

Configuration Guide

Configuring IBM WebSphere Application Server 7.0 for Web Authentication with SAS® 9.2 Web Applications

Configuring the System for Web Authentication

This document explains how to configure Web authentication with IBM WebSphere Application Server for the SAS Web applications. Before using this document, review “Web Authentication” in *SAS 9.2 Intelligence Platform: Security Administration Guide* to understand and verify that Web authentication is the appropriate choice for your environment.

The default security mechanism for SAS Web applications is to authenticate against the authentication provider of the SAS Metadata Server. An alternative authentication mechanism, Web authentication, is to configure WebSphere Application Server to authenticate against a user registry, such as an LDAP server, and to configure SAS Web applications to trust the authentication that WebSphere Application Server performs.

Here are the high-level steps that you must perform to configure Web authentication.

1. Update the `login.config` file in your SAS configuration directory so that it contains the necessary references to the web domain.
2. Add information about security constraints, an authentication method, and security roles to the SAS Logon Manager application. When you reinstall the application, provide a security role to user or group mapping to indicate which users have permission to access the application.
3. Copy SAS JAR files to the WebSphere Application Server installation.
4. Using the IBM WebSphere Integrated Solutions Console (known as the *administrative console*), update information about the login modules that the server uses for authentication and authorization when the system is configured for Web authentication. You must modify information for some login modules and add information for others.
5. Configure the SAS Remote Services application so that its classpath includes the location of the WebSphere Application Server classes that represent Java Authentication and Authorization Service (JAAS) principals. Logon Manager retrieves the current `Subject` from WebSphere Application Server and passes it to Remote Services.
6. Ensure that the correct JRE is used.
7. Restart Remote Services and WebSphere Application Server. You can also restart the Metadata Server.
8. Verify the configuration. You might need to create a web authentication domain and add new accounts in that domain for users.

Before Starting This Configuration

Before you try to configure Web authentication, you must enable WebSphere Application Server application security and have already configured the users and a user registry, such as an LDAP server for authentication. To enable the security, use the WebSphere administrative console and follow these steps:

1. Select **User account repository ► Federated repositories**.
2. Click **Set as current**.
3. Select **Configure ...** On the configure screen, type in your admin user name in **Primary administrative user name** and accept all other defaults.
4. Click **Ok** to go to the next screen.
5. Type in the **admin password** and click **Ok**.
6. Select **Global security**.
7. Check **Enable administrative security** and **Enable application security**.
8. Optionally, Use Java 2 security to restrict application access to local resources.
9. Click **Ok** and save the changes.

You must restart your DMGR for the changes to take effect.

To configure WebSphere users for authentication, follow these steps:

- Select **Manage Users**.
- Select **Create ...**
- Type in **userID**, such as **sasdemo**, and fill in other fields. Click **Create**. You can create as many userIDs as needed.

For more information about configuring a user registry, see "Chapter 2: Configuring the user registry" in the *WebSphere Application Server V7.0 Security Guide*. If you have **DefaultApplication.ear** installed and its role "All Role" is mapped to a user or group, you can verify the configuration by accessing a Web application on the server. You can use *snoop* by opening a Web browser to <http://HOSTNAME:9080/snoop>. If WebSphere Application Server is configured correctly, WebSphere Application Server asks you for credentials that are stored in the user registry.

Before beginning this configuration, be sure that the WebSphere Application Server that is hosting SAS Web applications is running. At the end of the procedure, you must start or restart Remote Services and all WebSphere Application Server processes.

Update the login.config Configuration File

Update the *SAS-config-dir/Lev1/Web/Common/login.config* file so that the **aliasdomain** property is set to **web**. The file content should resemble this example:

```
PFS {
  com.sas.services.security.login.OMILoginModule  required
    "host"="metadata-server-host"
    "port"="8561"
    "repository"="Foundation"
    "domain"="DefaultAuth"
    "trusteduser"="sastrust@saspw"
    "trustedpw"="encoded-password"
    "aliasdomain"="web"
```

```

        "debug"="false";
    };
    SCS {
        com.sas.services.security.login.OMILoginModule required
        "host"=" metadata-server-host "
        "port"="8561"
        "repository"="Foundation"
        "domain"="DefaultAuth"
        "trusteduser"="sastrust@saspw"
        "trustedpw"="encoded-password"
        "aliasdomain"="web"
        "holdopenconnection"="true";
        "debug"="false";
    };

```

The default value of aliasdomain is DefaultAuth.

Modify Logon Manager

To make the necessary changes to Logon Manager, you must edit the web.xml file. The web.xml file is located in the WEB-INF directory. To extract and edit the file, follow these steps.

1. Use the WebSphere administrative console to stop and uninstall SAS Web Infrastructure Platform applications (SASWebInfrastructrePlatformApplications9.2). You need to make changes to the corresponding *SAS-config-dir/Levl/Web/Staging/sas.wip.apps9.2.ear* (EAR) file.
2. Extract the sas.wip.apps9.2.ear file so that you can access the WEB-INF directory for Logon Manager.
 - a. In a temporary directory, extract the EAR file. You can use the jar command to do this:


```
jar xvf sas.wip.apps9.2.ear
```

File sas.svcs.logon.war is available in the extracted directory.
 - b. In a second temporary directory, extract sas.svcs.logon.war. You now have access to the Logon Manager WEB-INF directory.
3. Edit the file web.xml in the WEB-INF directory to add information about security constraints, an authentication method, and security roles. For example, just above the closing </web-app> tag, you might add these elements:

```

<security-constraint>
  <web-resource-collection>
    <web-resource-name>All resources</web-resource-name>
    <url-pattern>/*</url-pattern>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
  </web-resource-collection>

  <auth-constraint>
    <role-name>SASWebUser</role-name>
  </auth-constraint>
</security-constraint>

```

```

<login-config>
  <auth-method>BASIC</auth-method>
  <realm-name>myrealm</realm-name>
</login-config>

<security-role>
  <role-name>SASWebUser</role-name>
</security-role>

```

In this example, all pages are protected and only users who have been assigned the SASWebUser role can access them.

- Before you rebuild the WAR and EAR files, change directories from the WEB-INF directory to the lib directory inside it, and copy these JAR files to a temporary location:

```

sas.core.jar
sas.oma.omi.jar
sas.security.sspi.jar
sas.svc.connection.jar
sas.svc.sec.login.jar
sas.svc.sec.login.websphere.jar

```

Note: This step is not part of updating SAS Web Infrastructure Platform applications. However, it is preparation for a later step in configuring Web authentication.

- Rebuild the WAR and EAR files. You can use the jar command to create these files:

```

jar cvf sas.svcs.logon.war *
jar cvf sas.wip.apps9.2.ear *

```

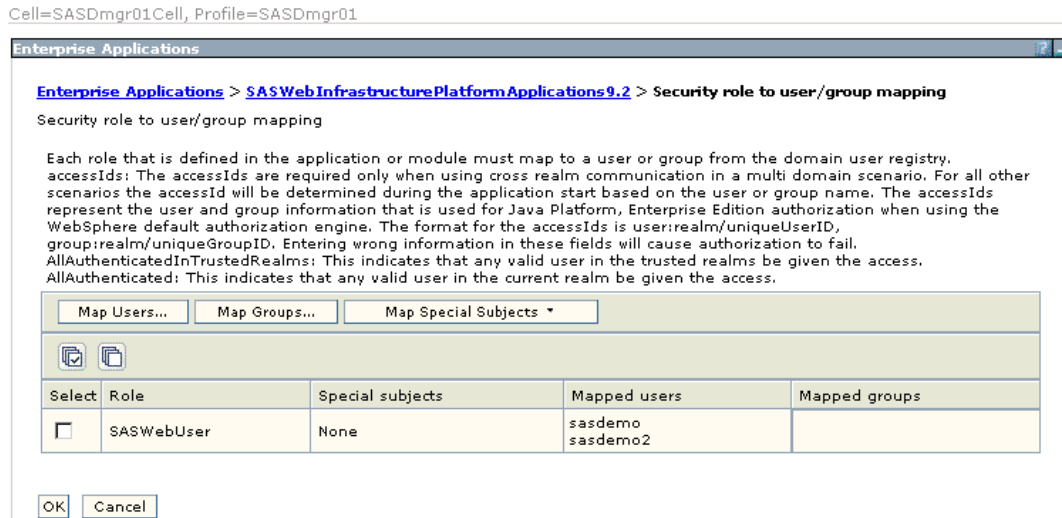
- Copy the EAR file to your staging directory. However, do not overwrite the original EAR file unless you already made a backup copy.

Reinstall the SAS Web Infrastructure Platform Applications EAR File

Use the WebSphere administration console to reinstall the EAR file and then map the role SASWebUser to users or groups. To reinstall the SAS Web Infrastructure Platform applications, follow these steps.

- Select **Applications ► Enterprise Applications**.
- Click **Check: SASWebInfrastructurePlatformApplications9.2.ear**. Select **Uninstall**. Save the results.
- Click **Install**.
- 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**.
- Finish running the installation wizard by accepting all defaults.
- From the **Enterprise Applications** page, select the newly installed application.
- On the page for that application, set the class-loading behavior for the EAR file:
 - Click **Class loading and update detection**.
 - 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**.)

8. For each WAR file in the EAR file, set the class-loader behavior:
 - a. On the main page for configuring the application (EAR), click **Manage Modules**.
 - b. Click the name of the Web module (WAR file) that you want to configure.
 - 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**.
 - d. Click **OK** to close the Manage Module page.
9. Set the startup order by selecting **Startup behavior**, set the **Startup order** value to 3, and then click **OK**.
10. Set the security mapping by clicking the **Security role to user/group mapping** link. Map the role SASWebUser to users. If this option is not appropriate for your site, then consider mapping the role to groups that are defined in your user registry. The following figure shows an example of mapping the role to the group **sasusers** that is defined in the user registry.



Note: If you do not see the Role that you entered in the web.xml file, then check that the correct EAR file is deployed and that the changes to the web.xml file are correct.

Copy SAS JAR Files to the WebSphere Installation

The “Modify Logon Manager” section instructed you to copy SAS JAR files to a temporary location. Copy those files now to the `WAS_INSTALL_ROOT/lib/ext` directory.

Make Changes to Application JAAS Logins

Using the WebSphere administrative console, change the JAAS application logins for PFS and SCS. To change application logins, follow these steps.

1. Select **Security ► Global Security ► Java Authentication and Authorization Service ► Application logins**.
2. For the PFS alias, make these changes to the first login module.

- a. Write down the properties that are used for this module, as shown in the following display:

Select	Name	Value
<input type="checkbox"/>	host	sdw01.na.sas.com
<input type="checkbox"/>	port	8561
<input type="checkbox"/>	repository	Foundation
<input type="checkbox"/>	domain	DefaultAuth
<input type="checkbox"/>	trusteduser	sastrust@sas pw
<input type="checkbox"/>	trustedpw	{sas002}1D3211221C1104E218FEC4002945FBA308AB65D4
<input type="checkbox"/>	debug	false

- b. Delete the module.
- c. Create a new module with a class name of `com.sas.services.security.login.TrustedLoginModule`.
- d. Add the properties that you recorded and a new custom property with the following values:
- Name:** `aliasdomain`
Value: `DefaultAuth`
- Note:* If you chose an authentication domain value other than **DefaultAuth** when you ran the SAS Deployment Wizard, then use the value you chose.
- e. Change the value of the custom property domain from **DefaultAuth** to **web**.
- f. Make sure its order is **1** (one) and authentication strategy is **Sufficient**.
3. For the **scs** alias, change the properties that are associated with the login module `com.sas.services.security.login.OMILoginModule`.
4. Add a new custom property with these values.

Name: `aliasdomain`
Value: `web`

Add a Login Module to the System JAAS Login WEB_INBOUND

Using the WebSphere administrative console, assign a new JAAS login module to the WEB_INBOUND JAAS alias.

1. Select **Security ► Global Security ► Java Authentication and Authorization Service ► System logins ► WEB_INBOUND ► JAAS login modules**.
2. Click **New**, provide this information, and click **OK**.
Module class name:
`com.sas.services.security.login.websphere.WSTrustedLoginModule`
Authentication strategy: `OPTIONAL`
3. Select the module that you just created and then click the **Custom Properties** link.
4. On the **Custom properties** page, for each of these name-value pairs, click **New**, enter the name-value pair, and click **OK**.

Name: aliasdomain
 Value: DefaultAuth

 Name: debug
 Value: false

 Name: domain
 Value: web

 Name: host
 Value: metadata-server-host

 Name: port
 Value: 8561 (or nondefault port)

 Name: repository
 Value: Foundation

 Name: trustedpw
 Value: encoded-password (for sastrust)

 Name: trusteduser
 Value: sastrust@saspw

Set the CLASSPATH for the Remote Services JVM

Modify the classpath for Remote Services so that the Java Virtual Machine (JVM) can locate the WebSphere Application Server classes that it needs when it starts. These JAR files are required and contain classes that represent JAAS principals that the JVM acquires from your WebSphere Application Server:

```

WAS_INSTALL_ROOT/plugins/com.ibm.ffdc.jar
WAS_INSTALL_ROOT/plugins/com.ibm.ws.admin.core.jar
WAS_INSTALL_ROOT/plugins/com.ibm.wsfp.main.jar
WAS_INSTALL_ROOT/plugins/com.ibm.ws.runtime.jar
WAS_INSTALL_ROOT/lib/bootstrap.jar
WAS_INSTALL_ROOT/plugins/com.ibm.ws.emf.jar
WAS_INSTALL_ROOT/plugins/org.eclipse.emf.ecore.jar
WAS_INSTALL_ROOT/plugins/org.eclipse.emf.common.jar
WAS_INSTALL_ROOT/lib/j2ee.jar
  
```

Important: You must enter the classpath all on one line, without spaces or carriage returns.

Windows

For Windows machines, the RemoteServices.bat script should resemble the following example:

```

:start2
  start "SAS Remote Services" "%JAVA_JRE_COMMAND%" ^
    -classpath "%CLASSPATH%" ^
    -Dsas.ext.config="D:\Program
Files\SAS\SASFoundationServices\9.2\sas.java.ext.config" ^
    -Djava.system.class.loader=com.sas.app.AppClassLoader
-Dsas.app.launch.config="%PICKLIST%" ^
    -Dsas.app.repository.path="%SASVJR_REPOSITORYPATH%" ^
    -Dsas.app.class.path="%REMOTESERVICESDIR%
c:\WebSphere\AppServer\plugins\com.ibm.ffdc.jar;
  
```

```

c:\WebSphere\AppServer\lib\bootstrap.jar;
c:\WebSphere\AppServer\plugins\com.ibm.ws.admin.core.jar;
C:\WebSphere\AppServer\plugins\com.ibm.wsfp.main.jar;
C:\WebSphere\AppServer\plugins\com.ibm.ws.runtime.jar;
C:\WebSphere\AppServer\plugins\com.ibm.ws.emf.jar;
C:\WebSphere\AppServer\plugins\org.eclipse.emf.ecore.jar;
C:\WebSphere\AppServer\plugins\org.eclipse.emf.common.jar;
C:\WebSphere\AppServer\lib\j2ee.jar" ^
-Djava.net.preferIPv4Stack=true -Djava.net.preferIPv6Addresses=false
-Dmulticast_udp_ip_ttl=1 ^ -Dsas.vjr.dir="%SASVJR_REPOSITORYPATH%"
-Dsas.lev.dir="%LEVEL_ROOT%" -Dsas.home.dir="%SAS_HOME%" ^ -
Dsas.services.information.types.path="D:\Program
Files\SAS\SASPlatformObjectFramework\9.2\plugins" ^
-Dsas.vm.identifier=Lev3:5093 ^ -Xms128m -Xmx128m -XX:+UseTLAB
-XX:+UseConcMarkSweepGC -XX:+DisableExplicitGC
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.awt.headless=true -Xss256k
-XX:NewSize=16m -XX:MaxNewSize=16m -XX:PermSize=64m -XX:MaxPermSize=64m ^
com.sas.framework.services.bootstrap.SASRemoteServices

goto end

```

If Remote Services is started as a Windows service, then you must make the same modification to the *SAS-config-dir\Lev1\Web\Applications\RemoteServices\wrapper.conf* file. The part of the file that sets the classpath should resemble the following example:

```

# This numbering starts at the endpoint of the including wrapper.conf
wrapper.java.additional.3=-Dsas.app.class.path="C:\SAS\Config\
Lev3\Web\Applications\RemoteServices\C:\IBM\WebSphere\AppServer\plugins\
com.ibm.ffdc.jar;C:\IBM\WebSphere\AppServer\plugins\com.ibm.ws.admin.core
.jar;C:\IBM\WebSphere\AppServer\plugins\com.ibm.wsfp.main.jar;C:\IBM\
WebSphere\AppServer\plugins\com.ibm.ws.runtime.jar;C:\IBM\WebSphere\
AppServer\lib\bootstrap.jar;C:\IBM\WebSphere\AppServer\plugins\com.ibm.
ws.emf.jar;C:\IBM\WebSphere\AppServer\plugins\org.eclipse.emf.ecore.jar;
C:\IBM\WebSphere\AppServer\plugins\org.eclipse.emf.common.jar;C:\IBM\
WebSphere\AppServer\lib\j2ee.jar"

```

UNIX

For a UNIX machine, the classpath property is set beneath the start2 tag and the changes should resemble the following example:

```

start2)
"$JAVA_JRE_COMMAND" -Dsas.ext.config="/opt/SAS/SASFoundation
Services/9.2/sas.java.ext.config" \ -classpath "$CLASSPATH" \
-Djava.system.class.loader=com.sas.app.AppClassLoader \
-Dsas.app.launch.config="$MERGER_PICKLIST" \
-Dsas.app.repository.path="$SASVJR_REPOSITORYPATH" \
-Dsas.app.class.path="$REMOTESERVICESDIR" \
com.sas.framework.picklist.PicklistMerger \
-primary"$PRIMARY_PICKLIST" \ "$PICKLIST" \ "$SECONDARY_PICKLIST1"
\ "$SECONDARY_PICKLIST2" cd $REMOTESERVICESLOGSDIR nohup
"$JAVA_JRE_COMMAND" -Dsas.ext.config="/opt/SAS/

```



```

SASFoundationServices/9.2/sas.java.ext.config" \
-classpath"%CLASSPATH" \
9
-Djava.system.class.loader=com.sas.app.AppClassLoader \
-Dsas.app.launch.config="$PICKLIST" \
-Dsas.app.repository.path="$SASVJR_REPOSITORYPATH" \
-Dsas.app.class.path="$REMOTESERVICESDIR:/opt/IBM/WebSphere/AppServer/
plugins/com.ibm.ffdc.jar:/opt/IBM/WebSphere/AppServer/plugins/com.
ibm.ws.admin.core.jar:/opt/IBM/WebSphere/AppServer/plugins/com.ibm.
wsfp.main.jar:/opt/IBM/WebSphere/AppServer/plugins/com.ibm.ws.runtime.
jar:/opt/IBM/WebSphere/AppServer/lib/bootstrap.jar:/opt/IBM/WebSphere/
AppServer/plugins/com.ibm.ws.emf.jar:/opt/IBM/WebSphere/AppServer/
plugins/org.eclipse.emf.ecore.jar:/opt/IBM/WebSphere/AppServer/plugins
/org.eclipse.emf.common.jar:/opt/IBM/WebSphere/AppServer/lib/j2ee.jar" \
-Djava.net.preferIPv4Stack=true -Djava.net.preferIPv6Addresses=false
-Dmulticast_udp_ip_ttl=1 \ ...

```

Use the Correct JRE

Open C:\SAS_Home\wrapper.conf and ensure that a Java 6 JRE is specified, for example:

```
wrapper.java.command=C:\Program Files (x86)\Java\jre6\bin\java.exe
```

Restart Remote Services and WebSphere Application Server

At this point, restart Remote Services and the WebSphere Application Server that is hosting SAS Web applications. After restart, when you log in to a SAS Web application, WebSphere Application Server handles authentication. You do not see the Logon Manager Web page; instead, a dialog box prompts you for your user ID and password. WebSphere Application Server authenticates the user ID and password that you enter against the user registry, such as an LDAP server, that you configured previously. You might not need to re-enter your user ID and password each time you start a SAS Web application because credentials are cached.

Set the WebApp.AuthDomain Property

Some applications such as SAS Enterprise Guide need to know the authentication domain that is associated with the SAS Web applications. Follow these steps.

1. Start SAS Management Console and connect to the SAS Metadata Server.
2. Select **Application Management ► Configuration Manager ► SAS Application Infrastructure**.
3. Right click **SAS Application Infrastructure** and select **Properties**.
4. Select **Advanced**.
5. Click **Add**.
6. Select **Property Name**, enter WebApp.AuthDomain.
7. Select **Property Value**, enter web.
8. Click **OK** until you are out of the dialogs.

Log On to Verify the Web Authentication Configuration

If your site was migrated from a previous SAS release and has user IDs and authentication domains already registered in metadata, try logging on to a SAS Web application such as SAS Web Report Studio.

Otherwise, follow these steps to test and confirm that Web authentication is properly configured.

1. Use SAS Management Console to create an authentication domain named web.
 - a. Right-click **User Manager** and select **Authentication Domains**.
 - b. Click **New**, enter web in the **Name** field, and click **OK**.
2. Choose a trial user ID that exists in your user registry. Use SAS Management Console to create a user definition for the user in the web authentication domain. Do not enter a password for the account.
3. Try logging on to a SAS Web application with the user ID.

If the log-on attempt fails, view the SAS Metadata Server log. Look for the format of the user ID that was used in the log-on attempt. Use SAS Management Console to modify the user definition so that the user account in the web authentication domain matches the user ID in the log. While you are troubleshooting, do not enter a password in the user definition because it has no effect on Web authentication. Also, do not try logging on with an internal account such as sasadm@saspw.

Note: As part of Web authentication, the user ID but not the password is checked against the user accounts that are stored in the SAS Metadata Repository. The user ID used to authenticate with the user registry must match exactly the user ID string found on the SAS Metadata Server for authentication to succeed. For example, if joe is the user ID in your user registry, the exact user ID string “joe” must also be found in the SAS Metadata Repository without a prefixed domain name.

Recommended Reading

As of December 2012:

- IBM Corporation, 2009. WebSphere Application Server V7.0 Security Guide. ibm.com/Redbooks. Available at <http://www.redbooks.ibm.com/redbooks/pdfs/sg247660.pdf>
- SAS Institute, Inc., 2009. *SAS 9.2 Intelligence Platform: Security Administration Guide*. Cary, NC: SAS Institute, Inc. Available at <http://support.sas.com/documentation/cdl/en/bisecag/61133/PDF/default/bisecag.pdf>.

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 © 2012 SAS Institute Inc., Cary, NC, USA. All rights reserved.