Configuration Guide

Configuring IBM WebSphere Application Server 6.1 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, you must secure WebSphere Application Server (see Chapter 3, "Administrative security" in *WebSphere Application Server V6.1 Security Handbook*). Also, you need to review "Web Authentication" in *SAS 9.2 Intelligence: 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.

- Update the `login.config` file in your SAS configuration directory so that it contains the necessary references to the `web` domain.
- 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.
- Copy SAS JAR files to the WebSphere Application Server installation.
- 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.
- 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.
- Restart Remote Services and WebSphere Application Server.
- 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 have already configured a user registry, such as an LDAP server, in WebSphere Application Server and you must enable WebSphere Application Server application security. For more information about configuring a user registry, see "Chapter 2: Configuring the user registry" in the *WebSphere Application Server V6.1 Security Handbook*. Verify the configuration by accessing a Web application on the server such as 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 its `web.xml` file. The `web.xml` file is located in its `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 (SASWebInfrstructrePlatformApplications9.2). You need to make changes to the corresponding *SAS-config-dir*`/Lev1/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.

Note: This example is for BASIC authentication. For FORM authentication, see Appendix: FORM Authentication.

4. 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.

5. 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 *
   ```

6. 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

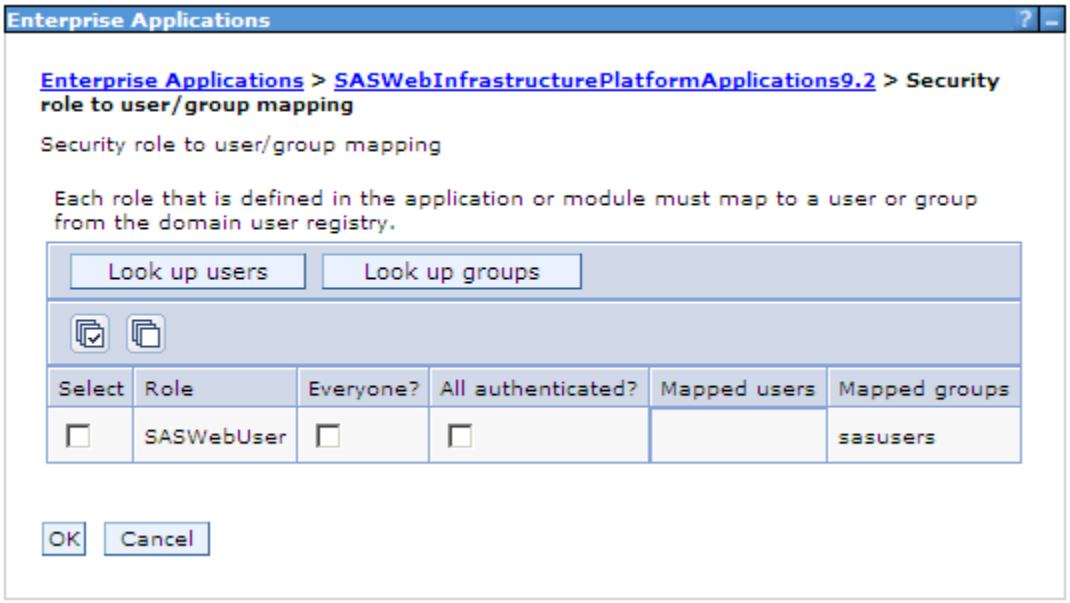
Use the WebSphere administration console to reinstall the EAR file and then map the role `SASWebUser` to users or groups. To reinstall the SAS Web Infrastructure Platform applications, follow these steps.

1. Select **Applications** > **Enterprise Applications**.
2. Click **Install**.
3. On the **Specify the EAR, WAR, JAR, or SAR module to upload and install** page, select the **Local file system** or **Remote file system** radio button, and then browse to the location of the EAR file. Select the EAR file and click **OK**. Click **Next**.
4. Finish running the installation wizard by accepting all defaults.
5. From the **Enterprise Applications** page, select the newly installed application.
6. On the page for that application, set the class-loading behavior for the EAR file:
   a. Click **Class loading and update detection**.
   b. On the **Class loader** page, set **Class loader order** to **Classes loaded with application class loader first**. (Leave the **WAR class loader policy** set to **Class loader for each WAR file in application**.)

   Note: To work around a defect in the administration console, set the polling interval to zero (0) before you click **OK**.

7. For each WAR file in the EAR file, set the class-loader behavior:
   a. On the main page for configuring the application (EAR), 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.
8. Set the startup order by selecting **Startup behavior**, set the **Startup order** value to 3, and then click **OK**.
9. Set the security mapping by clicking the **Security role to user/group mapping** link. Map the role `SASWebUser` to `All authenticated`. If this option is not appropriate for your site, then consider mapping

the role to users or 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** > **Secure administration, applications, and infrastructure** > **Java Authentication and Authorization Service** > **Application logins**.
2. For the `PFS` alias, make these changes to the first login module.
   a. Change the name of the Module class name from
   `com.sas.services.security.login.OMILoginModule`

   to

   `com.sas.services.security.login.TrustedLoginModule`.

   b. Add a new custom property with these 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.
   c. Change the value of the custom property `domain` from `DefaultAuth` to `web`.

3. For the SCS alias, change the properties associated with the login module com.sas.services.security.login.OMILoginModule. 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** > **Secure administration, applications, and infrastructure** > **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.ws.runtime_6.1.0.jar

WAS_INSTALL_ROOT/lib/bootstrap.jar

WAS_INSTALL_ROOT/plugins/com.ibm.ws.emf_2.1.0.jar

WAS_INSTALL_ROOT/plugins/org.eclipse.emf.ecore_2.2.1.v200609210005.jar

WAS_INSTALL_ROOT/plugins/org.eclipse.emf.common_2.2.1.v200609210005.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:\base\WebSphere\AppServer\
plugins\com.ibm.ws.runtime_6.1.0.jar;c:\base\WebSphere\AppServer\lib\
bootstrap.jar;c:\base\WebSphere\AppServer\plugins\com.ibm.ws.emf_2.1.0.jar;
c:\base\WebSphere\AppServer\plugins\org.eclipse.emf.ecore_2.2.1.v200609210005.jar;c
:\base\WebSphere\AppServer\plugins\org.eclipse.emf.common_2.2.1.v200609210005.jar;c
:\base\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:\base\WebSphere\AppServer\
plugins\com.ibm.ws.runtime_6.1.0.jar;c:\base\WebSphere\AppServer\lib\
bootstrap.jar;c:\base\WebSphere\AppServer\plugins\com.ibm.ws.emf_2.1.0.jar;
c:\base\WebSphere\AppServer\plugins\org.eclipse.emf.ecore_2.2.1.v200609210005.jar;c
:\base\WebSphere\AppServer\plugins\org.eclipse.emf.common_2.2.1.v200609210005.jar;c
:\base\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 "$CLASSPATH"  \
   -Djava.system.class.loader=com.sas.app.AppClassLoader \
   -Dsas.app.launch.config="$PICKLIST" \
   -Dsas.app.repository.path="$SASVJR_REPOSITORYPATH" \
   -Dsas.app.class.path="$REMOTESERVICESDIR:/opt/IBM/WebSphere/AppServer/
   plugins/com.ibm.ws.runtime_6.1.0.jar:/opt/was61/lib/bootstrap.jar:/opt/
   IBM/WebSphere/AppServer/plugins/com.ibm.ws.emf_2.1.0.jar:/opt/IBM/
   WebSphere/AppServer/plugins/org.eclipse.emf.ecore_2.2.1.v200609210005.jar:/opt/I
   BM/WebSphere/AppServer/plugins/org.eclipse.emf.common_2.2.1.v200609210005.jar:/o
   pt/IBM/WebSphere/AppServer/lib/j2ee.jar" \
   -Djava.net.preferIPv4Stack=true -Djava.net.preferIPv6Addresses=false
   -Dmulticast_udp_ip_ttl=1 \
   ...
```

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.

Appendix: FORM Authentication

Use the following instructions to set up a simple FORM authentication with WebSphere 6.1 and SAS 9.2M3.

1. To enable the custom logoff message, follow the instructions at **Sample 36785: Creating a custom message to display when users log off or time-out of the SAS® Business Intelligence Web applications.**

2. Extract the `sas.wip.apps9.2.ear` and `sas.scvs.login.war` files using the instructions in section Modify Logon Manager.
3. Modify the <*login-config*> section in web.xml as shown in the example below. The specification of the <*form-login-page*> and <*form-error-page*> are required, but the associated file names can differ from the example. The files also can be `.jsp` files instead of `.html` files.

```
<login-config>
<auth-method>FORM</auth-method>
<realm-name>Form Auth</realm-name>
<form-login-config>
     <form-login-page>/was_login.html</form-login-page>
     <form-error-page>/form_error.html</form-error-page>
</form-login-config>
</login-config>
```

4. Create a login form and error page file that are referenced in the `web.xml` file. The rest of the page can be formatted per the customer's needs. The ACTION specified in the example is required for successful login with WebSphere. Also use the exact name values in the input fields.

   Login form code example (`was_login.html`):

```
<FORM METHOD=POST ACTION="j_security_check">
<p>
<font size="2"> <strong> Enter user ID and password: </strong></font>
<BR><br>
<strong> User ID</strong> <input type="text" size="20" name="j_username">
<Br>
<strong> Password </strong>  <input type="password" size="20" name="j_password">
<BR>
<BR>
<font size="2">  <strong> And then click this button: </strong></font>
<input type="submit" name="login" value="Login">
</p>
```

   Error page code example (`form_error.html`):

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD HTML 4.0 Transitional//EN">
<html>
<head><title>A Form login authentication failure occurred</head></title>
<body>
Error Message
</body>
</html>
```

5. Save the files in root level of `sas.scvs.login.war`.
6. Modify the `custom_logoff.jsp` file. The following example automatically executes upon logoff, and redirects you back to the login page.

   The ACTION specified in the example is required to invalidate the authenticated WebSphere session. Otherwise, customize to the customer's requirements.

```
<html>
 <body onLoad="submitForm()">
     <FORM METHOD=POST ACTION="ibm_security_logout" NAME="myForm" ID="myForm">
     </form>
```

```
 </body>
<script type='text/javascript'>
     document.myForm.submit();
</script>
</html>
```

7. Rebuild the `.war` file and `.ear` file as described in Step 5 of the Modify Logon Manager.
8. Re-install sas.wip.apps9.2.ear.
9. Restart WebSphere server instance.

Recommended Reading

As of March 2010:

IBM Corporation, 2009. *WebSphere Application Server V6.1 Security Handbook*. ibm.com/Redbooks. Available at http://www.redbooks.ibm.com/abstracts/sg246316.html?Open.

SAS Institute, Inc., 2009. *SAS 9.2 Intelligence Platform: Security Administration Guide.* Cary, NC: SAS Institute, Inc. Available at http://support.sas.com/92administration.