

This is a simple guide to install Apache with SM

It assumes Tomcat is working correctly in <http://server:8280/sm>). After finishing the guide, it will work also going to <http://server/sm>

All the guides that describe this process have the goal to configure ssl+sso/lwssso. It's better practice to make sure Apache works before starting with the certificates and other complex configuration.

Install Apache 2.2

Download the binaries for httpd 2.2 (note: the official web site www.apache.org only provides source code that needs compilation. Use for example <https://www.apachehaus.com/cgi-bin/download.plx> for binaries). There are new versions (eg Apache 2.4), but the latest supported is 2.2

Follow the instructions to install it. Set the network domain and server name according to your system

Download the mod_jk connector

Download the Tomcat - Apache httpd connector module (mod_jk.so) from the Apache.org website (<http://archive.apache.org/dist/tomcat/tomcat-connectors/jk/binaries/windows/>). Look for a filename like "tomcat-connectors-1.2.40-windows-i386-httpd-2.2.x.zip"

Extract from the .zip file 'mod_jk.so' and save it in <Apache install folder>\Apache2.2\modules

Test Apache is correctly installed and running

Start service for Apache (normally it's called Apache2.2)

Open a web browser and go to <http://localhost> or http://<server_name> or http://<server_IP> . All of those should open a page that says "It works!"

If you have problems starting Apache because the port :80 is already in use, make sure IIS is not running.

Sometimes the services IIS starts automatically with Windows. If that's the case, change the startup type to "Manual" and restart Windows to make sure port :80 is free when Apache wants to use it.

Apache configuration files

mod_jk.conf

Go to folder <Apache Install folder>\Apache2.2\conf and create a file called "**mod_jk.conf**" with this content (make sure you enter the correct <Tomcat root install dir>):

```
#
# The following line instructs Apache to load the jk module
# Use the mod_jk.so file, for Tomcat 5.x and greater.
# The mod_jk.dll is for Tomcat 4.x and lower.
# Using the wrong version will causes load errors...!!
#

LoadModule jk_module modules/mod_jk.so

#ajp13 workersfile

JkWorkersFile conf/workers.properties

#mod_jk log file

JkLogFile logs/mod_jk.log

#
# Log level to be used by mod_jk
#
JkLogLevel error

#####
#                               SSL configuration                               #
#
# By default mod_jk is configured to collect SSL information from
# the apache environment and send it to the Tomcat workers. The
# problem is that there are many SSL solutions for Apache and as
# a result the environment variable names may change.
#
# The following (commented out) JK related SSL configuration
# can be used to customize mod_jk's SSL behaviour.
#
# Should mod_jk send SSL information to Tomcat (default is On)
# JkExtractSSL Off
#
# What is the indicator for SSL (default is HTTPS)
# JkHTTPSIndicator HTTPS
#
# What is the indicator for SSL session (default is SSL_SESSION_ID)
# JkSESSIONIndicator SSL_SESSION_ID
#
# What is the indicator for client SSL cipher suit (default is SSL_CIPHER)
# JkCIPHERIndicator SSL_CIPHER
#
# What is the indicator for the client SSL certificated (default is SSL_CLIENT_CERT)
# JkCERTSIndicator SSL_CLIENT_CERT
#
#
#####

#
# Root context mounts for Tomcat
#
#JkMount /example/*.jsp ajp13
#JkMount /example/servlet/* ajp13
```

```

JKMount /sm/*.jsp ajp13
JKMount /sm/*do ajp13
JkMount /sm/secure/* ajp13
JkMount /sm/messagebroker/* ajp13
JkMount /sm/configuration/* ajp13
JkMount /sm/* ajp13

#####
# Auto configuration for the /sm webapps context starts.
#####

#
# The following line makes apache aware of the location of the /sm webapps context
#
Alias /sm "<Tomcat root install dir>/webapps/sm"
<Directory "<Tomcat root install dir>/webapps/sm">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>

#
# The following line prohibits users from directly accessing WEB-INF
#
<Location "/sm/WEB-INF/">
    AllowOverride None
    deny from all
</Location>

#
# Use Directory too. On Windows, Location doesn't work unless case matches
#
<Directory "<Tomcat root install dir>/webapps/sm/WEB-INF/">
    AllowOverride None
    deny from all
</Directory>

#
# The following line prohibits users from directly accessing META-INF
#
<Location "/sm/META-INF/">
    AllowOverride None
    deny from all
</Location>

#
# Use Directory too. On Windows, Location doesn't work unless case matches
#
<Directory "<Tomcat root install dir>/webapps/sm/META-INF/">
    AllowOverride None
    deny from all
</Directory>

#####
# Auto configuration for the /sm webapps context ends.
#####

```

workers.properties

Go to folder <Apache Install folder>\Apache2.2\conf and create a file called **"workers.properties"** with this content (make sure you enter the correct *port=8009*):

```

#
#----- DEFAULT WORKER LIST -----
#-----
#
worker.list=ajp13

#
#----- DEFAULT ajp13 WORKER DEFINITION -----
#-----
#
worker.ajp13.port=8009
worker.ajp13.host=localhost
worker.ajp13.type=ajp13
#
worker.ajp13.lbfactor=1

#
#----- URIWORKERMAP DEFINITION -----
#-----
#
[uri:/sm/servlet/*]
info=Prefix mapping
[uri:/sm/*.jsp]
info=Extension mapping
[uri:/sm/*.do]
info=Extension mapping
[uri:/sm/attachments/*]
info=Extension mapping
[uri:/sm/cwc/nav.menu]
info=Extension mapping

```

httpd.conf

Go to folder <Apache Install folder>\Apache2.2\conf and edit "**httpd.conf**" and add this line at the end:

```
include conf/mod_jk.conf
```

Configure Tomcat file server.xml

We configured worker ajp13. But we have to make sure it must be in the Tomcat config file server.xml.

Go to <Tomcat install folder>\conf and edit file **server.xml**. Make sure the worker ajp13 is defined and the port is the one we configured in workers.properties (in the above example *worker.ajp13.port=8009*).

Note it's important to have the correct port, protocol and jvmRoute. They're normally in 2 different lines inside server.xml

```

<Connector port="8009" protocol="AJP/1.3" enableLookups="false" tomcatAuthentication="false"
redirectPort="8043" />

[..]

<Engine name="Catalina" defaultHost="localhost" jvmRoute="ajp13">

```

Verify everything is working as expected

Check web tier is accessible via Tomcat

Restart Tomcat, open a web browser and go to

http://<server_ip>:<Tomcat_port>/sm (eg <http://11.43.64.116:8080/sm>)

The SM login page should appear

Check web tier is accessible via Apache

Restart Apache, open a web browser and go to http://<server_ip>/sm (eg

<http://11.43.64.116/sm>)

The SM login page should appear