Configuring BEA WebLogic Server for Web Authentication with SAS® 9.2 Web Applications

This document describes how to configure Web authentication with BEA WebLogic for the SAS Web applications. Before you use this document, see "Web Authentication" in the *SAS 9.2 Intelligence Platform: Security Administration Guide* to understand and verify that Web authentication is the appropriate choice for your environment. The user registry for this sample configuration is the embedded LDAP server that is provided with WebLogic. If this is not an appropriate configuration for your site, refer to WebLogic documentation for information about security.

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

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

- Stop the SAS Web Infrastructure Platform applications and the application server.
- Update the `login.config` and the `weblogicLogin.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. You must also specify the security domain that it will use.
- Include SAS JAR files in the system classpath. This step provides the classes for the custom WebLogic authentication provider and Subject objects that SAS creates. These classes must be available to JMX clients such as the WebLogic Scripting Tool as well as the Administration Server and Managed Servers.
- Configure the SAS Remote Services application so that its classpath includes the location of the WebLogic classes that represent Java Authentication and Authorization Service (JAAS) principals. Logon Manager retrieves the current `Subject` from WebLogic and passes it to Remote Services.
- Configure the custom WebLogic authentication provider for SAS Web authentication in the WebLogic Administration Console. Add SAS users to the LDAP server that is embedded in WebLogic. Start the WebLogic Server and deploy SAS Web Infrastructure Platform applications.
- Verify the configuration. You might need to create a `web` authentication domain and add new accounts in that domain for users.

Stop SAS Web Infrastructure Platform Applications and the Application Server

In this procedure you are mostly reconfiguring Logon Manager, which is part of the SAS Web Infrastructure Platform applications (sas.wip.apps9.2.ear). You must stop Logon Manager before you can reconfigure it. Stopping Logon Manager prevents SAS Web applications from working because the applications redirect unauthenticated requests to Logon Manager. To stop Logon Manager and the WebLogic Server, follow these steps:

1. Access the WebLogic Administration Console. In a default configuration that SAS Deployment Wizard performs, the URL is http://hostname:7501/console.
2. Select **Lock & Edit** and click **Deployments**.
3. Select the check box for `sas.wip.apps9.2.ear` and select **Stop** > **Force Stop Now**.
4. Select the check box again and click **Delete**.
5. Select **Environment** > **Servers** > **SASServer1** > **Control**.
6. Select **Shutdown** > **Force Shutdown Now**.

Update the login.config File

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

```
PFS {
  com.sas.services.security.login.OMILoginModule required
    "host"="fully-qualified-hostname-for-metadata-server"
    "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"="fully-qualified-hostname-for-metadata-server"
    "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 the weblogicLogin.config File

Update the application policies for PFS and SCS in *SAS-config-dir*`/Lev1/Web/Common/SASDomain/weblogicLogin.config` file. File content should resemble this example:

```
PFS {
   com.sas.services.security.login.TrustedLoginModule  sufficient
   "host"="fully-qualified-hostname-for-metadata-server"
   "port"="8561"
   "repository"="Foundation"
   "domain"="web"
   "trusteduser"="sastrust@saspw"
   "trustedpw"="encoded-password"
   "aliasdomain"="DefaultAuth"
   "debug"="false";

   com.sas.services.security.login.TrustedLoginModule  optional
   "host"="fully-qualified-hostname-for-metadata-server"
   "port"="8561"
   "repository"="Foundation"
   "domain"="DefaultAuth"
   "trusteduser"="sastrust@saspw"
   "trustedpw"="encoded-password"
   "aliasdomain"="MidtierInternal"
   "debug"="false";
};
SCS {
   com.sas.services.security.login.OMILoginModule  required
   "host"="fully-qualified-hostname-for-metadata-server"
   "port"="8561"
   "repository"="Foundation"
   "domain"="web"
   "trusteduser"="sastrust@saspw"
   "trustedpw"="encoded-password"
   "debug"="false"
   "holdopenconnection"="true";
};
UsernamePassword {
   com.platform.SASLogin.UsernamePasswordLogin  required
   "debug"="false";
};
```

Modify Logon Manager

To make the necessary changes to Logon Manager, you must edit its `web.xml` and `weblogic.xml` files. Both files are located in the `WEB-INF` application directory. To extract and edit the file, follow the steps below.

In the second maintenance release for SAS 9.2, the Rebuild Web Applications option in the SAS Deployment Manager automatically extracts all EAR files and places them in the *SAS-configuration-directory*\Lev1\Web\Staging\exploded directory. If you need to extract an EAR file manually, follow steps 1 and 2. If the .ear file is already extracted, you can skip to step 3 and edit the file that is in *SAS-configuration-directory*\Lev1\Web\Staging\exploded\sas.wip.apps9.2.ear\sas.svcs.logon.war.

1. In a temporary directory, extract the *SAS-config-dir*`/Web/Staging/sas.wip.apps9.2.ear` (EAR) file so that you can access the Logon Manager `WEB-INF` directory. You can use the `jar` command to extract the file:

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

   File `sas.svcs.logon.war` is available in the extracted directory.

2. In a second temporary directory, extract the Logon Manager `sas.svcs.logon.war` (WAR) file:

   ```
   jar xvf sas.svcs.logon.war
   ```

   Within this particular directory, you can now access the Logon Manager `WEB-INF` directory.

3. Edit the `web.xml` file in the `WEB-INF` directory to add information about security constraints, an authentication method, and security roles. For example, 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 can be accessed only by users who have been assigned the `SASWebUser` role.

4. Edit the `weblogic.xml` file in the `WEB-INF` directory. In this sample configuration, you use this file to define a security role assignment that maps the `SASWebUser` role to a principal named `SASUsers`. Later in this example, you create the `SASUsers` group and add members to the group. File content should resemble this example:

   ```
   <?xml version='1.0' encoding='UTF-8'?>
   <weblogic-web-app xmlns="http://www.bea.com/ns/weblogic/90"
   ```

```
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <session-descriptor>
    <cookie-name>SASLogon.sessionID</cookie-name>
    <url-rewriting-enabled>false</url-rewriting-enabled>
  </session-descriptor>
  <context-root>SASLogon</context-root>
  <security-role-assignment>
    <role-name>SASWebUser</role-name>
    <principal-name>SASUsers</principal-name>
  </security-role-assignment>
</weblogic-web-app>
```

You can also map user roles through the WebLogic Administration Console. For information about doing this, see your WebLogic documentation.

5. 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.core.nls.jar

   sas.oma.omi.jar

   sas.security.sspi.jar

   sas.svc.connection.jar

   sas.svc.connection.nls.jar

   sas.svc.sec.login.jar

   sas.svc.sec.logon.nls.jar

   sas.svc.sec.login.weblogic.jar

   sas.svc.sec.login.weblogic.mbean.jar

   sas.svc.sec.login.weblogic.mbean.nls.jar

   sas.svc.sec.login.weblogic.nls.jar

   sas.svcs.security.weblogic.jar

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

6. 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 *
   ```

7. Copy the EAR file to your *SAS-config-dir*`/Web/Staging/sas.wip.apps9.2.ear` staging directory. However, do not overwrite the original EAR file unless you already made a backup copy.

Include SAS JAR Files in the System Classpath

The SAS JAR files that you copied in the previous step provide the classes that are needed for the SAS trusted login module. This step instructs you to copy them to a directory within the domain directory that hosts the SAS Web applications. Also, the classes in the JAR files must be made available to the system classpath. This step provides two strategies for adding them to the classpath. Follow these steps:

1. Make a `webauth` directory and then copy the SAS JAR files to the directory:

   ```
   mkdir SAS-config-dir/Lev1/Web/SASDomain/webauth
   ```

   The `sas.svc.sec.login.weblogic.mbean.jar` file contains an MBean implementation that lets you use the WebLogic Administration Console to add and configure the SAS trusted login module as a WebLogic authentication provider. The other JAR files contain supporting classes for the SAS trusted authentication provider.

2. Add the JAR files to the system classpath. Perform one of the following three options:

   - If your software was installed in 09w21 or later, the JAR files already exist in the file but are commented out. Remove the comment characters at the beginning of each line.
   - Modify the common environment script in the WebLogic installation. This makes the classes available to all the tools and servers that are configured to use the installation of WebLogic on the current machine. Modify the classpath in the following script:
     - `WL_HOME/common/bin/commEnv.sh`
     - For Windows operating environments, the script is named commEnv.cmd.

   Modify the wrapper script in `SASDomain/bin`. This option is effective if you started from an automated configuration performed by the SAS Deployment Wizard or used the sample domain that is provided with a manual configuration as a guide. If you choose this option, then you must use the tools and commands such as wlst.cmd, startNodeManager.cmd, and installNodeMgrSvc.cmd from the `SASDomain/bin` directory. Modify the classpath in the following wrapper script:
     - `SAS-config-dir/Lev1/Web/SASDomain/bin/commEnvSAS.sh`

     For Windows operating environments, the script is named commEnvSAS.cmd.

UNIX

For UNIX operating environments, add the lines in bold to include the JAR files in the classpath. Each of the environment variables must be added all on one line. Change the value of WEBAUTH to match the location of the JAR files in step 1. Make these additions immediately before the statement `export WEBLOGIC_CLASSPATH`.

```
. "${WL_HOME}/common/bin/commEnv.sh"

WEBAUTH="/opt/SAS/Config/Lev1/Web/SASDomain/webauth"

WEBAUTH_CLASSPATH="${WEBAUTH}/sas.core.jar:${WEBAUTH}/sas.core.nls.jar:
${WEBAUTH}/sas.oma.omi.jar:${WEBAUTH}/sas.security.sspi.jar:${WEBAUTH}/
sas.svc.connection.jar:${WEBAUTH}/sas.svc.connection.nls.jar:${WEBAUTH}
/sas.svc.sec.login.jar:${WEBAUTH}/sas.svc.sec.login.nls.jar:${WEBAUTH}/
sas.svc.sec.login.weblogic.jar:${WEBAUTH}/sas.svc.sec.login.weblogic.mb
ean.jar:${WEBAUTH}/sas.svc.sec.login.weblogic.mbean.nls.jar:${WEBAUTH}/
sas.svc.sec.login.weblogic.nls.jar:${WEBAUTH}/sas.svcs.security.weblogi
c.jar:${WEBAUTH}/sas.svcs.security.weblogic.nls.jar"

WEBLOGIC_CLASSPATH="${WEBAUTH_CLASSPATH}:${WEBLOGIC_CLASSPATH}"
export WEBLOGIC_CLASSPATH
```

Windows

For Windows operating environments, add the following lines to include the JAR files in the classpath. Each of the set commands must be entered all on one line. Change the value of WEBAUTH to match the location of the JAR files in step 1. Make these additions immediately before the statement `set WEBLOGIC_CLASSPATH`.

```
 call "%WL_HOME%\common\bin\commEnv.cmd"

set WEBAUTH=C:\SAS\Config\Lev1\Web\SASDomain\webauth

set WEBAUTH_CLASSPATH=%WEBAUTH%\sas.core.jar;%WEBAUTH%\
sas.core.nls.jar;%WEBAUTH%\sas.oma.omi.jar;%WEBAUTH%\sas.security.sspi.
jar;%WEBAUTH%\sas.svc.connection.jar;%WEBAUTH%\sas.svc.connection.nls.j
ar;%WEBAUTH%\sas.svc.sec.login.jar;%WEBAUTH%\sas.svc.sec.login.nls.jar;
%WEBAUTH%\sas.svc.sec.login.weblogic.jar;%WEBAUTH%\sas.svc.sec.login.we
blogic.mbean.jar;%WEBAUTH%\sas.svc.sec.login.weblogic.mbean.nls.jar;%WE
BAUTH%\sas.svc.sec.login.weblogic.nls.jar;%WEBAUTH%\sas.svcs.security.w
eblogic.jar;%WEBAUTH%\sas.svcs.security.weblogic.nls.jar

set WEBLOGIC_CLASSPATH=%WEBAUTH_CLASSPATH%;%WEBLOGIC_CLASSPATH%
```

3. Stop the WebLogic Administration Server.
4. If the node manager is installed as a Windows service, then uninstall it, reinstall it, and restart it. If you chose to modify the commEnvSAS.sh wrapper script, then you must use the node manager commands that are located in the `SASDomain\bin` directory.

Set the CLASSPATH for the Remote Services JVM and Restart

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

- *`WL_HOME`*`/server/lib/wls-api.jar`
- *`WEBAUTH`*`/sas.svc.sec.login.weblogic.jar`

Caution: For WebLogic 10.3.2 and later, a different .jar file is also required. Instead of ***`WL_HOME`*****`/server/lib/wls-api.jar`**, use ***`WL_INSTALL_ROOT`*****`/server/lib/wlthint3client.jar`**. ***`WL_INSTALL_ROOT`*** is a level above

WL_HOME. For example, if ***WL_HOME*** is **`C:\bea\wlserver_10.3`**, then ***WL_INSTALL_ROOT*** is **`C:\bea`**.

UNIX

1. Edit the file *SAS-config dir*`/Lev1/Web/Applications/RemoteServices/RemoteServices.sh`. Add the fully qualified path to the JAR files in the previous list to the `-Dsas.app.class.path` property:

   ```
   start2)
   "$JAVA_JRE_COMMAND"
   -Dsas.ext.config="/opt/SAS_92/SASFoundationServices/9.2/sas.java.
   ext.config" \
   -classpath "$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_92wls/
   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:/WL_HOME/server/
   lib/wls-api.jar:/opt/SAS/Config/Lev1/Web/SASDomain/webauth/sas.svc.
   sec.login.weblogic.jar” \
   ```

2. Restart Remote Services.

Windows

1. If Remote Services is configured to run as a service, then edit the *SAS-config-dir*`\Lev1\Web\Applications\RemoteServices\wrapper.conf` configuration file. The resulting property should resemble this example:

   ```
   wrapper.java.additional.3=-Dsas.app.class.path="C:\SAS\Config\Lev1\
   Web\Applications\RemoteServices; wl_home\server\lib\
   wls-api.jar;C:\SAS\Config\Lev1\Web\SASDomain\webauth\
   sas.svc.sec.login.weblogic.jar"
   ```

   If Remote Services is started from a script, then edit RemoteServices.bat to resemble this example:

   ```
   -Dsas.app.class.path="%REMOTESERVICESDIR%;wl_home\server\
   lib\wls-api.jar;C:\SAS\Config\Lev1\Web\SASDomain\webauth\
   sas.svc.sec.login.weblogic.jar"
   ```

2. Restart Remote Services.

Configure the WebLogic Authentication Provider

SAS provides a custom WebLogic authentication provider. The class files for the provider were copied to the WebLogic installation directory in an earlier step. To configure the SAS trusted login module as a WebLogic authentication provider, follow these steps:

1. Start the WebLogic Administration Server.
2. Access the WebLogic Administration Console. In a default configuration that SAS Deployment Wizard performs, the URL is http://hostname:7501/console.
3. Under Domain Structure, click **Security Realms**
4. Click **myrealm**—or the name of your realm, if you created your own—and click the **Providers** tab.
5. Click **Lock & Edit** to enable editing in the console, and click **New** in the Authentication Providers table.
6. In the **Name** field, enter `SASTrustedAuthenticator`. From the **Type** list, select `WLTrustAuthenticator`. Click **OK**.
7. The SASTrustedAuthenticator should appear last in the authenticator order below the default authenticator and identity asserter. Click the hyperlinked name for SASTrustedAuthenticator.
8. Set the value for **Control Flag** to `OPTIONAL`. Click the **Provider Specific** link. This is where you set the configuration values for the trusted login module. Set these values:

   Host: *"fully-qualified-hostname-for-metadata-server"*

   Domain: `web`

   Port: `8561`

   Encrypt: `false`

   Trusted User: sastrust@saspw

   Trusted Password: *encoded-password*

   Debug: `false`

   If the trusted user identity is a host identity, preface it with the domain and a single slash, such as *`domain\sastrust`*. This format should match the identity that is stored in the SAS metadata repository.
9. Click **Save** and then **Activate Changes**.
10. Restart the WebLogic Administration Server.

Do not set the control flag for other authentication providers to SUFFICENT. For more information, see section "Using Other Authentication Providers" on the next page.

Authentication with the WebLogic Internal LDAP

Add SAS users to the WebLogic embedded LDAP server.

1. Select **Security Realms** > **myrealm**. Click the **Users and Groups** tab.
2. Click **Groups**.
3. Click **New**, provide a name of `SASUsers` and click **OK**.
4. Click **Users**.

5. Click **New**, provide a user name (`sasdemo`) and password, and click **OK**.
6. Click the link for the new user account, and click **Groups**.
7. Select **SASUsers** from the list of available groups, use the right-arrow icon to assign the user to the group, and click **Save**.

Using Other Authentication Providers

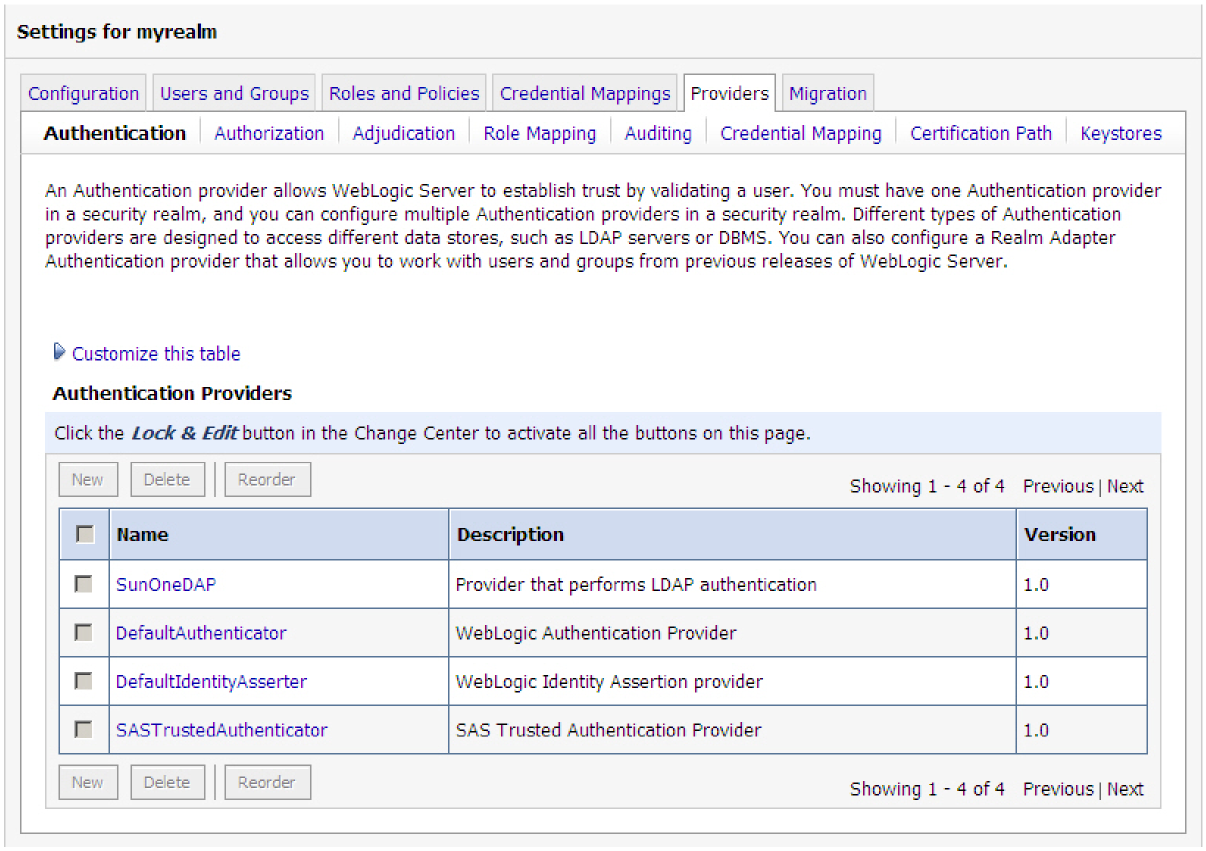
If the example of using the internal LDAP services does not apply to your situation, see your WebLogic documentation for information about the authentication providers that you should use. Some situations might require using more than one provider. See your WebLogic documentation also for information about changing the order of the authentication providers and control flags. In most cases, the SAS Web authentication provider should appear last in the list of providers and also have a control flag of `OPTIONAL`.

In order for the SAS Web authentication provider to execute, the other providers before it must have a control flag of OPTIONAL or REQUIRED. If you set the control flag for an authentication provider to SUFFICIENT, and it is executed earlier than the SAS Web authentication provider, then this setting prevents the SAS Web authentication provider from executing. Access through the SAS Logon Manager will fail. At least one authentication provider must have a control flag of REQUIRED (such as an LDAP or Active Directory provider) to prevent access by unauthenticated users. If an authentication provider is marked as REQUIRED, the WebLogic administrator credentials must be available in the user registry for that provider.

There are many ways to configure the authentication order. In general, if you are trying to authenticate to a third party such as LDAP, that provider should be first. Below is a sample for the authentication order and control flag values when you use a third party LDAP.

ThirdPartyLDAP	Control Flag=Required
DefaultAuthenticator	Control Flag=Optional
DefaultIndentityAsserter	No Control Flag
SASTrustedAuthenticator	Control Flag=Optional

To verify the control flag values, you need to click on each authentication provider.



Settings for myrealm

An Authentication provider allows WebLogic Server to establish trust by validating a user. You must have one Authentication provider in a security realm, and you can configure multiple Authentication providers in a security realm. Different types of Authentication providers are designed to access different data stores, such as LDAP servers or DBMS. You can also configure a Realm Adapter Authentication provider that allows you to work with users and groups from previous releases of WebLogic Server.

Authentication Providers

Click the ***Lock & Edit*** button in the Change Center to activate all the buttons on this page.

	Name	Description	Version
☐	SunOneDAP	Provider that performs LDAP authentication	1.0
☐	DefaultAuthenticator	WebLogic Authentication Provider	1.0
☐	DefaultIdentityAsserter	WebLogic Identity Assertion provider	1.0
☐	SASTrustedAuthenticator	SAS Trusted Authentication Provider	1.0

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.

Start the WebLogic Server

To start the WebLogic Server and install the rebuilt SAS Web Infrastructure Platform applications EAR file, follow these steps.

1. Select **Environment** > **Servers** > **SASServer1** > **Control** and click **Start**. Wait for the application server to start completely.

2. Select **Lock & Edit** and then **Deployments**.
3. Install the rebuilt `sas.wip.apps9.2.ear` file and deploy it to SASServer1.
4. Start the `sas.wip.apps9.2.ear` application.

Log On to Verify the Web Authentication Configuration

If your site was migrated from a previous SAS release and therefore already has user IDs and authentication domains already registered in metadata, try to log 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, such as the embedded LDAP server. 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

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