scope.
6. On the **General Properties** page, click **Protocol providers**.
7. On the **Protocol providers** page, click **smtp**.
8. On the **smtp** page, enter the following text in the **Class path** text area and click **OK**:


```

      ${WAS_INSTALL_ROOT}\\lib\\mail-impl.jar
      
```

Configure a JDBC DataSource

To configure the JDBC resource needed by the SAS Web applications, perform the following steps:

Note: The JAR files that are used in this step might not be part of your SAS installation, depending on the SAS products that are specified in your SAS plan file.

1. Select **Security > Secure administration, applications, and infrastructure > Java Authentication and Authorization Service > J2C authentication data**.
2. Click **New**.
3. On the **New** page, enter the following values and click **OK**.

Alias: SASJAAS

User ID: sastrust@saspw

Password: *unencoded-password* (for sastrust)

4. Select **Resources > JDBC > JDBC Providers**.
5. On the **JDBC providers** page, select **Server scope** and click **New**.
6. On the **Create new JDBC provider** page, enter the following values and click **Next**.

Database type: User-defined

Implementation class name: com.sas.tkts.TKTSConnectionPoolDataSource

Name: SAS TableServer JDBC Provider

7. On the **Enter database class path information** page, enter the locations of the following JAR files, delimiting the JAR files with new lines, and click **Next**.

```
C:\\SAS\\config\\Lev1\\Web\\Common\\TableServerDriver\\icu4j.jar
C:\\SAS\\config\\Lev1\\Web\\Common\\TableServerDriver\\log4j.jar
C:\\SAS\\config\\Lev1\\Web\\Common\\TableServerDriver\\sas.core.jar
C:\\SAS\\config\\Lev1\\Web\\Common\\TableServerDriver\\sas.core.nls.jar
C:\\SAS\\config\\Lev1\\Web\\Common\\TableServerDriver\\sas.icons.jar
C:\\SAS\\config\\Lev1\\Web\\Common\\TableServerDriver\\sas.icons.nls.jar
C:\\SAS\\config\\Lev1\\Web\\Common\\TableServerDriver\\sas.intrnet.javatools.jar
C:\\SAS\\config\\Lev1\\Web\\Common\\TableServerDriver\\sas.intrnet.javatools.nls.jar
C:\\SAS\\config\\Lev1\\Web\\Common\\TableServerDriver\\sas.nls.collator.jar
C:\\SAS\\config\\Lev1\\Web\\Common\\TableServerDriver\\sas.oda.tkts.jar
C:\\SAS\\config\\Lev1\\Web\\Common\\TableServerDriver\\sas.oda.tkts.nls.jar
C:\\SAS\\Config\\Lev1\\Web\\Common\\TableServerDriver\\sas.rutil.jar
C:\\SAS\\Config\\Lev1\\Web\\Common\\TableServerDriver\\sas.rutil.nls.jar
C:\\SAS\\Config\\Lev1\\Web\\Common\\TableServerDriver\\sas.security.sspi.jar
C:\\SAS\\config\\Lev1\\Web\\Common\\TableServerDriver\\sas.svc.connection.jar
C:\\SAS\\config\\Lev1\\Web\\Common\\TableServerDriver\\sas.svc.connection.nls.jar
```

Note: Your SAS configuration directory might differ from the previous example.

8. From the JDBC providers page, click the SAS TableServer JDBC Provider link.
9. On the SAS TableServer JDBC Provider page, click the Data sources link.
10. On the **Data sources** page, click **New**.
11. On the **Enter basic data source information** page, enter these values and click **Next**.

Data source name: SharedServices

JNDI name: sas/jdbc/SharedServices

Component-managed authentication alias and XA recovery authentication alias:

sdw01Node01/SASJAAS (or equivalent)

12. On the **Enter database specific properties for the data source** page, deselect the **Use this data source in container managed persistence (CMP)** check box, and click **Next**.
13. Click **Finish**.
14. From the **Data Sources** page, click the **SharedServices** link.
15. On the **General Properties** page, click **Custom Properties**. You must assign values to some properties. To do this, click on the name of the property, supply a value, and click **OK**. You can delete any property that you do not need by selecting the check box next to the property name and clicking **Delete**.
16. Set the following properties:


```
password: unencoded-password (for sastrust)
serverUrl: jdbc:sastkts://table-server-host:2171
user: sastrust@saspw
dataSourceName: SharedServices
serverName: table-server-host
portNumber: 2171
stmtPooling: 0
constring: (DSN=SharedServices)
```
17. Replace *table-server-host* with the fully qualified domain name of the host on which SAS Table Server is running. Also, if you changed the port number for SAS Table Server, replace the default value of 2171 with the actual port number. To determine the port number that was specified when SAS Deployment Wizard was run, open the `configuration.properties` file and search for the `iomsrv.tableserver.port` property.

Configure JAAS Login Modules

To configure the JAAS Login Modules needed by the SAS Web applications, perform the following steps:

1. Select **Security > Secure, administration, applications, and infrastructure > Java Authentication and Authorization Service > Application logins**.
2. On the **Application logins** page, click **New**.
3. On the **New** page, enter the value **PFS** in the **Alias** field and click **OK**. You will assign two JAAS login modules to this alias.
4. From the **Application logins** page, click **PFS**.
5. On the **PFS** page, click **JAAS login modules**.
6. On the **JAAS login modules** page, click **New**.
7. On the **New** page, enter these values and click **OK**.


```
Module class name: com.sas.services.security.login.OMILoginModule
Authentication strategy: SUFFICIENT
```
8. From the **JAAS login modules** page, click **com.sas.services.security.login.OMILoginModule**.
9. On the **com.sas.services.security.login.OMILoginModule** page, click **Custom properties**.

10. On the **Custom properties** page, for each of these name-value pairs, click **New**, provide the following name-value pairs, and click **OK**.

```
debug: false
domain: DefaultAuth
host: metadata-server-host
port: 8561 (or nondefault port)
repository: Foundation
trustedpw: encoded-password (for sastrust)
trusteduser: sastrust@saspw
```

To determine the encoded password for a password, you can use this short SAS program:

```
PROC PWENCODE IN='password' ;
RUN;
```

11. Add a second JAAS login module to the PFS application login. Create the module with the `com.sas.services.security.login.TrustedLoginModule` class name and the `OPTIONAL` authentication strategy. Set the following custom properties:

```
aliasdomain: MidtierInternal
debug: false
domain: DefaultAuth
host: metadata-server-host
port: 8561
repository: Foundation
trustedpw: encoded-password (for sastrust)
trusteduser: sastrust@saspw
```

12. Define a second JAAS alias, which has only one JAAS LoginModule assigned to it. Use `SCS` as the alias. To this alias, add the login module `com.sas.services.security.login.OMILoginModule`. Set the authentication strategy to `REQUIRED` and then set the following custom properties:

```
debug: false
domain: DefaultAuth
holdopenconnection: true
host: metadata-server-host
port: 8561 (or nondefault port)
repository: Foundation
trustedpw: encoded-password (for sastrust)
trusteduser: sastrust@saspw
```

13. Define a third JAAS alias, which has only one JAAS LoginModule assigned to it. Use the alias `UsernamePassword`. To this alias, add the login module `com.platform.SASLogin.UsernamePasswordLogin`. Set the authentication strategy to `REQUIRED`. This login module requires one custom property, `debug=false`.

Restart the WebSphere Application Server

This is the end of the configuration process. At this point, before you begin to deploy SAS Web applications, you should use **System administration > Save Changes to Master Repository** to save your changes. To make sure that the configuration has no errors, you can restart server1 by opening a command line window and changing the working directory to the `WAS_INSTALL_ROOT\AppServer\bin` directory and then run the following commands:

```
stopServer server1
startServer server1
```

Deploy SAS Web Applications

Install all EAR files that are provided in the `SAS-config-dir\Lev1\Web\Staging` directory. To install an EAR file, perform the following steps:

Note: Do not start the applications as you install them. Information about starting the applications is provided later in this document.

1. Select **Applications > Enterprise Applications**.
2. Click **Install**.
3. On the **Specify the EAR, WAR, JAR, or SAR module to upload and install** page, select the **Local file system** or **Remote file system** radio button, and then browse to the location of the EAR file. Select the EAR file and click **OK**. Click **Next**.
4. Finish running the installation wizard by accepting all defaults.
5. From the **Enterprise Applications** page, select the newly installed application.
6. On the page for that application, set the class-loading behavior for the EAR file:
 - a. Click **Class loading and update detection**.
 - b. On the **Class loader** page, set **Class loader order** to **Classes loaded with application class loader first**. (Leave the WAR class loader policy set to **Class loader for each WAR file in application**.)
Note: To work around a defect in the administration console, set the polling interval to zero (0) before you click **OK**.
7. For each WAR file in the EAR file, set the class-loader behavior:
 - a. On the main page for configuring the application (EAR), select **Manage Modules**.
 - b. Click the name of the Web module (WAR file) that you want to configure. (Some EAR files have only one module.)
 - c. On the configuration page for the WAR file, change the value of **Class loader order** to **Classes loaded with application class loader first** and click **OK**.

Start the SAS Web Applications

Once you deploy the applications, start them one at a time so that you can monitor the log for each application in isolation. Start the applications in the following order:

1. SAS Themes
2. SAS Web Infrastructure Platform Services
3. SAS Web Infrastructure Platform Applications
4. SAS Content Server

5. SAS Information Delivery Portal 4.2
6. Other applications

To prepare SAS Web applications to start in the correct order after a server restart, set the **Startup order** for each application. Use the following values:

SAS Themes (1)

SAS Web Infrastructure Platform Services (2)

SAS Web Infrastructure Platform Applications (3)

SAS Content Server (4)

SAS Information Deliver Portal 4.2 (5)

Other applications (value greater than 5)

Note: Restart WebSphere Application Server so that SAS Web applications display the SAS Logon Manager application log on Web page. The default authentication mechanism is SAS authentication. For information about configuring Web authentication, see *Configuring IBM WebSphere Application Server 6.1 for Web Authentication with SAS 9.2 Web Applications*.

Appendix A: JVM Options for Supported Platforms

The following sections are organized by operating system and processor architecture. Each section shows the optimally tuned Java options for WebSphere Application Server on a machine with a minimum of 8GB of memory.

Linux on Intel (32 or 64-bit)

```
-Xms1024m -Xmx1024m -Xss256k -Xmso256k -Xgcpolicy:optavgpause
-Djava.net.preferIPv4Stack=true -Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true
-Dsas.container.identifier=websphere -Dsas.auto.publish.port=9080
-Dcom.sas.services.logging.disableRemoteList=true
-Dcom.sas.services.logging.disableRemoteLogging=true -Dmulticast.address=
IP-address -Dmulticast.port=port-number -Dcom.sas.log.config.url=URL-for-log-
file
```

AIX (64-bit)

```
-Xms1024m -Xmx2048m -Xss256k -Xmso256k -Xgcpolicy:optavgpause
-Djava.net.preferIPv4Stack=true -Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true
-Dsas.container.identifier=websphere -Dsas.auto.publish.port=9080
-Dcom.sas.services.logging.disableRemoteList=true
-Dcom.sas.services.logging.disableRemoteLogging=true -Dmulticast.address=
IP-address -Dmulticast.port=port-number -Dcom.sas.log.config.url=URL-for-log-
file
```

Solaris on Intel (64-bit)

```
-Xms1024m -Xmx2048m -Xss256k -d64 -XX:NewSize=128m -XX:MaxNewSize=256m
-XX:+UseTLAB -XX:+UseConcMarkSweepGC -XX:+CMSIncrementalMode
-XX:+DisableExplicitGC -XX:PermSize=512m -XX:MaxPermSize=512m
-Dcom.sun.management.jmxremote -Djava.net.preferIPv4Stack=true
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true
-Dsas.container.identifiier=websphere -Dsas.auto.publish.port=9080
-Dcom.sas.services.logging.disableRemoteList=true
-Dcom.sas.services.logging.disableRemoteLogging=true -Dmulticast.address=
IP-address -Dmulticast.port=port-number -Dcom.sas.log.config.url=URL-for-log-
file
```

Solaris on SPARC (64-bit)

```
-Xms1024m -Xmx2048m -Xss256k -d64 -XX:NewSize=128m -XX:MaxNewSize=256m
-XX:+UseTLAB -XX:+UseConcMarkSweepGC -XX:+CMSIncrementalMode
-XX:+DisableExplicitGC -XX:PermSize=512m -XX:MaxPermSize=512m
-Dcom.sun.management.jmxremote -Djava.net.preferIPv4Stack=true
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true
-Dsas.container.identifiier=websphere -Dsas.auto.publish.port=9080
-Dcom.sas.services.logging.disableRemoteList=true
-Dcom.sas.services.logging.disableRemoteLogging=true -Dmulticast.address=
IP-address -Dmulticast.port=port-number -Dcom.sas.log.config.url=URL-for-log-
file
```

Windows (32-bit)

```
-Xms768m -Xmx768m -Xss160k -Xms0160k -Xgcpolicy:optavgpause
-Djava.net.preferIPv4Stack=true -Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true
-Dsas.container.identifiier=websphere -Dsas.auto.publish.port=9080
-Dcom.sas.services.logging.disableRemoteList=true
-Dcom.sas.services.logging.disableRemoteLogging=true -Dmulticast.address=
IP-address -Dmulticast.port=port-number -Dcom.sas.log.config.url=URL-for-log-
file
```

Windows (64-bit)

```
-Xms640m -Xmx1920m -Xss256k -Xms0256k -Xgcpolicy:gencon -Xdisableexplicitgc
-Djava.net.preferIPv4Stack=true -Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true
-Dsas.container.identifiier=websphere -Dsas.auto.publish.port=9080
-Dcom.sas.services.logging.disableRemoteList=true
-Dcom.sas.services.logging.disableRemoteLogging=true -Dmulticast.address=
IP-address -Dmulticast.port=port-number -Dcom.sas.log.config.url=URL-for-log-
file
```

SAS and all other SAS Institute product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. Other brand and product names are registered trademarks or trademarks of their respective companies.

® indicates USA registration.

Copyright © 2009 SAS Institute Inc., Cary, NC, USA. All rights reserved.