

Configuration Guide

Configuring Apache HTTP Server as a Reverse Proxy Server for SAS 9.2 Web Applications Deployed on BEA WebLogic Server 9.2

This document describes how to configure Apache HTTP Server as a reverse proxy server to a WebLogic Server that is hosting the SAS Web applications. This document supplements the WebLogic Server documentation that describes configuring Apache HTTP Server with a WebLogic Server plug-in. The WebLogic Server documentation is provided in [Using Web Server Plug-Ins with WebLogic Server](#).

Configure the WebLogic Server Plug-In to Apache HTTP Server

To configure Apache HTTP Server with the WebLogic Server plug-in, follow these steps:

1. Locate the correct plug-in in `WL_HOME/weblogic92/server/plugin/` based on your operating system and architecture:

| Operating System | Location |
|------------------|--|
| Solaris | <code>WL_HOME/weblogic92/server/plugin/solaris/sparc</code> <code>WL_HOME/weblogic92/server/plugin/solaris/sparc/largefile</code> <code>WL_HOME/weblogic92/server/plugin/solaris/x86</code> <code>WL_HOME/weblogic92/server/plugin/solaris/x86/largefile</code> |
| Windows 32-bit | <code>WL_HOME/weblogic92/server/plugin/win/32</code> |
| HP-UX 11i | <code>WL_HOME/weblogic92/server/plugin/hpux11/IPF64</code> |

After locating the correct directory, choose the correct file:

| Apache HTTP Server Version | Regular Encryption | 128-bit Encryption |
|-------------------------------|---------------------------|------------------------------|
| Standard Apache Version 2.0.x | <code>mod_wl_20.so</code> | <code>mod_wl128_20.so</code> |
| Standard Apache Version 2.2.x | <code>mod_wl_22.so</code> | <code>mod_wl128_22.so</code> |

2. Copy the plug-in to the `APACHE_HOME/modules` directory.

3. Edit the `APACHE_HOME/conf/httpd.conf` file so that Apache HTTP Server:

- loads the WebLogic plug-in
- configures the plug-in with information about the WebLogic Server
- uses the plug-in for the SAS Web applications

Add the `LoadModule`, `IfModule`, and `Location` directives to the `httpd.conf` file. See the following example:

```
LoadModule weblogic_module      modules/mod_wl_22.so

<IfModule mod_weblogic.c>
    WebLogicHost hostname1.example.com
    WebLogicPort 7001
    #
    # Debug ALL
    # DebugConfigInfo ON
</IfModule>

# uncomment these entries to provide access to the
# SAS Web Administration Console from Apache HTTP Server
#<Location /SASAdmin>
#   WLCookieName SASAdmin.sessionID
#   SetHandler weblogic-handler
#</Location>
#
#<Location /SASContentServer>
#   SetHandler weblogic-handler
#</Location>

<Location /SASBIDashboard>
    WLCookieName SASBIDashboard.sessionID
    SetHandler weblogic-handler
</Location>

<Location /SASBIWS>
    WLCookieName SASBIWS.sessionID
    SetHandler weblogic-handler
</Location>

<Location /SASLogon>
    WLCookieName SASLogon.sessionID
    SetHandler weblogic-handler
</Location>

<Location /SASPackageViewer>
    WLCookieName SASPackageViewer.sessionID
    SetHandler weblogic-handler
</Location>

<Location /SASPortal>
    WLCookieName SASPortal.sessionID
    SetHandler weblogic-handler
</Location>
```

```

<Location /SASPreferences>
    WLCookieName SASPreferences.sessionID
    SetHandler weblogic-handler
</Location>

<Location /SASSharedApps>
    WLCookieName SASSharedApps.sessionID
    SetHandler weblogic-handler
</Location>

<Location /SASSharedPortlets>
    WLCookieName SASSharedPortlets.sessionID
    SetHandler weblogic-handler
</Location>

<Location /SASStoredProcess>
    WLCookieName SASStoredProcess.sessionID
    SetHandler weblogic-handler
</Location>

<Location /SASTheme_default>
    # Serving static HTML is configured later. Set this
    # to default-handler so that Apache HTTP Server serves it.
    SetHandler default-handler
</Location>

<Location /SASWebDoc>
    WLCookieName SASWebDoc.sessionID
    SetHandler weblogic-handler
</Location>

<Location /SASWebReportStudio>
    WLCookieName SASWebReportStudio.sessionID
    SetHandler weblogic-handler
</Location>

<Location /SASWebOLAPViewer>
    # Notice the lower case session ID for this application.
    WLCookieName SASWebOLAPViewer.sessionid
    SetHandler weblogic-handler
</Location>

# uncomment this if the site hosts SAS solution software that uses
# desktop applications to access SAS Web applications
#<Location /SASWIPClientAccess>
#   SetHandler weblogic-handler
#</Location>

# uncomment this entry to provide access to SAS Web services
#<Location /SASWIPSoapServices>
#   SetHandler weblogic-handler
#</Location>

```

Configure Apache HTTP Server to Serve SAS Themes Static Content

This step is not required, but does take advantage of using Apache HTTP Server to serve the static HTML and images that are part of the SAS Themes application. Using Apache HTTP Server for this purpose shifts the processing load of serving static HTML and images from the Web application server to the HTTP server. Follow these steps:

1. In a temporary directory, extract the contents of *SAS-config-dir/Levn/Web/Staging/sas.themes.ear*:

```
jar xf /opt/SAS/Config/Lev1/Web/Staging/sas.themes.ear
```

The WAR file *sas.theme.default.war* is extracted.
2. Change directory to *APACHE_HOME/htdocs* and make a new directory named *SASTheme_default*.
3. Extract the *sas.theme.default.war* file into *APACHE_HOME/htdocs/SASTheme_default*.

The directory name, *SASTheme_default*, is used because it is the value for the context root that is defined in the *META-INF/application.xml* file for the *sas.themes.ear* application.

Reconfigure the Connections for the SAS Web Applications

Although configuring Apache HTTP Server as a proxy server means that the SAS Web applications are accessed through the HTTP server and port, the SAS Web applications maintain the connection information in the SAS Metadata Server for resource access and inter-application communication. In the initial configuration performed by the SAS Deployment Wizard, the connections are based on the Web application server host and port number. Because the HTTP server is configured to proxy the SAS Web applications, you must change the connection information for the external (customer-facing) SAS Web applications. Change the connection information to use the HTTP server host name and port number. The following list identifies the SAS Web applications that must be reconfigured:

- BI Dashboard 4.2
- BI Web Services for Java 9.2
- Help Viewer Meta Config 9.2
- Information Delivery Portal 4.2
- Logon Manager 9.2
- Package Viewer 4.2
- Preferences Manager 9.2
- SASTheme_default
- Shared Applications 9.2
- Shared Portlets 9.2
- Stored Process Web App 9.2
- Web OLAP Viewer 4.2
- Web Report Studio 4.2

- Web Administration Console 9.2 (optional, if you want SAS Web application administration to be available from the Apache HTTP Server)
- Web Infra Platform Soap Svcs 9.2 (optional, if you want SAS Web services available from the Apache HTTP Server)
- Web Infra Platform Client Access 9.2 (optional, if the site has a SAS solution with desktop applications that access SAS Web applications)

To change the connection access information, follow these steps in SAS Management Console:

1. Select **Application Management > Configuration Manager**.
2. Right-click the SAS Web application that you want to reconfigure, and select **Properties**.
3. Click the **Connection** tab, set **Host Name** and **Port Number** to the host name and port number of the HTTP server, and then click **OK**.

Change the Connection for SAS Content Server

If the SAS Web Administration Console application is proxied, then perform these steps. Similar to the change required for each of the SAS Web applications, SAS metadata must be changed to identify the host name and port of the HTTP server. To reconfigure the host name and port of the SAS Content Server in SAS metadata, follow these steps in SAS Management Console:

1. Select **Server Manager > SAS Content Server**.
2. Right-click the **Connection: SAS Content Server** icon in the right panel and select **Properties**.
3. Click the **Options** tab and set the **Host name** and **Port number** values to the host name and port number of the HTTP server.
4. Click **OK**.
5. In SAS Management Console, select the **Folders** tab.
6. Right-click the **SAS Folders** icon at the root of the folder tree in the left pane and select **Properties**.
7. Select the **Content Mapping** tab and use the **Server** menu to select **SAS Content Server**. The **URL** field then shows the HTTP Server host name and port. Click **OK**.

Change the WebDAV Repository URL

If the SAS Web Administration Console application is proxied, then perform these steps. There are five applications that use SAS metadata to identify the connection information for the SAS Content Server. These applications are identified in the following list:

- Remote Services
- SASPackageViewer4.2 Local Services
- SASPortal4.2 Local Services
- SASStoredProcess9.2 Local Services
- SASWebReportStudio4.2 Local Services

To reconfigure the WebDAV URL for the applications, follow these steps in SAS Management Console:

1. Select **Environment Management > Foundation Services Manager**.
2. Select the application and then select **Core > Information Service**.
3. Right-click **Information Service** and select **Properties**.
4. On the **Information Service Properties** dialog box, click the **Service Configuration** tab and then click **Configuration**.
5. On the **Information Service Configuration** dialog box, click the **Repositories** tab.
6. Select **WebDAV** and then click **Edit**.
7. Change the **Host** and **Port** values to the host name and port of the HTTP server.
8. Click **OK** to close the **Information Service Configuration** dialog box.
9. Click **OK** to close the **Information Service Properties** dialog box.

SAS Web Report Studio 4.2 Specific Update

By default, SAS Web Report Studio 4.2 uses a special redirection filter. When used with a proxy server, this filter must be disabled. To disable the filter, perform the following steps with SAS Management Console:

1. Select **Application Management > Configuration Manager**.
2. Right-click **Web Report Studio 4.2**, and select **Properties**.
3. Click **Advanced**, and then click **Add**.
4. Enter a Property Name of `App.RedirectionFilterDisabled` and a Value of `true`.
5. Restart SAS Web Report Studio 4.2 from the Administration Console.

Start the Software Applications and Verify the Configuration

Start the software applications in the following order:

1. SAS Remote Services
2. Apache HTTP Server
3. the Web application server

After the Web application server is available and the SAS Web applications are active, verify that the configuration is valid by opening a Web browser to an application such as SAS Information Delivery Portal: <http://httpserver.example.com/SASPortal>. If the configuration is valid, then you are challenged for log on credentials and then the SAS Information Delivery Portal application is available.

Troubleshooting

The following list identifies the high-level steps used to troubleshoot this configuration:

1. Confirm that Apache HTTP Server is running and that your Web browser is connecting to Apache HTTP Server by viewing the `APACHE_HOME/logs/access.log` file as you make a request.
2. Uncomment the `Debug` and `DebugConfigInfo` options in the `httpd.conf` file and restart Apache HTTP Server.

The `Debug` option enables logging from the WebLogic Server plug-in to record debugging information to `/tmp/wlproxy.log` on UNIX systems and `c:\TEMP\wlproxy.log` on Windows systems.

The `DebugConfigInfo` option enables viewing the configuration parameters for the plug-in. When this option is ON, it enables a special query parameter, `__WebLogicBridgeConfig`.

Open a Web browser to http://httpserver.example.com/SASPortal?__WebLogicBridgeConfig.

If the Web page does not display at all, then either Apache HTTP Server or the WebLogic Server plug-in is not installed correctly. If the Web page does display, then confirm that host, port, and status is OK for the servers in the General Server List.

3. View the WebLogic Server log. If there is no activity in the log when a URL for a SAS Web application is accessed, then check the following items:
 - Use SAS Management Console to review the properties for the SAS Web application and confirm that the host and port values are set to the Apache HTTP Server. Use the information in section “Reconfigure the Connections for the SAS Web Applications” for instructions on how to access the metadata.
 - Use the WebLogic Server Administration Console to check the server start up parameters. Confirm that the `-Dsas.auto.publish.port=` parameter identifies the port that the WebLogic Server is listening on. For a single WebLogic Server topology, the SAS Deployment Wizard sets the port to 7001.

Recommended Reading

The following URLs are current as of October 2009.

Oracle Corporation. 2007. *Using Web Server Plug-Ins with WebLogic Server*. Available at http://download.oracle.com/docs/cd/E13222_01/wls/docs92/pdf/plugins.pdf.

Oracle Corporation. 2007. *Using WebLogic Server Clusters*. Available at http://download.oracle.com/docs/cd/E13222_01/wls/docs92/pdf/cluster.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 © 2009 SAS Institute Inc., Cary, NC, USA. All rights reserved.