

Administrator's Guide for SAS[®] Analytics Platform 1.5



Copyright Notice

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *Administrator's Guide for SAS® Analytics Platform 1.5*, Cary, NC: SAS Institute Inc., 2009.

Administrator's Guide for SAS® Analytics Platform 1.5

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

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

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

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

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

® indicates USA registration.

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

Table of Contents

Introduction	1
Migration	3
Configuration	5
Ports	6
Embedded Tomcat HTTP Server.....	6
SAS Foundation Services	7
RMI Services	7
Multicast Discovery Servers	10
Startup Options	10
Windows Service	11
File System Layout	11
Unconfiguring the SAS Analytics Platform Server using the SAS Deployment Manager	12
Securing RMI Services Using JSSE	13
Configuration Options	13
Create Keystore Certificate	14
List Certificates	15
Delete Certificate	16
Using a Self-Signed Certificate	17
Export Certificate from Keystore	17
Import Keystore's Certificate into Truststore	18
Using a Certificate Signed by a Certificate Authority	20
Import CA's Primary Certificate into JRE/lib/security/cacerts	20
Import CA's Intermediate Certificate into JRE/lib/security/cacerts.....	22
Generate Certificate Signature Request (CSR)	23
Submit CSR to Certificate Authority (CA)	24
Import Certificate Authority's Reply into Keystore	24
Runtime	27
Starting the SAS Analytics Platform Server	27
Starting the SAS Analytics Platform Server under Windows	27
Starting the SAS Analytics Platform Server under UNIX.....	27
Stopping the SAS Analytics Platform Server	28
Stopping the SAS Analytics Platform under Windows	28
Stopping the SAS Analytics Platform under UNIX	28
SAS Enterprise Miner Personal Workstation	28
Firewall	28
Monitoring the SAS Analytics Platform Server	30
Monitoring Using the Analytics Platform Console Application	30
Monitoring Using a Web Browser	34
Monitoring Using a JMX Console	36
Java Web Start Client Considerations	38
Windows Service Administration	38
Troubleshooting	39

Introduction

The SAS Analytics Platform provides a common application framework for the following analytical applications:

- SAS Enterprise Miner/SAS Text Miner
- SAS Forecast Server
- SAS Model Manager
- SAS Warranty Analysis

Centralizing common application functionality into one installable component simplifies the overall installation and administration process for these applications, especially when one takes advantage of the server functionality of the SAS Analytics Platform.

Most analytics applications that use the SAS Analytics Platform require the platform to be run as a mid-tier server, which provides access to its installed applications via remote clients.

SAS Enterprise Miner also allows you to run the SAS Analytics Platform as an embedded service, so that running a mid-tier server is not necessary. Running the SAS Analytics Platform in this way is commonly referred to as a *Personal Workstation* deployment, and is useful for users who prefer to have the entire application (client, remote, and foundation components) available on one machine, without any dependency on the availability of a network connection.

The SAS Analytics Platform provides applications a common access point to the SAS Foundation Services, the SAS Metadata Server, and the various SAS workspace servers defined in the metadata server.

Migration

The SAS Deployment Wizard can be used to migrate configuration properties from a SAS Analytics Platform 1.4 image to configure SAS Analytics Platform 1.5 using a migration package which was created using the SAS Migration Utility.

The SAS Migration Utility's `smu.properties` file contains a property named `SMU.apcore.migration.is_enabled` which is set to **false** by default. One will need to change this value to **true** once migration scripts are available for all of the v913 SAS Analytics Platform-based applications (SAS Enterprise Miner, SAS Forecast Server, SAS Model Manager, and SAS Warranty Analysis) which are configured at the customer's site. **The SAS Migration Utility only allows a customer's configuration to be migrated once**, so if there is a product such as SAS Warranty Analysis whose release occurs later than SAS Enterprise Miner, SAS Forecast Server and SAS Model Manager, then the customer would need to either wait until SAS Migration Utility scripts were available for all configured products or perform the migration without migrating SAS Warranty Analysis' configuration.

When a SAS Analytics Platform configuration is migrated it will use some of the previous settings as its new configuration defaults as described below.

- Multicasting
 - Uses the v913 multicasting preference as the default value for the SAS Deployment Wizard prompt used to specify whether multicast services should be enabled.
- RMI Security
 - Uses the v913 RMI security mode preference as the default for the SAS Deployment Wizard prompt used to specify whether none, some or all RMI services are to be secured using the Java Secure Socket Extension (JSSE).
 - Uses the v913 RMI security preferences as the default values for the SAS Deployment Wizard prompts used to select which collections of services should be secured using JSSE.
 - RMI Registry
 - SAS Analytics Platform
 - SAS Enterprise Miner
 - SAS Forecast Server
 - SAS Model Manager
 - SAS Warranty Analysis

The original port settings are described in the SAS Analytics Platform Server's migration package, but their values are not migrated since each SAS Analytics Platform 1.5 configured image defaults to using a unique set of ports which allows one to independently configure up to ten images side-by-side.

Configuration

The SAS Deployment Wizard is used to both install and configure the SAS Analytics Platform. One may optionally migrate a configuration from a v913 SAS Analytics Platform. Use the SAS Deployment Manager to unconfigure the SAS Analytics Platform (see “Unconfiguring the SAS Analytics Platform Server using the SAS Deployment Manager” on page 12).

The SAS Deployment Wizard will present a series of prompts which will require responses from you to specify the configuration for a SAS Analytics Platform Server as summarized below.

- **SAS Metadata Server**
 - host
 - port
- **Embedded Tomcat HTTP Server**
 - whether the embedded HTTP server should be started
 - HTTP port

Note: This provides the ability to launch applications, such as SAS Enterprise Miner and SAS Forecast Studio, from the SAS Analytics Platform monitor page using Java Web Start.
- **Ports**
 - **Embedded HTTP Server port** - The port used to communicate with the embedded HTTP server.
 - **RMI Registry port** - The port used by the RMI registry to listen for client lookup requests. SAS Analytics Platform clients only need to know the host name and RMI registry port to locate the SAS Analytics Platform Server.
RMI Plain-Text port - The port used by the RMI services which use default non-secure sockets. If JSSE security is enabled for “All” RMI services, this port is not used.
 - **RMI Secure Sockets port** - The port used by RMI services which are secured using JSSE. This port is only used when JSSE security is enabled.
- **SAS Analytics Platform Startup**
 - Automatically start the SAS Analytics Platform Server - This is the default behavior.
 - Enable automatic discovery of the server via multicasting - This option is not enabled by default. You should enable this option if the SAS Forecast Server client application needs to discover SAS Analytics Platform Servers by broadcasting a multicast message.
SAS Analytics Platform IP Multicast:
 - **IP Multicast Port** - the multicast port used to communicate presence of a SAS Analytics Platform Server to applications.
 - **IP Multicast Netaid Port** - the multicast port used to communicate presence of a SAS Analytics Platform Netaid server to applications.
 - **IP Multicast TTL** - The multicast “time to live” parameter which can be used to restrict the scope of the multicast communication.
- **SAS Analytics Platform Startup Timeout Period** - Specify the time to wait for the SAS Analytics Platform to start up.

- **Security mode**
 - None (default) - Use default, non-secure sockets for all RMI services
 - Some - Use JSSE to secure your choice of the following groups of RMI services
 - RMI Registry
 - SAS Analytics Platform
 - SAS Enterprise Miner
 - SAS Forecast Server
 - SAS Model Manager
 - SAS Warranty Analysis
 - All - Use JSSE to secure all RMI services

Note: If you select to secure "Some" or "All" RMI services, the remainder of the SAS Analytics Platform-specific SAS Deployment Wizard screens will guide you through creating a SAS Analytics Platform certificate and optionally importing that certificate into the truststore. The truststore for the client-side JRE (if on another machine) must be configured manually to import the certificate.

Ports

Ports are allocated in blocks of 10 corresponding to the 10 levels which can be configured on a given machine, where each LevN image defaults to a unique port. The following table summarizes the SAS Analytics Platform Server's default ports for each level.

Server	Lev1	Lev2	Lev3	Lev4	Lev5	Lev6	Lev7	Lev8	Lev9	Lev0
Embedded Tomcat HTTP Server	6401	6402	6403	6404	6405	6406	6407	6408	6409	6410
RMI Registry	6411	6412	6413	6414	6415	6416	6417	6418	6419	6420
RMI Service	6421	6422	6423	6424	6425	6426	6427	6428	6429	6430
RMI Service secured using JSSE	6431	6432	6433	6434	6435	6436	6437	6438	6439	6440
Multicast Discovery Server	6441	6442	6443	6444	6445	6446	6447	6448	6449	6450
Multicast Netaid Discovery Server	6451	6452	6453	6454	6455	6456	6457	6458	6459	6460

If the SAS Analytics Platform Server is protected by a firewall (see "Firewall" on page 28), then one must enable the client applications access by allowing them to open connections to the server's configured ports.

Embedded Tomcat HTTP Server

The SAS Analytics Platform Server can be configured to start an embedded Tomcat HTTP Server within the SAS Analytics Platform Server's Java Virtual Machine (JVM).

The SAS Analytics Platform Server's Web application allows one to:

- launch configured applications
- view the server's runtime status
- view the server's configuration

An advanced configuration option may be used to specify whether a Web client's access to the SAS Analytics Platform Server's Web application should be filtered based upon the IP address associated with the client's HTTP request. This allows one to control which Web application features are presented to a client. Note that the remote address may be the address of a proxy server. By default, this filter is disabled.

During configuration, if the Status Filter is enabled, then one will be prompted to list the IP addresses which are allowed to access the SAS Analytics Platform's applications, status and configuration. By default requests from all IP addresses may access applications, but only requests from the local machine may access the SAS Analytics Platform's status and configuration.

SAS Foundation Services

The SAS Analytics Platform Server will load metadata describing its SAS Foundation Services into the SAS Metadata Server when it is configured. This Foundation Services deployment metadata defines the configuration for the following services which are used by the SAS Analytics Platform Server:

- Authentication Service
- Information Service
- Logging Service
- Session Service
- Stored Process Service
- User Service

RMI Services

Client applications communicate with the SAS Analytics Platform Server using RMI-based services. By default, RMI services use default sockets which are not secure. The SAS Analytics Platform has been coded to enable its RMI services to use sockets which can be secured using the Java Secure Sockets Extension (JSSE). The following applications may be configured to secure their RMI services using JSSE:

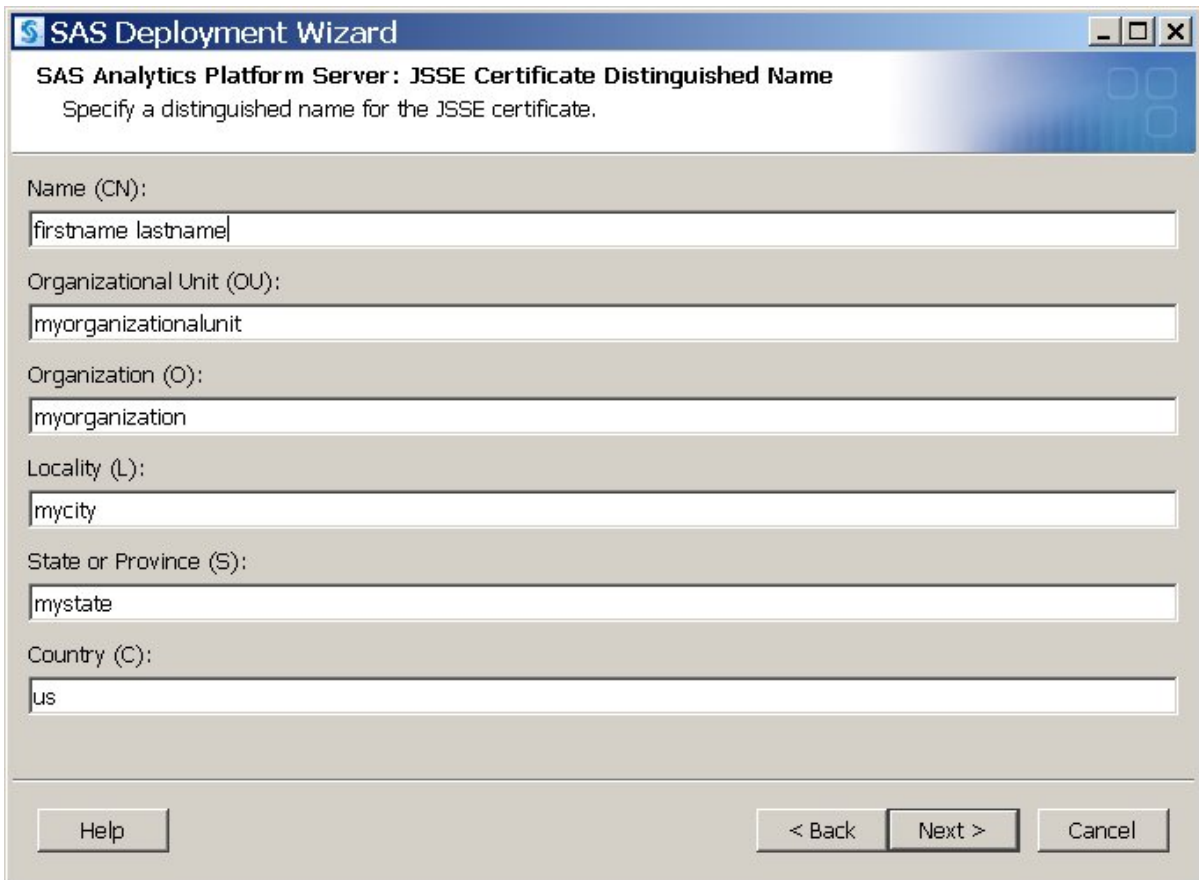
- SAS Enterprise Miner
- SAS Forecast Server
- SAS Model Manager
- SAS Warranty Analysis

The [JSSE Reference](http://java.sun.com/j2se/1.5.0/docs/guide/security/jsse/JSSERefGuide.html), located at <http://java.sun.com/j2se/1.5.0/docs/guide/security/jsse/JSSERefGuide.html>, provides a guide to the JSSE.

One may specify to secure none, some or all RMI communications as described in the following table.

Mode	Description
None	Use default (non-secure) sockets for RMI communication.
Some	Use JSSE to secure RMI communications based on preferences specified in the <code>jsse_selection.config</code> file.
All	Use JSSE to secure RMI communications for all services which have been coded to support RMI security. Note that RMI services which have not been coded to add a capability to be secured will use default (non-secure) sockets.

If RMI services are to be secured using JSSE, then one must configure a JSSE keystore for the SAS Analytics Platform Server. If RMI services are to be secured, then a JSSE keystore is configured with a self-signed certificate whose distinguished name was specified when the SAS Analytics Platform Server was configured using the SAS Deployment Wizard which will prompt for the certificate's distinguished name as shown below.



All configured LevN images share a common JRE. Per JSSE best practices, the JRE's default JSSE truststore (`JRE/lib/security/jssecacerts`) will be used. Passwords to access both the keystore and truststore must be specified when the SAS Analytics Platform Server is configured. The SAS Analytics Platform Server's SAS Deployment Wizard configuration script will configure both the keystore and truststore assuming that the person is permitted to execute the JRE's keytool and has write permission to the `JRE/lib/security` directory where the "jssecacerts" truststore file is located.

If the person who is configuring the SAS Analytics Platform Server lacks appropriate permissions, then the customer will need to manually configure the keystore and truststore as a manual post configuration process.

One may choose to either use this approach to automatically configure the JSSE keystore and default truststore (`JRE/lib/security/jssecacerts`) or one may choose to manually configure a keystore and/or truststore. The motivation for manually configuring a keystore would be to use a certificate which has been verified by a Certificate Authority, such as Verisign. Use of such a certificate eliminates the need to manually configure a truststore for a client application's JRE. These two approaches are summarized below. See the section entitled "Securing RMI Services Using JSSE" for additional details.

Self-Signed Certificate

If security is selected for RMI Services when the SAS Analytics Platform Server is configured a self-signed certificate will be created in the JSSE keystore specified by the user. By default a keystore named "apcore.keystore" is created in the SAS Analytics Platform's configuration directory, but an alternate location may be specified if desired.

This certificate will automatically be imported into the JRE's truststore (`JRE/lib/security/jssecacerts`) if the option to import the certificate into the truststore is selected when the SAS Analytics Platform Server is configured. Since one truststore is shared by all LevN configured images a certificate alias of "apcore_<LevelNumber>" is used, so that each LevN image remains independent.

If a client application is on another machine or is configured to use a different JRE than the SAS Analytics Platform Server, then one must manually import the server's certificate into the client JRE's truststore.

See the section entitled "Using a Self-Signed Certificate" for information which describes how to import the server's public certificate into the default JSSE truststore used by your client applications.

Certificate Authority Signed Certificate

Another option is to have a Certificate Authority (CA), such as Verisign, sign your certificate, so that you don't need to configure the client JRE's truststore. In this case, when the client application attempts to connect to the SAS Analytics Platform Server it will prompt its user to accept the server's certificate which has been signed by the CA.

During configuration, one may optionally create a certificate signature request (CSR) file for the certificate which has been created in the keystore. This CSR file can be submitted to a Certificate Authority (CA) such as VeriSign to be signed. The CA will return a signed certificate which will need to be loaded into the keystore using the JRE's keytool utility.

It is recommended that the keystore is created in a directory which does not reside within the SAS Analytics Platform configuration directory which is deleted when the SAS Analytics Platform Server is unconfigured. Also note that if multiple LevN images are configured, that one may want to share a keystore among all configured LevN images rather than individually configuring a keystore for each LevN image.

See "Using a Certificate Signed by a Certificate Authority" on page 20 which describes how to manually configure the SAS Analytics Platform Server's keystore to use a certificate which has been signed by a CA.

Securing Selected Services

If the user decides that security is only required for some of the RMI services, then the user may select the "Some" security mode and then select the groups of services which need to be secured using JSSE.

- RMI Registry
- SAS Analytics Platform
- SAS Enterprise Miner
- SAS Forecast Server
- SAS Model Manager
- SAS Warranty Analysis

For example, one may specify that the SAS Analytics Platform, RMI Registry and SAS Enterprise Miner services are to be secured using JSSE while the SAS Forecast Server, SAS Model Manager and SAS Warranty Analysis services will not be secured.

Multicast Discovery Servers

The SAS Analytics Platform Server may be configured to enable two multicast discovery servers. Multicast discovery servers are disabled by default since only the SAS Forecast Server application uses this feature. If Multicast Discovery is enabled, then one can configure the multicast address and ports for the following multicast servers:

- Discovery Server - used to obtain runtime status
- NetAid Discovery Server - used by clients to locate SAS Analytics Platform Servers

The default multicast address is the IPv4 address 239 . 192 . 65 . 80. Since the SAS Forecast Studio client is configured to use this IPv4 address, this address is not configurable using the SAS Deployment Wizard.

One may also specify the Time-To-Live (TTL) which is the maximum number of hosts a datagram may transit before it is discarded. One may specify a value of 0 to restrict the datagram to the localhost.

The following table summarizes the default multicast ports for each LevN image.

Multicast Server	Lev1	Lev2	Lev3	Lev4	Lev5	Lev6	Lev7	Lev8	Lev9	Lev0
Discovery Server	6441	6442	6443	6444	6445	6446	6447	6448	6449	6450
Netaid Discovery Server	6451	6452	6453	6454	6455	6456	6457	6458	6459	6460

Startup Options

The SAS Analytics Platform Server can be configured to automatically start once it has been configured by the SAS Deployment Wizard. A startup timeout defines the amount of time, in units of seconds, before a startup attempt is deemed a failure.

If this option is not selected, then one must manually start the SAS Analytics Platform Server using its script.

Startup preferences, including the minimum and maximum heap size for the JVM as well as additional JVM options may be configured.

If the SAS Analytics Platform Server is configured on a multi-homed machine, then one should specify the "java.rmi.server.hostname" property as an additional JVM option (for example, "-Djava.rmi.server.hostname=10.192.33.45")

Windows Service

If the SAS Analytics Platform Server is configured to be started as a service on the Windows platform, then the following defaults will be used to specify the service's name, display name and description.

SAS Deployment Wizard

SAS Analytics Platform Server: Windows Service
Specify Windows service information for SAS Analytics Platform Server.

Service Name:
SAS [Config-Lev2] Analytics Platform Server

Service Display Name:
SAS [Config-Lev2] Analytics Platform Server

Service Description:
Analytics Platform Server at Config-Lev2 on port 6412

Help < Back Next > Cancel

File System Layout

The SAS Analytics Platform configures into a directory whose location is specified by the SAS Deployment Wizard. This directory contains scripts used to start/stop the application, configuration files and the following sub-directories:

- **apps** – the *applications directory* into which individual applications are configured, each within their own subdirectory. For example:
 - EnterpriseMiner
 - ForecastServer
 - ModelManager
 - WarrantyAnalysis
- **conf** – the *configuration directory* for the embedded Tomcat HTTP Server.

- **lib** – If an HTTP Server is enabled in the SAS Analytics Platform Server, then this “lib” directory will contain its Web application archive, `sas.apps.session.war`, which is created when the SAS Deployment Wizard configures the SAS Analytics Platform Server or when one uses the SAS Deployment Manager to rebuild the SAS Analytics Platform Server Web application. The customer may also add Java library (`.jar`) files to this directory if necessary. Note that SAS `.jar` files are obtained from the SAS Versioned Jar Repository (VJR), so this “lib” directory should not contain any SAS `.jar` files. Each application will have its own “lib” directory (for example, `apps/EnterpriseMiner/lib`, `apps/ForecastServer/lib`, etc.) to allow the customer to add `.jars` if necessary.
- **Logs** – this directory contains log files.
- **Temp** - this directory contains temporary files.
- **wars** - this directory is used to rebuild Web archive files for the embedded Tomcat HTTP Server.
- **work** - used by the internal Web server when enabled.

Unconfiguring the SAS Analytics Platform Server using the SAS Deployment Manager

If it becomes necessary to unconfigure the SAS Analytics Platform Server using the SAS Deployment Manager, follow these steps:

1. Stop the SAS Analytics Server using its script (Windows: **`AnalyticsPlatform.bat stop`** or the Windows Shortcut if installed; UNIX: **`AnalyticsPlatform.sh stop`**).
2. Start the SAS Deployment Manager.
 - a. Select the **Remove Existing Configuration** radio button.
 - b. Choose the configuration to be removed from the **Select Configuration Directory** table.
 - c. Specify the **user ID** and **Password** to be used to connect to the SAS Metadata Server.
 - d. Select the products which are to be removed.
 - e. Unconfigure the products.

Securing RMI Services Using JSSE

This section describes procedures which may be used to configure the server's keystore and the client JRE's truststore to enable a client to access the SAS Analytics Platform Server's RMI services if they have been configured to be secured using the Java Secure Socket Extension (JSSE).

Instructions are provided for keytool procedures which may be used to configure a JSSE keystore for the SAS Analytics Platform Server and to configure a client application JRE's truststore. These procedures may be used to manually configure JSSE keystore and optionally client JRE truststores if the SAS Analytics Platform Server's RMI services have been configured to be secured using the JSSE.

For more information about working the Java's Key and Certificate Management Tool, `keytool`, visit the following sites:

- [Key and Certificate Management Tool for Windows](http://java.sun.com/j2se/1.5.0/docs/tooldocs/windows/keytool.html) at <http://java.sun.com/j2se/1.5.0/docs/tooldocs/windows/keytool.html>.
- [Key and Certificate Management Tool for Solaris and Linux](http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html) at <http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html>.

Configuration Options

If RMI services are to be secured using the JSSE, then one must configure the SAS Analytics Platform Server's keystore using one of the following two approaches:

- "Using a Self-Signed Certificate" on page 17.
- "Using a Certificate Signed by a Certificate Authority" on page 20.

Note that if a self-signed certificate is used, then one must also configure the client JRE's truststore (`JRE/lib/security/jssecacerts`).

The following keytool procedures are applicable to using a self-signed certificate and using a certificate which has been signed by a Certificate Authority.

- "Create Keystore Certificate" on page 14.
- "List Certificates" on page 15.
- "Delete Certificate" on page 16.

Create Keystore Certificate

This procedure is used to create a self-signed certificate in a JSE keystore for use by the SAS Analytics Platform Server. Required inputs are described in the following table.

Required Input	Example
keystore's filename	<i>apcore.keystore</i>
key algorithm	<i>RSA</i>
keystore's password	<i>secretPassword</i>
certificate's alias	<i>analyticsplatformserver</i>
certificate's distinguished name	<ul style="list-style-type: none"> • <i>CN=MyWebSite</i> • <i>OU=MyOrganizationalUnit</i> • <i>O=MyOrganization</i> • <i>L=MyCity</i> • <i>ST=MyStateOrProvince</i> • <i>C=US</i>

For example, to create a certificate for the SAS Analytics Platform Server's keystore:

1. Open a command window.
2. Ensure that the path contains the JRE/bin folder. For example:

```
set path=%path%;C:\Program Files\Java\jre1.5.0_12\bin
```
3. Change to your keystore's directory. Choose a directory which does not reside under the SAS Analytics Platform Server's configuration directory since the configuration directory will be deleted when the SAS Analytics Platform Server is unconfigured.
4. Issue the following command to create a self-signed certificate in your keystore file.

```
keytool -genkey
        -alias <certificateAlias>
        -keystore <keystoreFile>
        -storepass <keystorePassword>
        -keyalg <keyAlgorithm>
```

For example, to create a certificate whose alias is "analyticsplatformserver" in the keystore file "apcore.keystore," change to the directory which will contain the keystore and then issue the following command:

```
keytool -genkey
        -alias "analyticsplatformserver"
        -keystore apcore.keystore
        -storepass secretPassword
        -keyalg RSA
```

The keytool will then prompt you to specify the following:

- a. Enter your first and last name (for example, *MyWebSite*).
 - b. Enter the name of your organizational unit (for example, *MyOrganizationalUnit*).
 - c. Enter the name of your organization (for example, *MyOrganization*).
 - d. Enter the name of your city or locality (for example, *MyCity*).
 - e. Enter the name of your state or province (for example, *MyStateOrProvince*).
 - f. Enter the two-letter country code (for example, *US*).
 - g. If you are satisfied with the information that you entered in the previous steps, enter **"y"** at the prompt.
 - h. Accept the prompt to use the keystore's password as the certificate's password.
5. Verify that the keytool created a keystore file named `"apcore.keystore"`.
 6. Use the `"List Certificates"` procedure described on page 15 to verify that a certificate whose alias is `"analyticsplatformserver"` was created in the keystore.

List Certificates

This procedure is used to view a listing of the certificates which are defined in a JSE keystore or truststore.

Required Input	Example
keystore's filename	<i>apcore.keystore</i>
keystore's password	<i>secretPassword</i>

For example, to delete a certificate from the JRE's default JSE truststore (JRE/lib/security/jssecacerts):

1. Open a command window.
2. Ensure that the path contains the JRE/bin folder. For example:

```
set path=%path%;C:\Program Files\Java\jre1.5.0_12\bin
```
3. Change to the directory which contains the truststore file.
4. Issue the following command to delete the certificate from the truststore file.

```
keytool -list
       -keystore <keystoreFile>
       -storepass <keystorePassword>
```

For example, to list the certificates in the keystore file “*apcore.keystore*” whose password is “*secretPassword*” one would issue the following command:

```
keytool -list
       -keystore apcore.keystore
       -storepass secretPassword
```

Delete Certificate

This procedure is used to delete a certificate from a keystore or truststore. Required inputs are described in the following table.

Required Input	Example
keystore's filename	<i>jssecacerts</i>
keystore's password	<i>changeit</i>
certificate's alias	<i>analyticsplatform</i>

For example, to delete a certificate from the JRE's default JSSE truststore (JRE/lib/security/jssecacerts):

5. Open a command window.
6. Ensure that the path contains the JRE/bin folder. For example:

```
set path=%path%;C:\Program Files\Java\jre1.5.0_12\bin
```
7. Change to the directory which contains the truststore file (JRE/lib/security).
8. Issue the following command to delete the certificate from the truststore file.

```
keytool -delete
       -alias <certificateAlias>
       -keystore <truststoreFile>
       -storepass <truststorePassword>
```

For example, to delete the certificate whose alias is “*analyticsplatform*” from the truststore file “*jssecacerts*” whose default password is “*changeit*” one would issue the following command:

```
keytool -delete
       -alias "analyticsplatform"
       -keystore jssecacerts
       -storepass changeit
```

Using a Self-Signed Certificate

If security is selected for RMI Services when the SAS Analytics Platform Server is configured a self-signed certificate will be created in the JSSE keystore. By default a keystore named "apcore.keystore" is created in the SAS Analytics Platform's configuration directory, but an alternate location may be specified if desired.

This certificate will be imported into the JRE's default truststore (JRE/lib/security/jssecacerts) if the option to import the certificate into the truststore is selected when the SAS Analytics Platform Server is configured. Since one truststore is shared by all LevN configured images a certificate alias of "apcore_<LevelNumber>" (for example, apcore_1 for a Lev1 image) is used, so that each LevN image remains independent.

If a client application is on another machine or is configured to use a different JRE than the SAS Analytics Platform Server, then one must manually import the server's certificate into the client JRE's truststore.

Use this approach if you do not want to have a Certificate Authority sign a certificate for the SAS Analytics Platform Server and will ensure that the server's public self-signed certificate is imported into the JRE's default truststore used by all client applications. Use the following procedure to configure a self-signed certificate in the server's keystore and then import the server's public certificate into the client JRE's truststore.

1. "Create Keystore Certificate" on page 14.
2. "Export Certificate from Keystore" on page 17.
3. "Import Keystore's Certificate into Truststore" on page 18 (for each client JRE).

Export Certificate from Keystore

This procedure is used to export a certificate from a JSSE keystore. Required inputs are described in the following table.

Required Input	Example
keystore's filename	<i>apcore.keystore</i>
keystore's password	<i>secretPassword</i>
certificate's alias	<i>analyticsplatformserver</i>
name of the output file created by the keytool which will contain the server's public certificate	<i>analyticsplatformserver.cer</i>

For example, to create a certificate for the SAS Analytics Platform Server's keystore:

1. Open a command window.
2. Ensure that the path contains the JRE/bin folder. For example:

```
set path=%path%;C:\Program Files\Java\jre1.5.0_12\bin
```

3. Change to your keystore's directory. Choose a directory which does not reside under the SAS Analytics Platform Server's configuration directory structure since the configuration directory will be deleted when the SAS Analytics Platform Server is unconfigured.
4. Issue the following command to create a self-signed certificate in your keystore file.

```
keytool -export
        -alias <certificateAlias>
        -keystore <keystoreFile>
        -storepass <keystorePassword>
        -file <certificate.cer>
```

For example, to export a public certificate for the alias "analyticsplatformserver" in the keystore file "apcore.keystore," change to the directory which will contain the keystore and then issue the following command:

```
keytool -export
        -alias "analyticsplatformserver"
        -keystore apcore.keystore
        -storepass secretPassword
        -file analyticsplatformserver.cer
```

5. Verify that the keytool created a certificate file named "analyticsplatform.cer". This file contains the server's public self-signed certificate which can be imported into a client JRE's default JSSE truststore (JRE/lib/security/jssecacerts) to enable the client to connect to a SAS Analytics Platform's RMI services which have been secured using the JSSE.
6. Use the "Import Keystore's Certificate into Truststore" procedure below to import the server's public certificate into the client JRE's truststore (JRE/lib/security/jssecacerts).

Import Keystore's Certificate into Truststore

This procedure is used to take a self-signed certificate which has been exported from the SAS Analytics Platform Server's JSSE keystore and then import it into a client application's JRE truststore.

For a client to authenticate a server either the server's keystore must contain a certificate which has been signed by a Certificate Authority (CA) or the client's truststore must contain the server's public certificate. This procedure is used for the case where a self-signed certificate is used by the server which requires the client to import a public certificate which has been exported from the server's keystore.

When you create a TrustManager, the Sun implementation first checks for an alternate cacerts file before falling back to the standard cacerts file. This enables you to provide a JSSE-specific set of trusted certificates separate from those which may be present in the cacerts for code signing purposes.

JSSE uses the following search order to locate the truststore:

1. `-Djavax.net.ssl.trustStore=<trustStoreFilePath>`
2. `<java_home>\lib\security\jssecacerts`
3. `<java_home>\lib\security\cacerts`

Note that if the `jssecacerts` file is found, then the search stops and the `cacerts` file is not used. Per best practice recommendations, the server's public certificate will be imported into our client JRE's `jssecacerts` file which contains the client's trusted certificates.

Required inputs are described in the following table.

Required Input	Example
truststore's filename	<i>jssecacerts</i>
truststore's password	<i>changeit</i>
certificate's alias	<i>analyticsplatformserver</i>
name of the file which contains the public certificate which was exported from the SAS Analytics Platform Server's keystore	<i>analyticsplatformserver.cer</i>

For example, to import the SAS Analytics Platform Server's public certificate into the client JRE's default JSSE truststore:

1. Open a command window.
2. Ensure that the path contains the JRE/bin folder. For example:

```
set path=%path%;C:\Program Files\Java\jre1.5.0_12\bin
```
3. Change to your client JRE's `lib/security` directory which is the where the JSSE code will search for the `jssecacerts` file.
4. Issue the following command to import the server's public certificate into the client JRE's truststore file (`JRE/lib/security/jssecacerts`).

```
keytool -import
        -alias <certificateAlias>
        -keystore <truststoreFile>
        -storepass <truststorePassword>
        -file <analyticsplatfromserver.cer>
```

For example, to import the server's public certificate for the alias "analyticsplatformserver" into the truststore file "jssecacerts," change to the `JRE/lib/security` directory and then issue the following command:

```
keytool -import
        -alias "analyticsplatformserver"
        -keystore jssecacerts
        -storepass changeit
        -file analyticsplatformserver.cer
```

5. Use the "List Certificates" procedure on page 15 to verify that a certificate whose alias is "analyticsplatformserver" was created in the `jssecacerts` truststore.

Using a Certificate Signed by a Certificate Authority

This section describes the procedures which are used to configure the SAS Analytics Platform Server's keystore to use a certificate which has been signed by a Certificate Authority. If one uses this approach, then one does not need to configure the client JRE's truststore since the server's certificate can be verified by the Certificate Authority. The client will be prompted to accept the server's certificate which will have been verified by the Certificate Authority.

[Verisign](#) provides instructions to describe how to install its primary and intermediate certificates into the JRE's `lib/security/cacerts` file if the certificates are not already present. If the certificate is already defined in the `cacerts` file when one attempts to use the `keytool` to import the certificate then a prompt is issued providing notification that the certificate is already defined. There is no need to replace the certificate if the certificate is already defined.

1. "Import CA's Primary Certificate into JRE/lib/security/cacerts" (below).
2. "Import CA's Intermediate Certificate into JRE/lib/security/cacerts" (on page 22).

Once the Certificate Authority's primary and root certificates are defined in the JRE's `lib/security/cacerts` file, then one must create a certificate in the SAS Analytics Platform Server's keystore, generate a certificate signature request (CSR) file which will be submitted to the Certificate Authority and then import the CA's reply back into the keystore.

1. "Create Keystore Certificate" on page 14.
2. "Generate Certificate Signature Request (CSR)" on page 23.
3. "Submit CSR to Certificate Authority (CA)" on page 24.
4. "Import Certificate Authority's Reply into Keystore" on page 24.

Import CA's Primary Certificate into JRE/lib/security/cacerts

This procedure is used to import a certificate authority's primary root certificate into the `JRE/lib/security/cacerts` file. Required inputs are described in the following table. Refer to the CA's documentation for specific instructions. For example, view [Verisign's](#) instructions for Tomcat. Verisign's Primary PCA Root Certificates may be downloaded from <http://www.verisign.com/support/roots.html>.

Required Input	Example
keystore's filename	<i>cacerts (JRE/lib/security)</i>
keystore's password	<i>changeit</i>
certificate's alias	<i>intermediateCA</i>
Name of the file which contains the Certificate Authority's intermediate certificate.	<i>intermediateCA.cer</i>

For example, to import Verisign's primary certificate into the `JRE/lib/security/cacerts` file:

1. Open a command window.
2. Ensure that the path contains the `JRE/bin` folder. For example:

```
set path=%path%;C:\Program Files\Java\jre1.5.0_12\bin
```
3. Download the root certificates from <http://www.verisign.com/support/roots.html> and unzip the file to the `C:\temp` directory.
4. Change to your `JRE/lib/security` directory where the JRE's `cacerts` file is located.
5. Issue the following command to import the Certificate Authority's intermediate certificate into your keystore file.

```
keytool -import
        -alias <certificateAlias>
        -keystore <keystoreFile>
        -storepass <keystorePassword>
        -file <intermediateCA.cer>
```

For example, to import a certificate for the alias "intermediateCA" into the JRE's `cacerts` file, change to the `JRE/lib/security` directory and then issue the following command:

```
keytool -import
        -alias "intermediateCA"
        -keystore cacerts
        -storepass changeit
        -file "C:\temp\Root Download Package\Verisign
        Roots\PCA3ss_v4.509"
```

If the keytool reports that the CA's certificate is already present in the `cacerts` file, then there is no need to re-import it.

Import CA's Intermediate Certificate into JRE/lib/security/cacerts

This procedure is used to import a certificate authority's intermediate certificate into the `JRE/lib/security/cacerts` file. Required inputs are described in the following table. Refer to the CA's documentation for specific instructions. For example, view [Verisign's](http://sww.sas.com/computersecurity/ca/intermediate_verisign.509) instructions for Tomcat. Verisign's intermediate certificate may be obtained from http://sww.sas.com/computersecurity/ca/intermediate_verisign.509.

Required Input	Example
keystore's filename	<i>cacerts (JRE/lib/security)</i>
keystore's password	<i>changeit</i>
certificate's alias	<i>intermediateCA</i>
Name of the file which contains the Certificate Authority's intermediate certificate.	<i>intermediateCA.cer</i>

For example, to import Verisign's intermediate certificate into the `JRE/lib/security/cacerts` file:

1. Open a command window.
2. Ensure that the path contains the `JRE/bin` folder. For example:

```
set path=%path%;C:\Program Files\Java\jre1.5.0_12\bin
```
3. Change to your `JRE/lib/security` directory where the JRE's `cacerts` file is located.
4. Issue the following command to import the Certificate Authority's intermediate certificate into your keystore file.

```
keytool -import
        -alias <certificateAlias>
        -keystore <keystoreFile>
        -storepass <keystorePassword>
        -file <intermediateCA.cer>
```

For example, to import a certificate for the alias "intermediateCA" into the JRE's `cacerts` file, change to the `JRE/lib/security` directory and then issue the following command:

```
keytool -import
        -alias "intermediateCA"
        -keystore cacerts
        -storepass changeit
        -file intermediateCA.cer
```

If the keytool reports that the CA's certificate is already present in the `cacerts` file, then there is no need to re-import it.

Generate Certificate Signature Request (CSR)

This procedure is used to generate a file which contains a certificate signature request for a self-signed certificate defined in the SAS Analytics Platform Server's JSSE keystore. This CSR file may then be submitted to a certificate authority to be verified. Note that the alias must be the same value used to generate the certificate in the keystore.

Required Input	Example
keystore's filename	<i>apcore.keystore</i>
keystore's password	<i>secretPassword</i>
certificate's alias	<i>analyticsplatformserver</i>
name of the output file created by the keytool which will contain the certificate signature request (CSR)	<i>analyticsplatformserver.csr</i>

For example, use the following procedure to generate a certificate signature request (CSR) file which can be submitted to a Certificate Authority:

1. Open a command window.
2. Ensure that the path contains the JRE/bin folder. For example:

```
set path=%path%;C:\Program Files\Java\jre1.5.0_12\bin
```
3. Change to your keystore's directory. Choose a directory which does not reside under the SAS Analytics Platform Server's configuration directory structure since the configuration directory will be deleted when the SAS Analytics Platform Server is unconfigured.
4. Issue the following command to generate the CSR which is to be submitted to the Certificate Authority.

```
keytool -certreq
        -alias <certificateAlias>
        -keystore <keystoreFile>
        -storepass <keystorePassword>
        -file <certificateSignatureRequest.csr>
```

For example, to import a signed certificate for the alias "analyticsplatformserver" in the keystore file "apcore.keystore," change to the directory which will contain the keystore and then issue the following command:

```
keytool -certreq
        -alias "analyticsplatformserver"
        -keystore apcore.keystore
        -storepass secretPassword
        -file analyticsplatformserver.csr
```

5. Verify that the keytool created a CSR file named "analyticsplatform.csr." This file contains the certificate signature request which must be submitted to a Certificate Authority.

Submit CSR to Certificate Authority (CA)

This procedure is used to submit a certificate signature request (CSR) file to a certificate authority (CA), such as Verisign, to be verified. The CA will reply with a file which must then be imported into the SAS Analytics Platform Server's JSE keystore.

1. Submit the certificate signature request (CSR) file to a Certificate Authority (CA) such as Verisign. Contact your company's IT, Network or Security group for assistance.
2. The CA will reply with an e-mail providing the certificate as a file attachment or included in the body of the e-mail message. If the certificate is in the body of the e-mail, then copy/paste it into a text file using either Notepad or vi. Do not use an editor which may add extra characters thereby corrupting the certificate's signature.

Import Certificate Authority's Reply into Keystore

This procedure is used to take a certificate authority's reply to a certificate signature request (CSR) and then import it into the SAS Analytics Platform Server's keystore. Required inputs are described in the following table. Note that the alias must be the same value used to generate the private key and the CSR which was submitted to the CA.

Required Input	Example
keystore's filename	<i>apcore.keystore</i>
keystore's password	<i>secretPassword</i>
certificate's alias	<i>analyticsplatformserver</i>
Name of the file which contains the Certificate Authority's reply to the certificate signature request.	<i>Cert.cer</i>

For example, to import an SSL certificate obtained from a Certificate Authority into the SAS Analytics Platform Server's keystore:

1. Open a command window.
2. Ensure that the path contains the JRE/bin folder. For example:

```
set path=%path%;C:\Program Files\Java\jre1.5.0_12\bin
```
3. Change to your keystore's directory. Choose a directory which does not reside under the SAS Analytics Platform Server's configuration directory structure since the configuration directory will be deleted when the SAS Analytics Platform Server is unconfigured.
4. Issue the following command to import the Certificate Authority signed certificate into your keystore file.

```
keytool -import -trustcacerts
        -alias <certificateAlias>
        -keystore <keystoreFile>
        -storepass <keystorePassword>
        -file <certificateReplyFromCA.cer>
```

For example, to import a signed certificate for the alias "analyticsplatformserver" in the keystore file "apcore.keystore," change to the directory which will contain the keystore and then issue the following command:

```
keytool -import -trustcacerts  
    -alias "analyticsplatformserver"  
    -keystore apcore.keystore  
    -storepass secretPassword  
    -file Cert.cer
```


Runtime

This section discusses runtime considerations for the SAS Analytics Platform Server.

Starting the SAS Analytics Platform Server

The SAS Analytics Platform is started by the use of a script. When the SAS Analytics Platform Server first starts, it needs to connect to the SAS Metadata Server in order to initialize its runtime environment. By default, the server is configured to use a persisted set of credentials in order to connect to the SAS Metadata Server.

All configured applications are discovered during this initialization phase. Thus, if you want to configure another SAS Analytics Platform application, then one must stop the SAS Analytics Platform Server, use the SAS Deployment Wizard to configure the new application and then restart the SAS Analytics Platform Server so that the new application will be recognized.

On the Windows platform, the SAS Analytics Platform Server can be automatically started when it is configured using the SAS Deployment Wizard. One may also configure the SAS Analytics Platform Server as a Windows Service. The sections below detail the steps required to start the SAS Analytics Platform for each supported OS environment.

Starting the SAS Analytics Platform Server under Windows

By default, the server is configured to use the Trusted User credential in order to connect to the SAS Metadata Server.

DOS Console Output

Use the Windows shortcut to start the server.

Start → Programs → SAS → SAS Configuration → Config – LevN → Analytics Platform Server: Start

This will start the SAS Analytics Platform Server in a DOS console window. The server is ready to receive clients when the message “Waiting for clients” appears at the bottom of the screen.

You can also open a DOS window and change to the directory where the SAS Analytics Platform Server is configured and type:

```
AnalyticsPlatform.bat start
```

or

```
AnalyticsPlatform.bat start > .\Logs\myLog.txt
```

This lets you control the name of the log file used to capture the various system messages.

Starting the SAS Analytics Platform Server under UNIX

Under UNIX, you will need to open a terminal session and change to the directory where the SAS Analytics Platform Server is configured. Issue the command:

```
./AnalyticsPlatform.sh start
```

The server is ready to receive clients when the message “Analytics Platform – started” appears at the bottom of the screen.

By default, system log messages are sent to `stdout`, so you can use common UNIX shell syntax to redirect these messages to a log file:

```
./AnalyticsPlatform.sh start > ./Logs/myLog.txt
```

Stopping the SAS Analytics Platform Server

Occasionally it may be necessary to stop the server, such as in the case when you need to restart it when configuring a new application or unconfiguring an application. You can either use the scripts provided to issue the shutdown command or shutdown the server using the SAS Analytics Platform Server Console application.

Stopping the SAS Analytics Platform under Windows

Use the Windows shortcut:

Start → Programs → SAS → SAS Configuration → Config - LevN → Analytics Platform Server: Stop

This will cause the SAS Analytics Platform Server to stop after approximately 5 seconds. Alternatively, you can open a DOS window, navigate to the directory where the SAS Analytics Platform is configured and issue the command specifying the number of seconds to wait before initiating the shutdown:

```
AnalyticsPlatform.bat stop -t seconds
```

Stopping the SAS Analytics Platform under UNIX

Open a terminal session, make sure you have an X server running and available, and change to the directory where the SAS Analytics Platform Server is configured. Issue the command specifying the number of seconds to wait before initiating the shutdown:

```
./AnalyticsPlatform.sh stop -t seconds
```

Note: The SAS Analytics Platform server can also be shutdown using the UNIX `sas.servers` script.

SAS Enterprise Miner Personal Workstation

The SAS Analytics Platform is typically run as a mid-tier server to allow clients to remotely access the set of applications which are configured in the SAS Analytics Platform Server.

Note for SAS Enterprise Miner Users: SAS Enterprise Miner offers the ability to run as a personal workstation. Therefore, it is not necessary to run the SAS Analytics Platform as a server if you only intend to run SAS Enterprise Miner in this mode.

A SAS Enterprise Miner personal workstation application will have the SAS Analytics Platform installed on the client machine, but will run it internally as its exclusive “client.”

However, if you intend to have multiple clients share a common access point that is centrally administered, then it is necessary to dedicate one mid-tier machine for the SAS Analytics Platform Server. Using this mid-tier also allows individual applications to realize benefits beyond centralized administration.

Firewall

The applications whose middle tier is provided by the SAS Analytics Platform can have clients access the SAS Analytics Platform Server through a firewall. To enable clients outside of the firewall to access the SAS Analytics Platform Server, it is necessary to permit client computers to

bi-directionally access the following ports (see "Ports" on page 6) which are described in the section which discusses Configuration:

- RMI Registry
- RMI Service (if RMI services are not configured to be secured)
- RMI Service secured using JSSE (if RMI services are configured to be secured)
- Embedded Tomcat HTTP Server (if enabled)
- Multicast Discovery Service (if multicast services are enabled)
- Multicast Netaid Discovery Service (if multicast services are enabled)

The following table summarizes the default port requirements for a Lev1 configuration:

Lev1 Default Port	Description
6401	If the embedded Tomcat HTTP Server was enabled during configuration, then this HTTP port must be opened bi-directionally.
6411	The RMI Registry port must be opened bi-directionally.
6421	If you configured the SAS Analytics Platform Server to not secure its RMI Services using the JSSE, then this port must be opened bi-directionally.
6431	If you configured the SAS Analytics Platform Server to secure its RMI Services using the JSSE, then this port needs to be open bi-directionally.
6441	If multicast services were enabled during configuration, then this multicast port must be opened to allow clients to access the multicast Discovery Service. This service is used by the Forecast Studio client's application logon dialog's "Find Servers" feature.
6451	If multicast services were enabled during configuration, then this multicast port must be opened to allow clients to access the multicast Netaid Discovery Service.

Windows XP (at Service Pack 2 level) contains a firewall that is often enabled. The Security Center Windows Firewall exceptions must include the following for SAS Analytics Platform family products (including SAS Enterprise Miner, SAS Forecast Studio, SAS Inventory Management Studio, and SAS Model Management Studio) clients to be able to access the SAS Analytics Platform Server, and for other SAS Java applications. Default paths are shown.

SAS Analytics Platform, SAS Enterprise Miner, SAS Forecast Studio, SAS Model Manager, and SAS Warranty Analysis products use the SAS private JRE version 1.5 or later.

C:\Program Files\SAS\Shared Files\JRE\1.5\bin

SAS Analytics Platform family product Java Web Start (JWS) clients use the publicly installed Sun Java library, which must be at version 1.5 or later. A typical install would locate the library in varying locations, but the default is:

C:\jdk1.5.0_12\jre\bin

or

C:\Program Files\Java\jre1.5.0_12\bin

Entries in the firewall exception list are usually set up automatically by the security center, but if there is no entry for the SAS private JRE's java.exe you must add one. The XP security center tested a program-type exception named "Java," and when edited showed the path to the SAS private JRE's java.exe in the "Path:" field of the exception properties.

Testing indicates this is the only exception entry necessary. You must change the "scope" of the program entry (there's no "port" entry involved) following these steps:

1. Launch the Security Center (from Control Panel) and enter Windows Firewall.
2. Select the **Exceptions** tab and the Java entry in the list.
3. Click **Edit**. The path will contain the path to the Java library noted above.
4. Click the **Change scope** button.
5. The tightest security is obtained by selecting the **Custom list** and entering the IP address of the client machine itself, a comma, the IP address of the machine on which the SAS Analytics Platform or shared platform (mid-tier) server (for SAS Enterprise Miner, SAS Forecast Server or SAS Model Manager) runs, a slash, and a full mask. For example,

```
192.168.9.73,192.168.9.83/255.255.255.255
```

The alternative is to either select the radio button that says **My network (subnet) only** if the server is in the same subnet, or to select **Any computer (including those on the internet)**. Since these options apply only for the SAS private copy of Java and only SAS Enterprise Miner will use it, there's minimal risk in allowing either option.
6. Click **OK** recursively to exit the firewall dialog and the settings are active immediately.

This will allow that Java program to communicate on any port with anything running on either machine. The firewall will protect that program from receiving anything on any port from any other machine if you used the **Custom list** option.

Monitoring the SAS Analytics Platform Server

The SAS Analytics Platform Server provides some monitoring tools that help you when running the SAS Analytics Platform as a mid-tier server. Please see the following sections for more information:

- "Monitoring Using the Analytics Platform Console Application" below.
- "Monitoring Using a Web Browser" on page on page 34.
- "Monitoring Using a JMX Console" on page on page 36.

Monitoring Using the Analytics Platform Console Application

The SAS Analytics Platform Server Console is a Java Swing application which allows one to view basic configuration and runtime status information for the server.

- Configuration
 - List of configuration properties
 - List of configured applications
- Status
 - List of the number of client user sessions currently attached to the server
 - Number of active SAS workspace sessions that have been initiated via this server
- Administration
 - Capability to shutdown the SAS Analytics Platform Server

Starting the Console

Before starting the SAS Analytics Platform console, you must have already started the SAS Analytics Platform Server on the same machine where you plan to run the console.

Windows

On the Windows platform one may start the SAS Analytics Platform Server Console using either its Windows shortcut or its script.

- Use the Windows shortcut:
Start → Programs → SAS → SAS Configuration → Config – LevN → Analytics Platform Server Console: Start
- Issue the command:
AnalyticsPlatformConsole.bat start

UNIX

On a UNIX platform one should start the SAS Analytics Platform Server Console application using the following procedure:

1. Open a terminal session.
2. Ensure that an X server is running and available.
3. Ensure that the DISPLAY environment variable points back to the machine being used for your terminal session.
4. Change to the directory where the SAS Analytics Platform Server is configured.
5. Issue the command:
./AnalyticsPlatformConsole.sh start

Problems

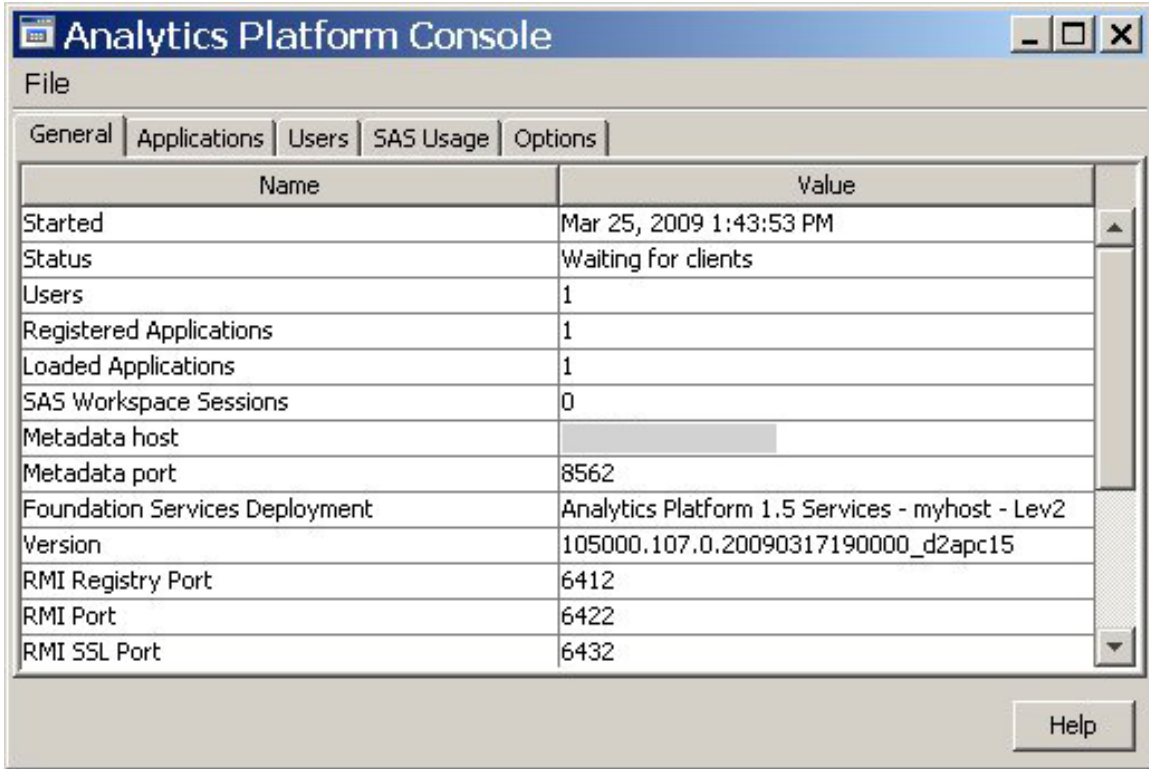
- A warning dialog is presented with a message that says “Could not reach the Analytics Platform Server” or “Server is not running.” If the SAS Analytics Platform Server is not running on this machine, start the SAS Analytics Platform Server and then try to start the SAS Analytics Platform Server Console. If the server fails to start, then check the SAS Analytics Platform log files to determine why the server startup failed.
- (UNIX) if you see a message that says “A graphical screen environment is required to run the console,” it means that your X environment is not set up correctly. Make sure you have set the DISPLAY environment variable to point back to the client machine you are using for your terminal session.

Using the Console

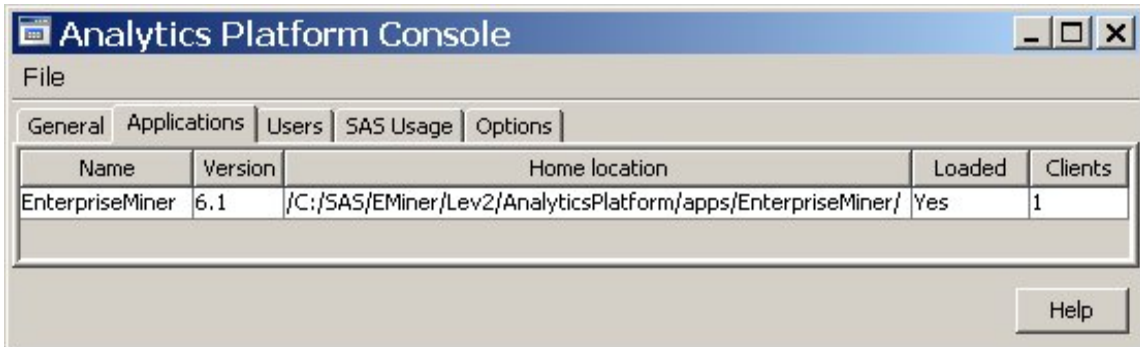
The **Analytics Platform Console** window uses a tabbed layout to organize its information.

- **General** - provides an overview of configuration properties and runtime status
- **Applications** - displays the list of installed applications, whether they have been loaded, and the number of clients which are currently using that application
- **Users** - displays the users which are on-line, when their session started, and from which IP address they are connected
- **SAS Usage** - displays a list of active SAS workspace sessions
- **Options** - controls the refresh polling rate of the console window and provides an action which can be used to shut down the server

The **General** tab provides an overview of the overall status of the system.

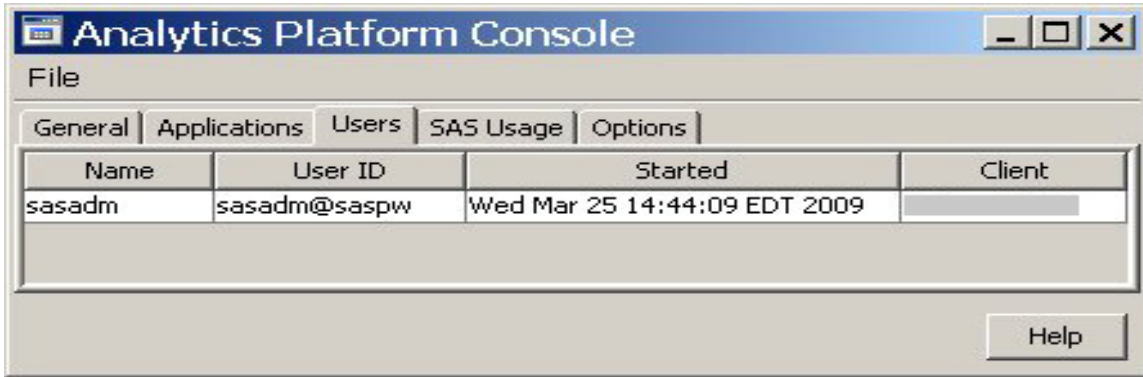


The **Applications** tab displays the list of installed applications, whether they have been loaded, and the number of clients which are currently using that application.



Note: If “Yes” appears in the **Loaded** column for an application, it means that the application Mid-Tier software has been loaded into memory. This happens when that application is requested by a user since the SAS Analytics Platform Server was started. SAS Model Manager is always loaded since it forces SAS Analytics Platform to load it during start up time. If it is not loaded, something is wrong with SAS Model Manager.

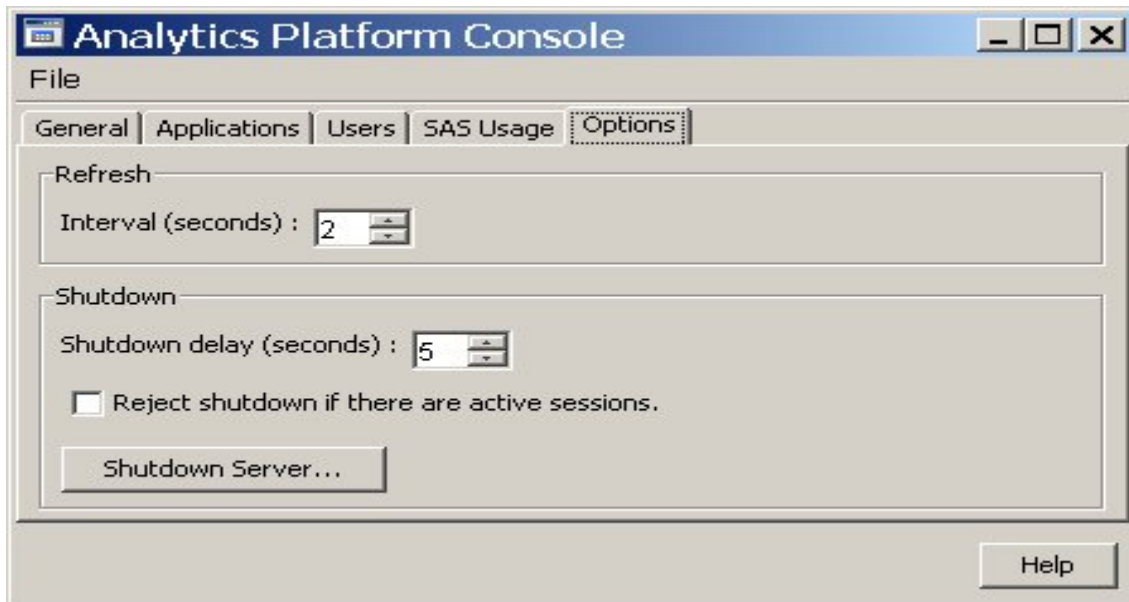
The **Users** tab displays the users which are on-line, when their session started, and from which IP address they are connected.



The **SAS Usage** tab displays a list of active SAS workspace sessions.



The **Options** tab lets you control the refresh rate of the console window and provides you with a button which can be used to shut down the server (when you invoke this action, the console application will exit).



Monitoring Using a Web Browser

It is possible to monitor the SAS Analytics Platform Server remotely using your Web browser. The information provided via your Web browser, and the information provided by the console application overlap to a large degree. However, it is *not* possible to shutdown the server using the Web interface, and you must manually refresh the information using your browser's refresh button.

Note: *This remote monitoring facility is only available when the embedded HTTP server is enabled. By default, the SAS Analytics Platform is installed with this feature enabled.*

Once you have started the SAS Analytics Platform Server, use your Web browser to point to the server machine, remembering to add the port to the URL. The Lev1 default port is 6401, but you can configure the embedded Tomcat HTTP Server's port when you configure the SAS Analytics Platform Server using the SAS Deployment Wizard.

For example, if the machine on which you are running the SAS Analytics Platform Server is called `myhost.mydomain.com` using the default Lev1 port, then you can point your Web browser to:

```
http://myhost.mydomain.com:6401
```

If everything is running correctly, you will see a page similar to the one on the next page which is organized using as many as three tabs: Applications, Activity and Configuration. If you have enabled the Status Filter servlet, then the tabs that you see will depend upon the IP address from which the page was accessed. Note that the Status Filter servlet is disabled by default (which is the same behavior as in SAS Analytics Platform 1.4). If the Status Filter is enabled during configuration, then one may configure the IP addresses which are allowed to access each tab. For example, one may want to allow anyone to access the Applications tab, but only allow access to the Activity and Configuration tabs from the machine on which the Analytics Platform Server was deployed.

The **Applications** tab lists the configured applications. Some applications may provide links to their own Web pages while others may provide a link which can be used to start a client application using Java Web Start technology. Java Web Start lets clients which have Java installed launch applications from a Web page without having to run a separate install program on the client's machine. You can access the "Applications" tab directly at:
<http://myhost.mydomain.com:6401/AnalyticsPlatform/Status?page=applications>.

Name	Version	Loaded	Clients	Java Web Start
EnterpriseMiner	6.1	Yes	0	Launch

The **Activity** tab displays dynamic system information such as the time that the server started, the number of users on-line, the number of active SAS workspace sessions, etc. You can access the "Activity" tab directly at:
<http://myhost.mydomain.com:6401/AnalyticsPlatform/Status?page=summary>.

Summary	Value
Started	March 25, 2009 1:43:53 PM EDT
Status	Waiting for clients
SAS Workspace Sessions	0
Total Users	1
Registered Applications	1
Loaded Applications	1

User ID	Name	Session Started	Client Address
sasadm@saspw	sasadm	March 25, 2009 2:00:54 PM EDT	

No SAS workspace sessions are currently active.

The **Configuration** tab displays the SAS Analytics Platform Server's configuration information. You can access the "Configuration" tab directly at: <http://myhost.mydomain.com:6401/AnalyticsPlatform/Status?page=configuration>.

The screenshot shows the SAS Analytics Platform Server Status page with the Configuration tab selected. The page title is "Server Status • March 25, 2009 2:14:12 PM EDT". The Configuration section is titled "Configuration" and contains the following information:

Analytics Platform Server	
Host	[Redacted]
Root location	/C:/SAS/EMiner/Lev2/AnalyticsPlatform/
Version	105000.107.0.20090317190000_d2apc15
Discovery Enabled	No
RMI Registry Port	6412
RMI Port	6422
RMI SSL Port	6432
RMI security	The following RMI services are secured:
SAS Foundation	
Foundation Services Deployment	Analytics Platform 1.5 Services - myhost - Lev2
Metadata Host	[Redacted]
Metadata Port	8562
SAS Metadata Repository	Foundation
Authentication Domain	DefaultAuth

The Environment section is titled "Environment" and contains the following information:

Operating System	Windows XP
Operating System	5.1
Version	
Java version	1.5.0_12-b04

Monitoring Using a JMX Console

[Java Management Extensions](#) (JMX) allow Java applications to be monitored and managed via:

- management beans (MBeans) which allow one to monitor the JVM's use of threads, memory, etc.
- services which provide a management bean, such as the SAS Foundation Services. They may also be managed by invoking operations from a third party JMX console.

JMX Consoles

There are many third party JMX consoles which may be used. Some of the more popular JMX consoles are:

- [jconsole](#)
- [MC4J](#)
- [jManage](#)

Sun discusses [Monitoring and Management using JMX](#) using its jconsole which is available in the JDK's bin directory. Local JMX access is enabled by defining the property "com.sun.management.jmxremote." A `jmx.config` file, located in the directory where the SAS Analytics Platform is configured, is provided to enable one to configure JMX properties. One may edit the `jmx.config` file to specify additional properties which may be required to configure the application to be accessed by a particular third party JMX console. Refer to the third party JMX console's documentation for details which describe how to configure the application to be accessed by the JMX console.

The JRE may be monitored using the jconsole to observe:

- Threads
- Memory
- Classes
- MBeans
- VM

The following SAS Foundation Services also provide MBeans which allow one to manage runtime state:

- Discovery Service MBean
 - List services which can be discovered
 - Get details for discoverable services
- Logging Service MBean
 - List logging contexts
 - Change a logging context's priority
 - Get count of allocated logging contexts
- Session Service MBean
 - Get a summary of all active session contexts
 - Quiesce/resume the Session Service
 - Determine if the Session Service is quiesced
 - Destroy all or a specific session context
- User Service MBean
 - Get a list of active user contexts
 - Destroy a user context
 - Get a count of authenticated users
 - Get a count of users who failed to authenticate
 - Get failure details
 - Get the date the User Service was started

Java Web Start Client Considerations

Automatic downloads of client files can be accomplished using Java Web Start (JWS). Deploying client files in this manner eliminates the need to install the client application manually on each desktop machine. It also eliminates the possibility of the client application version not matching the server version. When you launch the application in this manner, all of the required JAR files are automatically downloaded to the desktop. You might be prompted a few times for security purposes and asked if you want to create a desktop icon. If a new version is installed on the server, then the updated version automatically installs before the client application is invoked.

Clients for SAS Enterprise Miner, SAS Forecast Studio, and SAS Model Manager can be launched by the Java Web Start facilities enabled within the SAS Analytics Platform Server. The clients can be launched by clicking their link on the **Applications** tab found in the Applications section of the SAS Analytics Platform Server Status page.

The clients can also be launched by a direct URL reference or shortcut:

- SAS Enterprise Miner
<http://server.domain.com:port/EnterpriseMiner/main.jnlp>
- SAS Forecast Studio
<http://server.domain.com:port/Forecasting/main.jnlp>
- SAS Model Manager
<http://server.domain.com:port/ModelManager/main.jnlp>

SAS Analytics Platform family product Java Web Start clients use the publicly installed Java JRE library, which must be at version 1.5 or later. These products are available from Sun Microsystems, IBM, or Hewlett Packard, depending on the platform of the client workstation. The most common workstation is the Windows platform, and Sun's Java install automatically updates the Java Web Start launching mechanism within Web browsers installed on Windows.

Windows Service Administration

One may start, stop, restart and remove the SAS Analytics Platform Server's Windows Service using the launch script. Windows shortcuts, if installed, may also be used to start and stop the SAS Analytics Platform's Windows Service. One should use these scripts if there is ever a need to stop and start the SAS Analytics Platform Server.

The SAS Analytics Platform Server's Windows Service will be stopped and removed when the SAS Deployment Manager unconfigures the SAS Analytics Platform Server.

Windows Service Action	Command Line
Start	AnalyticsPlatform.bat start
Stop	AnalyticsPlatform.bat stop
Restart	AnalyticsPlatform.bat restart
Uninstall	AnalyticsPlatform.bat remove

Troubleshooting

This section provides instructions to help troubleshoot.

The SAS Analytics Platform Server's log file, `AnalyticsPlatform.log`, is located in its "Logs" directory. If the SAS Analytics Platform Server has been configured to be started as a Windows Service, then the "Logs" directory will also contain the service's log file `wrapper.log`.

Remote clients are unable to connect when the SAS Analytics Platform Server is running on a multi-homed machine

If the SAS Analytics Platform Server's machine is multi-homed, then one should designate the IP address of the host to which the client will connect by setting the "java.rmi.server.hostname" property as an additional JVM option when configuring the SAS Analytics Platform Server.

1. Unconfigure the SAS Analytics Platform Server using the SAS Deployment Manager (see "Unconfiguring the SAS Analytics Platform Server using the SAS Deployment Manager" on page 12).
2. Configure the SAS Analytics Platform Server using the SAS Deployment Wizard and specify "`-Djava.rmi.server.hostname=ip-address-of-this-machine`" in the Additional JVM Options text field.

For example:

```
-Djava.rmi.server.hostname=10.40.12.43
```

The SAS Analytics Platform Server shuts down when your UNIX session is terminated

Depending on the protocol in which your UNIX session was established, you may find that the SAS Analytics Platform Server shuts down when your UNIX session is terminated. In this case, it may be necessary to unset your display prior to starting the SAS Analytics Platform Server.

For example:

```
unset DISPLAY  
./AnalyticsPlatform.sh start
```

The Forecast Studio application log on window fails to locate the SAS Analytics Platform Server using the "Find Servers" feature

This can be caused by the following factors:

- The multicast discovery service was not enabled in the SAS Analytics Platform Server.
 1. Unconfigure the SAS Analytics Platform Server using the SAS Deployment Manager (see "Unconfiguring the SAS Analytics Platform Server using the SAS Deployment Manager" on page 12).
 2. Use the SAS Deployment Wizard to configure the SAS Analytics Platform Server and enable its multicast discovery servers by selecting the **Enable automatic discovery of the server via multicasting** checkbox.
 3. Start the SAS Analytics Platform Server if it was not automatically started when the SAS Deployment Wizard configured the server.

- The SAS Forecast Studio client application is using the wrong multicast address and port for the SAS Analytics Platform Server's multicast discovery service. By default, the SAS Forecast Studio client is configured to use the IPv4 multicast address "239.192.65.80" and "6441" for the multicast port which corresponds to the defaults for a Lev1 SAS Analytics Platform Server. If the SAS Analytics Platform Server has been configured to use a different multicast address or port, then the SAS Forecast Studio client's launch configuration must be updated to specify the multicast address/port using a system property named "sas.apcore.logon.netaid.multicast.servers."

For example:

```
-Dsas.apcore.logon.netaid.multicast.servers="239.192.65.80:7777"
```

If multiple SAS Analytics Platform Servers need to be found and they were configured to use a multicast address/port other than "239.192.65.80:6441," then the SAS Forecast Studio client's launch configuration must be updated to specify the multicast address/ports of the multicast discovery services for the configured SAS Analytics Platform Servers.

For example:

```
-Dsas.apcore.logon.netaid.multicast.servers="239.192.65.80:6441,  
239.192.65.80:6442"
```

- If the client is running from behind a firewall, then the Find Servers feature will fail. Enter the server location in the **Server** field and click **Log On**.
- If multicasting is disabled, then the Find Servers feature will fail. Enter the server location in the **Server** field and click **Log On**.

Note: The SAS Forecast Server 3.1: Administrator's Guide contains information about how to configure SAS Forecast Server. For a copy of this guide, contact your SAS consultant or SAS Technical Support.

When clicking the Java Web Start launch link, a prompt appears to save a jnlp file, instead of launching the application

If the Java plug-in is not installed in the Web browser, then one specify that the Java Web Start executable should be used to open the jnlp file which specifies the client application's launch configuration. Therefore, you have the following options:

Using Internet Explorer (Windows Platform)

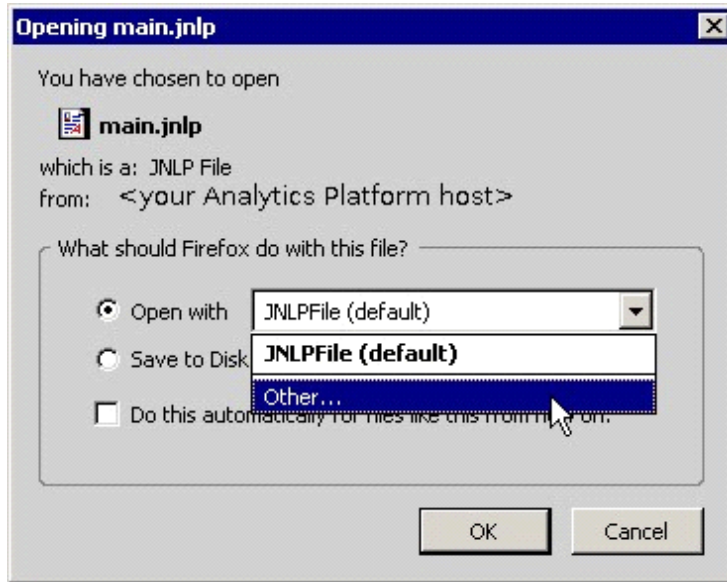
1. Install the full Java runtime environment by visiting <http://www.java.com> to install the Java plug-in into your Web browser.
2. Restart Internet Explorer and revisit the Java Web Start link for your application.

Using Firefox (All Platforms)

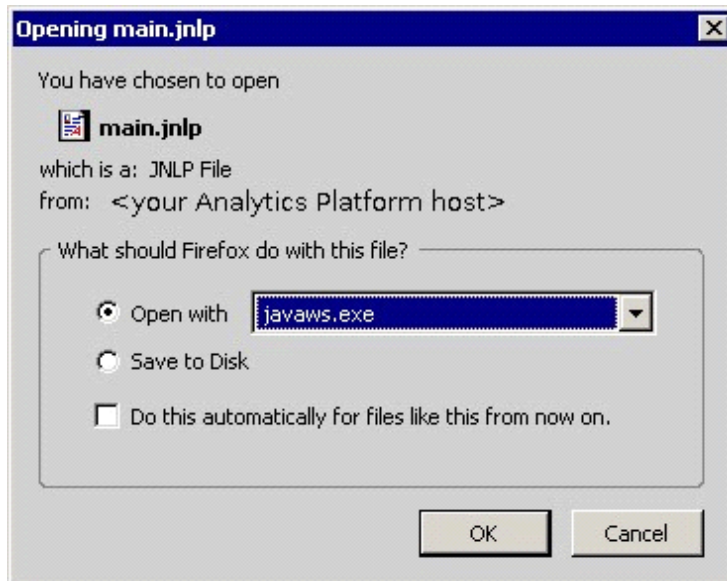
Use a different browser, such as Mozilla Firefox. Firefox allows you to associate an application with a file type. To configure Firefox for Java Web Start, do the following:

1. Install Firefox by visiting <http://www.mozilla.com>.
2. Start Firefox and click the Java Web Start launch link for your application.

- When prompted with choices on what to do with the file, select **Other** from the **Open with** drop-down list.



- Navigate to where the SAS private JRE has been installed. For example, C:\Program Files\SAS\Shared Files\JRE\1.5.0_12\bin and select the Java Web Start executable **javaws.exe**.





THE
POWER
TO KNOW.

support.sas.com

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