

Building, Deploying and Configuring Web ARchive, OpenSSO

Anant Kadam Aravindan Ranganathan
anant.kadam@sun.com aravind@sun.com
Member of Technical Staff Senior Staff Engineer
Sun Microsystems Inc.

Dennis Seah
dennis.seah@sun.com
Staff Engineer
Sun Microsystems Inc.

July 2006

Abstract

This paper describes the building of Web ARchive WAR file for Open SSO web applications; ways to deploy it; and how does the configuration *JSP™* setup the web application.

1 Introduction

OpenSSO system provides Single Sign On and Access Control solutions. It comprises of server and agents where the former is responsible for authentication, session control, access control policies evaluation; and audit logging; and the latter is the policy enforcement point.

In this paper, we describe an easy and convenient way of deploying and configuring the server on a *J2EE* compliant web container.

2 Building of OpenSSO server WAR file

The WAR file is built with *Ant*¹. The target is *war* and it does the followings

1. copy all the *jar* files (except *j2ee.jar*) under *oposso/products/amserver/extlib* directory to *WEB-INF/lib* directory;
2. copy java classes under *oposso/products/amserver/built/classes* to *WEB-INF/classes* directory;

¹Apache Ant is a Java-based build tool. <http://ant.apache.org/>

3. copy lockhart ² libraries *jars* under `opensso/resources/lockhart/lib` directory to `WEB-INF/lib` directory;
4. copy `browserVersion.js` under `opensso/resources/lockhart/js` to `js` directory;
5. copy `css_generic.css` and `styles` under `opensso/resources/lockhart/css` to `css` directory;
6. copy all DTD files under `opensso/resources/lockhart/dtd` to `com.sun.web.ui/dtd` directory;
7. copy all HTML files under `opensso/resources/lockhart/dtd` to `com.sun.web.ui/html` directory;
8. copy all images files under `opensso/resources/lockhart/images` to `com.sun.web.ui/images` directory;
9. copy all javascript files under `opensso/resources/lockhart/js` to `com.sun.web.ui/js` directory;
10. copy all *JSP* TM under `opensso/resources/lockhart/jsp` to `com.sun.web.ui/jsp` directory;
11. copy `cc.tld` under `opensso/resources/lockhart/xml` to `WEB-INF` directory;
12. copy all XML files under `opensso/resources/lockhart/xml` to `WEB-INF//com.sun.web.ui` directory;
13. copy `jato.jar` under `opensso/resources/jato` to `WEB-INF/lib` directory;
14. copy `jato.tld` under `opensso/resources/jato` to `WEB-INF` directory;
15. copy HTML files (doc) under `docs/console/javahelp/en` to `html/en/help` directory;
16. copy javahelp ³ libraries *jars* under `opensso/resources/javahelp/lib` directory to `WEB-INF/lib` directory;
17. copy `css` files under `opensso/resources/javahelp/css` to `html/en/help/css` directory;
18. copy `ents` files under `opensso/resources/javahelp/css` to `html/en/help/ents` directory;
19. copy `graphics` files under `opensso/resources/javahelp/css` to `html/en/help/graphics` directory;
20. copy `index.html` under `opensso/products/amserver/www/authentication/UI` directory to `base` directory;

²Sun common web components

³Java online help system

21. copy all images under `opensso/products/amserver/www/authentication/UI/login_images` to `login_images` directory;
22. copy all javascript files under `opensso/products/amserver/www/authentication/UI/js` to `js` directory;
23. copy all *JSP*TM and HTML files under `opensso/products/amserver/www/authentication/UI` to `config/auth/default` directory;
24. copy the following XML files from `opensso/products/amserver/xml` directory to `config/auth/default` and `config/auth/default.en` directories;
 - (a) `AD.xml`
 - (b) `Anonymous.xml`
 - (c) `Application.xml`
 - (d) `Cert.xml`
 - (e) `Datastore.xml`
 - (f) `HTTPBasic.xml`
 - (g) `JDBC.xml`
 - (h) `LDAP.xml`
 - (i) `Membership.xml`
 - (j) `MSISDN.xml`
 - (k) `NT.xml`
 - (l) `WindowsDesktopSSO.xml`
25. copy `opensso/products/amserver/war/xml/serverconfig.xml` to `WEB-INF/template/sms` directory;
26. copy all the properties files (except `AMConfig.properties`) under `opensso/products/amserver/resources` directory to `WEB-INF/classes` directory;
27. copy `isAlive.jsp`, `Debug.jsp` and `amadm.jsp` from `opensso/products/amserver/www/jsp` directory to base deployment directory;
28. copy all DTD files (including `auth.tld`) from `opensso/products/amserver/xml` directory to `WEB-INF` directory;
29. copy `SunAMClientData.xml` and `defaultDelegationPolicies.xml` from `opensso/products/amserver/war/xml/` directory to `WEB-INF/template/sms` directory;
30. copy `AMConfig.properties` from `opensso/products/amserver/war/xml/template/sms` directory to `WEB-INF/template/sms` directory;
31. copy `amAccessControl.xml` and `amConsoleConfig.xml` under `opensso/products/amserver/xml/console` directory to `WEB-INF/classes` directory;
32. copy all resources under `opensso/products/amserver/www/console` to `console` directory;

3 Deploying of OpenSSO server WAR file

The method of deploying WAR file is specified to the Web/Application Server. Please refer to the document of the Web/Application Server for more information.

4 Configuring of OpenSSO server

The OpenSSO server has to be configured after the WAR file is deployed. There is a filter, `amSetupFilter` that intercepts all requests; and redirect them to the `configurator.jsp` if the product is not yet configured.

The product is configured if the following conditions are met.

1. Check for the existence of a file named `AMConfig_{hostname}_{deploymenturi}`, where `hostname` is the host name of the server where the filter is hosted; and `deploymenturi` is the deployment URI of where filter is deployed.
2. This file contains the directory where the access manager configuration data is stored. Check the existence of the directory.

The filter allows all requests to pass through if the product is configured. Otherwise, requests shall be redirected to `configurator.jsp`.

This is what the `configurator.jsp` does

1. Present a web form to capture the following values
 - (a) Host Name
 - (b) Protocol
 - (c) Port Number
 - (d) Deployment URI
 - (e) Super Administrator Password (for OpenSSO server)
 - (f) Directory where OpenSSO server's configuration data shall be stored.
 - (g) Cookie Domain
 - (h) Platform Locale
2. Invoke `AMSetupServlet.processRequest` method to configure the server. This method does the followings
 - (a) Perform validations on the values that are captured by the `configurator.jsp`
 - (b) Tag swaps a set of service schema files;
 - (c) Register these scheme files;
 - (d) Load a set of configuration data; and
 - (e) Re-initialize all static variables.
3. Redirect the request to the login page after the product is successfully configured.