

IBM® Security Access Manager  
for Versions 6.1.1, 7.0, 8.0, 9.0

# **Red Hat JBoss Enterprise Application Platform Integration Guide**



## Note

---

*Before using this information and the product it supports, read the information in [‘Notices’](#) on page 56*

This edition applies to Version 3.1 release i of the IBM Security Access Manager Integration with Red Hat JBoss Enterprise Application Platform and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2012, 2015.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

## Contents

---

|   |           |
|---|-----------|
| <b>PREFACE .....</b>  | <b>6</b>  |
| <b>About this publication .....</b>   | <b>6</b>  |
| <b>Intended audience.....</b>   | <b>6</b>  |
| <b>Access to publications and terminology .....</b>   | <b>6</b>  |
| Publication Library .....   | 6         |
| IBM Terminology website .....   | 7         |
| <b>Accessibility .....</b>  | <b>7</b>  |
| <b>Support information.....</b>   | <b>8</b>  |
| <b>Statement of Good Security Practices.....</b>  | <b>8</b>  |
| <b>INTRODUCING THE INTEGRATION .....</b>  | <b>9</b>  |
| <b>Introduction .....</b>   | <b>9</b>  |
| <b>Integration Product Version Information.....</b>   | <b>10</b> |
| <b>Integration Product Contents .....</b>   | <b>10</b> |
| <b>Paths and Conventions Used in this guide .....</b>   | <b>10</b> |
| <b>PREPARING THE ENVIRONMENT .....</b>  | <b>12</b> |
| <b>IBM Security Access Manager 8 and 9 Runtime for Java prerequisites for Oracle .....</b>                                    | <b>12</b> |
| <b>IBM SECURITY ACCESS MANAGER SINGLE SIGN-ON TO JBOSS<br/>ENTERPRISE APPLICATION PLATFORM .....</b>                          | <b>13</b> |
| <b>Installing and configuring the IBM Security Access Manager Login Module in Full Mode<br/>    Configuration.....</b>        | <b>14</b> |
| Configuring Trust Validation .....  | 16        |
| <b>Installing and configuring the IBM Security Access Manager Login Module in Standalone<br/>    Mode Configuration .....</b> | <b>18</b> |
| Using an IBM Java Runtime Environment.....  | 18        |
| Using an Oracle Java Runtime Environment .....  | 18        |
| <b>IBM SECURITY ACCESS MANAGER CONFIGURATION .....</b>  | <b>19</b> |
| <b>Web Gateway Appliance Configuration .....</b>  | <b>19</b> |
| Junction Management .....   | 19        |
| Session Management.....   | 20        |
| <b>WebSEAL Configuration .....</b>  | <b>21</b> |
| Junction Creation .....   | 21        |
| Session Management.....   | 22        |

---

|   |           |
|---|-----------|
| <b>IBM SECURITY ACCESS MANAGER FOR MOBILE CONFIGURATION (OPTIONAL)</b>                            | <b>24</b> |
| Mobile Authorization Decision Point Configuration   | 24        |
| Configuring Web Reverse Proxy   | 24        |
| Creating the Policy Attributes  | 25        |
| Policy Authoring  | 26        |
| Attaching the Policy  | 26        |
| <b>CONFIGURING JBOSS ENTERPRISE APPLICATION PLATFORM FOR SINGLE SIGN ON</b>                       | <b>27</b> |
| Installing and Configuring the IBM Security Access Manager Login Module                           | 27        |
| Configuring Application Server Wide Single Sign-On  | 28        |
| Configuring Individual Applications for Single Sign-On  | 29        |
| Configuring Java Web Applications for Single Sign On  | 31        |
| Mapping IBM Security Access Manager Groups to Roles in Java Web Application                       | 32        |
| Configuring IBM Security Access Manager for the Sample Application SecTest.ear                    | 32        |
| Mapping IBM Security Access Manager Groups to Roles   | 32        |
| Starting the JBoss Server   | 33        |
| Validating the Integration  | 33        |
| <b>IBM SECURITY ACCESS MANAGER AUTHORIZATION (JACC) FOR JBOSS ENTERPRISE APPLICATION PLATFORM</b> | <b>35</b> |
| Access Manager JACC Policy Provider Installation and Configuration                                | 35        |
| JBoss Enterprise Application Platform Configuration   | 36        |
| Server-wide Authorization Configuration   | 37        |
| Individual Application Authorization Configuration  | 37        |
| IBM Security Access Manager Configuration   | 38        |
| Validating the Integration  | 40        |
| <b>REMOVING THE INTEGRATION</b>   | <b>41</b> |
| Removing Integration for Standalone Mode Configuration  | 41        |
| Removing Configuration and Files  | 41        |
| Removing Integration for Full Mode Configuration  | 41        |
| Removing the IBM Security Access Manager Configuration  | 41        |
| Removing the IBM Security Access Manager Environment  | 41        |
| Removing the IBM Security Access Manager Configuration  | 42        |
| <b>TAM LOGIN MODULE CONFIGURATION OPTIONS</b>   | <b>43</b> |

|   |           |
|---|-----------|
| <b>TROUBLESHOOTING .....</b>  | <b>45</b> |
| <b>Collecting Support Data .....</b>  | <b>54</b> |
| Collecting Data for IBM Security Access Manager for JBoss Enterprise Application Platform ..... | 54        |
| Enable IBM Security Access Manager Login Module Trace .....                                     | 54        |
| JACC Provider Caching .....   | 55        |
| <b>NOTICES .....</b>  | <b>56</b> |
| <b>TRADEMARKS .....</b>   | <b>58</b> |

## Preface

---

### About this publication

This guide explains how to configure and manage your IBM® Security Access Manager installation to work with Red Hat JBoss Enterprise Application Platform.

This document assumes that both IBM Security Access Manager and JBoss Enterprise Application Platform are installed, configured and running on your network. This guide does not provide details on the installation and administration of these products, except where necessary to achieve integration.

### Intended audience

This guide is for users who are responsible for the installation, deployment and administration of IBM Security Access Manager and JBoss Enterprise Application Platform.

Readers must be familiar with the following concepts:

- PC and UNIX operating systems
- Security Management
- Internet protocols, including HTTP, HTTPS, TCP/IP and SSL
- Lightweight Directory Access Protocol (LDAP) and directory services
- A supported user registry
- Authentication and authorization

The reader must also be familiar with the administration of IBM Security Access Manager, Security Access Manager WebSEAL and JBoss Enterprise Application Platform.

### Access to publications and terminology

The following publications complement the information contained in this document:

#### Publication Library

These publications complement the information that is contained in this publication:

##### Base Information

- *IBM® Tivoli® Access Manager Base Installation Guide*

Explains how to install, configure, and upgrade Access Manager software, including the Web portal manager interface.

- *IBM Security Access Manager Base Administrator's Guide*

Describes the concepts and procedures for using Access Manager services. Provides instructions for managing tasks from the Web portal manager interface and by using the pdadmin command.

##### WebSEAL Information

- *IBM Security Access Manager WebSEAL Installation Guide*

Provides installation, configuration, and removal instructions for the WebSEAL server and the WebSEAL application development kit.

## Integration Guide

- *IBM Security Access Manager WebSEAL Administrator's Guide*

Provides background material, administrative procedures, and technical reference information for using WebSEAL to manage the resources of your secure Web domain.

- *IBM Security Access Manager WebSEAL Developer's Reference*

Provides administration and programming information for the Cross-domain Authentication Service (CDAS), the Cross-domain Mapping Framework (CDMF), and the Password Strength Module.

## Web Gateway Appliance Information

- *IBM Security Access Manager Web Gateway Appliance Administration Guide*

Provides information about configuring and maintaining a Security Access Manager environment.

- *IBM Security Web Gateway Appliance Configuration Guide for Web Reverse Proxy*

Provides configuration procedures and technical reference information for the Web Gateway Appliance.

- *IBM Security Web Gateway Appliance Web Reverse Proxy Stanza Reference*

Provides a complete stanza reference for the Web Gateway Appliance Web Reverse Proxy.

## Mobile Information

- *IBM Security Access Manager for Mobile Administration Guide*

Describes how to manage, configure, and deploy an existing IBM Security Access Manager environment.

- *IBM Security Access Manager for Mobile Configuration Guide*

Explains how to complete the initial configuration of IBM Security Access Manager for Mobile.

## IBM Terminology website

The IBM Terminology website consolidates terminology for product libraries in one location. You can access the Terminology website at <http://www.ibm.com/software/globalization/terminology>

## Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

## Support information

IBM Support provides assistance with code-related problems and routine, short duration installation or usage questions. You can directly access the IBM Software Support site at <http://www.ibm.com/software/support/probsub.html>

## Statement of Good Security Practices

IT system security involves protecting systems and information through prevention, detection and response to improper access from within and outside your enterprise. Improper access can result in information being altered, destroyed, misappropriated or misused or can result in damage to or misuse of your systems, including for use in attacks on others. No IT system or product should be considered completely secure and no single product, service or security measure can be completely effective in preventing improper use or access. IBM systems, products and services are designed to be part of a comprehensive security approach, which will necessarily involve additional operational procedures, and may require other systems, products or services to be most effective. IBM DOES NOT WARRANT THAT ANY SYSTEMS, PRODUCTS OR SERVICES ARE IMMUNE FROM, OR WILL MAKE YOUR ENTERPRISE IMMUNE FROM, THE MALICIOUS OR ILLEGAL CONDUCT OF ANY PARTY.



## Introducing the Integration

---

### Introduction

This integration guide describes the process for enabling IBM Security Access Manager for Web Single Sign-on (SSO) to Red Hat JBoss Enterprise Application Platform server.

The following scenarios are supported:

#### Access Manager Single Sign-On to JBoss Enterprise Application Platform Server

In this scenario, a user first authenticated to IBM Security Access Manager. The user identity is passed to the JBoss Enterprise Application Platform server and a Subject for the user is established inside the Java™ 2 Enterprise Edition container. Role requirements for access to the container are established through IBM Security Access Manager group memberships to Java 2 EE role mapping. IBM Security Access Manager for JBoss Enterprise Application Platform can be configured for either

- Full Mode (Recommended) configuration as described in *Installing and configuring the IBM Security Access Manager Login Module in Full Mode Configuration* on page 14; or
- Standalone mode configuration as described in *Installing and configuring the IBM Security Access Manager Login Module in Standalone Mode Configuration* on page 18.

#### JACC Authorization for JBoss Enterprise Application Platform Server

This scenario provides fine grained authorization of J2EE objects, such as EJBs, evaluated by IBM Security Access Manager. The Java Authorization Contract for Containers (JACC) standard is the mechanism through which authorization decisions are passed from JBoss Enterprise Application Platform server to IBM Security Access Manager JACC policy provider. Role requirements for access to the container are established and controlled through IBM Security Access Manager ACL evaluations for defined Roles in the IBM Security Access Manager object space.

## Integration Product Version Information

For information about the supported product versions, see the Release Notes.

## Integration Product Contents

The integration solution is packaged as a .zip file which contains the following:

| File Name  | Description   |
|--|---|
| isam_jboss_int_guide.pdf                         | The integration guide   |
| lib/TAM611/PD.jar                                | Implementation of IBM Security Access Manager Runtime for Java for Oracle Java environments.<br>This implementation requires Oracle Java security implementation and provider classes.<br><b>Supported Oracle JREs</b><br>1.6: TAM611/PD.jar, ISAM7/PD.jar<br>1.7: TAM611/PD.jar, ISAM7/PD.jar, ISAM8/PD.jar, ISAM9/PD.jar<br>1.8: ISAM8/PD.jar, ISAM9/PD.jar |
| lib/ISAM7/PD.jar                                 |   |
| lib/ISAM8/PD.jar                                 |   |
| lib/ISAM8/PDlic.txt                              |   |
| lib/ISAM9/PD.jar                                 |   |
| lib/ISAM9/PDlic.txt                              |   |
| com/ibm/security/webssso/main/TAMJBoss3.1.jar    | Login Module and Policy Provider components for JBoss integration   |
| com/ibm/security/webssso/main/rbpf.jar           | IBM utility library contains provider framework for J2EE role mapping   |
| com/ibm/security/webssso/main/AMJACCProvider.jar | IBM Security Access Manager JACC provider   |
| com/ibm/security/webssso/main/module.xml         | JBoss module definition file for the com.ibm.security.webssso module.   |
| 3.1-JBOSS.README                                 | Release Notes   |
| sample/SecTest.ear                               | Sample application for validating the integration.  |
| sample/cookie-handler.js                         | Sample JavaScript to remove JBoss cookies.  |

Table 1: Integration Package Contents

## Paths and Conventions Used in this guide

Throughout this document, a number of abbreviated configuration values are used to provide a system independent representation. During configuration, replace those values with the appropriate values for your environment. Example values for the abbreviations on a Windows platform are provided in the following table.

| Abbreviation    | Example Value   |
|-----------------|---|
| <JAVA_HOME>     | C:\Program Files\Java\jre1.8  |
| <IBM_JAVA_HOME> | C:\Program Files\IBM\Java80<br>This refers to the IBM JRE installed with Tivoli Policy Directory, not an existing installation. |
| <ISAM_HOME>     | C:\Program Files\Tivoli\Policy Director   |

| Abbreviation             | Example Value  |
|--------------------------|--|
| <JBOSS_HOME>             | C:\Program Files\EAP-6.4.0\jboss-eap-6.4   |
| <WEBSEAL_HOME>           | C:\Program Files\Tivoli\PDWeb  |
| <PD_CONFIG_PROPS>        | file:///C:/Program Files/EAP-6.4.0/jboss-eap-6.4/standalone/configuration/jboss.properties |
| <POLICY_SERVER_HOSTNAME> | The fully qualified domain name of the IBM Security Access Manager Policy Server.          |
| <AUTHZ_SERVER_HOSTNAME>  | The fully qualified domain name of the IBM Security Access Manager Authorization Server.   |
| <JBOSS_EAP_HOSTNAME>     | The fully qualified domain name of the JBoss Enterprise Application Platform server.       |

Table 2: Paths and Conventions

The example commands in this document include space characters in the Windows paths that would typically cause errors during execution. If you are completing this integration in a Windows environment, use the 8.3 formatted paths wherever possible. For example, C:\Program Files is typically represented as C:\Progra~1. Use the /x option of the Windows **dir** command to determine the 8.3 name.

The Distinguished Name (DN) prefix (cn=) and suffix (dc=iswga) values in the sample commands are examples only. You must change these values to suit your environment.

## Preparing the Environment

---

The IBM Security Access Manager Runtime for Java (*PDJRTE*) is required for the JBoss Enterprise Application Platform server to interface with IBM Security Access Manager. The PDJRTE is available in the “IBM Security Access Manager for Web – Application Developer Kit”. For more information, see the support page.

- When *Installing and configuring the IBM Security Access Manager Login Module in Full Mode Configuration*, PDJRTE must be installed and configured when using either IBM and Oracle Java environments.
- When *Installing and configuring the IBM Security Access Manager Login Module in Standalone Mode Configuration*, only PD.jar for IBM or Oracle Java must be added to the classpath of the Java environment.

The Oracle Java Runtime environment requires extra configuration to substitute the IBM Security Access Manager Runtime for Java (PD.jar) with an Oracle compatible version and to add support for IBM Java Security provider classes.

## IBM Security Access Manager 8 and 9 Runtime for Java prerequisites for Oracle

To use the Oracle runtime environment with IBM Security Access Manager 8 or 9 Runtime for Java, you must include a number of IBM Java files.

These files can be copied from an IBM Java Runtime Environment of the same major version as the Oracle Java version.

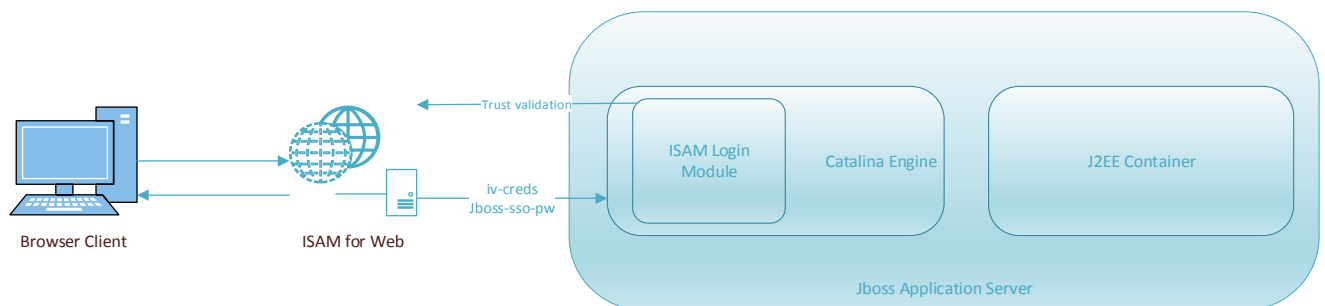
When Full Mode configuration is enabled, you must configure the Oracle Java runtime to use the IBM JCE security provider.

## IBM Security Access Manager Single Sign-On to JBoss Enterprise Application Platform

In this scenario, JBoss Enterprise Application Platform is configured for Single Sign-On with IBM Security Access Manager.

A user authenticates to IBM Security Access Manager, the user identity is passed to JBoss Enterprise Application Platform and a Subject is established for the user inside the Java 2 EE container using an IBM Security Access Manager Login Module implementation.

JBoss uses the Login Module only for resources that require authenticated access, as determined by the constraints in `web.xml` for the application. The container provides the web-based access control for the protected resources and determines which roles are required for access.



Identity assertion is achieved by configuring IBM Security Access Manager to pass the user ID in the `iv-creds` or `iv-user` HTTP request header. The `iv-creds` header contains a serialized data structure that holds the user identity, group memberships, authentication mechanism and other attributes.

It is preferable to use `iv-creds` as it removes the need for the Login Module to access the Authorization Server and user repository to build the in-memory credential for the user. This configuration improves the scalability of the solution.

The `iv-user` HTTP request header contains only the user ID and can be used when the web server has an HTTP request header size limit that affects using `iv-creds`. As no group or role data is passed in `iv-user` HTTP request header, additional access to the IBM Security Access Manager Authorization Server is required for each user's logon process to retrieve group memberships.

- In *Full Mode configuration*, a secondary IBM Security Access Manager authentication mechanism is used to establish trust with the JBoss server. A user ID from the Login Module configuration is combined with a password in the Basic Authentication (BA) header from IBM Security Access Manager and validated against the user repository.
- In *Standalone Mode configuration* no secondary authentication is performed. No configuration of the IBM Security Access Manager Java Runtime is required and `iv-creds` must be used.

The user identity from IBM Security Access Manager is then used to establish the user's identity in the Java 2 EE container.

User roles for the Java 2 EE container are established from the group memberships of the authenticated user, which are contained in the IBM Security Access Manager Principal (PDPrincipal). For example, an IBM Security Access Manager group membership of `Finance` results in the role `Finance` being granted to the authenticated Subject in JBoss.

## Installing and configuring the IBM Security Access Manager Login Module in Full Mode Configuration

Complete the following steps on the JBoss Enterprise Application Platform server:

1. Install an IBM Java Runtime environment available on the installation media or downloaded from IBM FixCentral.
2. Install the IBM Security Access Manager Runtime for Java. If prompted by the installer, configure the IBM Java Runtime environment installed in Step 1 during installation.
3. If using an Oracle Java Runtime Environment:
  - i. Take a backup of `PD.jar` from `<ISAM_HOME>\java\export\pdjrt\`
  - ii. Replace `PD.jar` with the same version of `PD.jar` from `lib\` directory in the integration package.  
Ensure the replaced `PD.jar` has the same permissions as the original, including file ownership.  
On UNIX based systems, the file owner is typically `ivmgr` user.
  - iii. If configuring IBM Security Access Manager Runtime for Java version 8 or 9:
    - a) Take a backup of `PDlic.txt` in `<ISAM_HOME>\.configure\`
    - b) Replace `PDlic.txt` corresponding to the version of the copied `PD.jar` from the integration package.
    - c) Copy the IBM PKCS package from the `<IBM_JAVA_HOME>\jre\lib` directory from an IBM Java installation to the `<JAVA_HOME>\jre\lib\ext` directory of the Oracle Java installation
      - `lib\ibmpkcs.jar`
    - d) Add `<JAVA_HOME>\jre\lib\ext\ibmpkcs.jar` to the `CLASSPATH` environment variable.
    - e) Copy the IBM JCE provider and dependency jar files from the `<IBM_JAVA_HOME>\jre\lib\ext` directory from an IBM Java installation to the `<JAVA_HOME>\jre\lib\ext` directory of the Oracle Java installation
      - `lib\ext\ibmjceprovider.jar`
      - `lib\ext\CmpCrmf.jar`
      - `lib\ext\ibmkeycert.jar`
    - f) Edit `<JAVA_HOME>\jre\lib\security\java.security`.
    - g) Locate the preconfigured security providers and append the IBM JCE provider:
 

```
security.provider.x=com.ibm.crypto.provider.IBMJCE
```

For example:

```
security.provider.1=sun.security.provider.Sun
...
security.provider.10=sun.security.mscapi.SunMSCAPI
security.provider.11=com.ibm.crypto.provider.IBMJCE
```

h) Save and close the `java.security` file.

4. Prepare the Java Runtime environment that the JBoss Enterprise Application Platform server uses. Execute the Java command or the `pdjrtecfg` (.bat for Windows) script contained in `<ISAM_HOME>\sbin\`

#### Using Java

This example command should be executed on a single line in command prompt:

- `"<JAVA_HOME>\jre\bin\java" -Dpd.home="<ISAM_HOME>" -classpath "<ISAM_HOME>\java\export\pdjrte\PD.jar" com.tivoli.pd.jcfg.PDJrteCfg -action config -java_home "<JAVA_HOME>\jre" -host <POLICY_SERVER_HOSTNAME> -port 7135 -config_type full -domain <domain_name>`

#### Using script

Substitute parameter values in the interactive script representing your JBoss and Java environment.

- `pdjrtecfg -action config -interactive`

| Prompt   | Parameter Value   |
|--|---|
| Specify the full path of the Java Runtime Environment (JRE) to configure for Security Access Manager | Value of <code>&lt;JAVA_HOME&gt;</code> .<br>For example: <code>C:\Program Files\Java\jre1.8</code> |
| Enter 'full' or 'standalone' for the configuration type  | full  |
| Policy server host name  | Value of <code>&lt;POLICY_SERVER_HOSTNAME&gt;</code>  |
| Security Access Manager policy server port number  | 7135  |
| Enter the Security Access Manager policy server domain   | <code>&lt;domain_name&gt;</code> .<br>For example: Default  |
| Do you want to use Tivoli Common Directory logging   | n   |

5. Configure a new IBM Security Access Manager Authorization client for the Login Module, used to validate a trusted connection between IBM Security Access Manager and JBoss Enterprise Application Platform.

This example command should be executed on a single line in command prompt:

```
"<JAVA_HOME>\jre\bin\java" -Dpd.cfg.home="<JAVA_HOME>\jre" -classpath
"<ISAM_HOME>\java\export\pdjrte\PD.jar" com.tivoli.pd.jcfg.SvrSslCfg -
action config -admin_id sec_master -admin_pwd <password> -appsvr_id
<jboss-ssso> -port 7201 -mode remote -policysvr
<POLICY_SERVER_HOSTNAME>:7135:1 -authzsvr <AUTHZ_SERVER_HOSTNAME>:7136:1
-host <JBoss_EAP_HOSTNAME> -cfg_file
"<JBoss_HOME>\standalone\configuration\jboss-ssso.properties" -key_file
"<JBoss_HOME>\standalone\configuration\jboss-ssso.ks"
```

Where `<jboss-ssso>` is the name assigned to the authorization client user in IBM Security Access Manager.

## Configuring Trust Validation

You must configure the IBM Security Access Manager instance that you are integrating with the JBoss Enterprise Application Platform server.

A number of the configuration steps can be performed in the web-based Local Management Interface or in the **pdadmin** command line utility. To use the **pdadmin** command line utility"

- Connect to the console of the IBM Security Access Manager for Web Appliance as **admin**. Use the **isam admin** command to start the **pdadmin** command line utility. For example:

```
Welcome to the IBM Security Access Manager
Welcome to the IBM Security Access Manager appliance
Enter "help" for a list of available commands
isamappliance> isam admin
pdadmin>
```

**Note:** The equivalent command for IBM Security Access Manager Web Gateway Appliance is **wga admin**.

Complete the following configuration steps for your target environment, either IBM Security Access Manager for Web or IBM Security Access Manager WebSEAL.

### Using IBM Security Access Manager Web

1. Create a user account that can be used to validate the trusted connection between the IBM Security Access Manager and JBoss Enterprise Application Platform via the **pdadmin** command-line utility.

**NOTE:** If the trusted user's password expires, trust validation will fail. Ensure that you manage the password policy as required.

For example:

```
pdadmin sec_master> user create -no-password-policy jboss-sso-user
uid=jboss-sso-user,dc=iswga "JBoss SSO User" "JBoss SSO User"
<complex-password>
pdadmin sec_master> user modify jboss-sso-user account-valid yes
```

2. Set the **basicauth-dummy-passwd** parameter in the **[junction]** stanza of IBM Security Access Manager WebSEAL configuration.
  1. Open the Web Gateway Appliance Web console.
  2. Select **Secure Web Settings** → **Reverse Proxy**.
  3. Select the reverse proxy instance to configure.
  4. Select **Manage** → **Configuration** → **Edit Configuration File**.
  5. In the Advanced Configuration File Editor locate **basicauth-dummy-passwd** in junction stanza
  6. Replace the password **dummy** with **<complex password>** set while creating jboss-sso-user.

```
[junction]
basicauth-dummy-passwd = <complex-password>
```

**NOTE:** You must restart the Web Reverse Proxy instance after this change.



## Using IBM Security Access Manager WebSEAL

1. Create a user account that can be used to validate the trusted connection between the IBM Security Access Manager and JBoss Enterprise Application Platform via the **pdadmin** command-line utility.

**NOTE:** If the trusted user's password expires, trust validation will fail. Ensure that you manage the password policy as required.

For example:

```
pdadmin sec_master> user create -no-password-policy jboss-sso-user  
uid=jboss-sso-user,dc=iswga "JBoss SSO User" "JBoss SSO User"  
<complex-password>  
pdadmin sec_master> user modify jboss-sso-user account-valid yes
```

2. Set the **basicauth-dummy-passwd** parameter in the **[junction]** stanza of IBM Security Access Manager WebSEAL configuration file in `<WEBSEAL_HOME>\etc` to the password of the new **jboss-sso-user**.

```
[junction]  
basicauth-dummy-passwd = <complex-password>
```

**NOTE:** You must restart the WebSEAL instance after this change.

Proceed to IBM Security Access Manager Configuration on page 19.

## Installing and configuring the IBM Security Access Manager Login Module in Standalone Mode Configuration

When using Standalone Mode configuration the IBM Security Access Manager Runtime for Java does not connect to the IBM Security Access Manager Policy server or Authorization server.

This configuration mode is not suitable if *IBM Security Access Manager Authorization (JACC) for JBoss Enterprise Application Platform* on page 35 is to be configured.

The authenticated user must be passed using the `iv-creds` HTTP request header.

### Using an IBM Java Runtime Environment

1. Install an IBM Java Runtime environment available on the installation media or downloaded from IBM FixCentral.
2. Install the IBM Security Access Manager Runtime for Java. If prompted by the installer, configure the IBM Java Runtime environment installed in Step 1 during installation.

### Using an Oracle Java Runtime Environment

1. Copy `PD.jar` from `lib\` directory in the integration package to `<JAVA_HOME>\jre\lib\ext`.  
Ensure the `PD.jar` has the same permissions as an IBM Security Access Manager Runtime for Java installation, including file ownership.  
On UNIX based systems, the file owner is typically `ivmgr` user.
2. If configuring IBM Security Access Manager Runtime for Java version 8 or 9:
  - i. Copy the IBM PKCS package from the `<IBM_JAVA_HOME>\jre\lib` directory from an IBM Java installation to the `<JAVA_HOME>\jre\lib\ext` directory of the Oracle Java installation
    - `lib\ibmpkcs.jar`
  - ii. Add `<JAVA_HOME>\jre\lib\ext\ibmpkcs.jar` to the `CLASSPATH` environment variable.

**Note:** To assist with large scale, automated deployments, the `PD.jar` and `ibmpkcs.jar` files can be deployed in the JBoss module extension directory. The `module.xml` file must be updated to include references to the required classes from these packages.

Proceed to IBM Security Access Manager Configuration on page 19.

## IBM Security Access Manager Configuration

---

Complete the following configuration steps on Web Gateway Appliance or WebSEAL for integration with JBoss Enterprise Application Platform.

### Web Gateway Appliance Configuration

It is recommended that Forms authentication is used so that single sign-out can be achieved with JBoss Enterprise Application Platform. To disable Basic Authentication and enable Forms Authentication:

1. Open the Web Gateway Appliance Web console.
2. Select **Secure Web Settings** → **Reverse Proxy**.
3. Select the preferred instance.
4. Select **Edit**.
5. Select the **Authentication** tab.
6. Under **Basic Authentication** → **Transport**, select **None**.
7. Under **Forms** → **Forms Authentication**, select **Both**.
8. Click **Save**.
9. Deploy the pending changes.
10. Restart the reverse proxy instance.

If Forms Authentication is not used the browser window must be closed to terminate the IBM Security Access Manager session.

### Junction Management

The selection of the most appropriate junctions for each environment is outside the scope of this integration. See the [IBM Security Access Manager Knowledge Center Junctions](#) page for details of each junction type.

The following example uses junctions to eliminate the requirement of path filtering. The junction name must exactly match the Context Root for JBoss Enterprise Application Platform. For example, **/jbossso**.

**NOTE:** Before creating the junction, ensure that JBoss Enterprise Application Platform has been started and is accessible from IBM Security Access Manager.

Create a junction to the JBoss Enterprise Application Platform.

1. Open the Web Gateway Appliance Web console.
2. Select **Secure Web Settings** → **Reverse Proxy**.
3. Select the preferred instance.
4. Select **Manage** → **Junction Management**.
5. Select **New** → **Standard Junction**.
6. On the Junction tab
  - a. Add the junction name **/jbossso**
  - b. For Junction Type, select **TCP**.

7. On the Servers tab
  - a. Click **New** to add a target backend server.
  - b. Specify the connection details of the JBoss Enterprise Application Platform URL:
    - Hostname: jboss.ibm.com
    - TCP or SSL Port: 443
    - Virtual Host: webseal.ibm.com
  - c. Click **Save**.
8. On the Identity tab
  - a. Select **HTTP Header Identity Information: IV-USER and IV-CREDS**
  - b. Under **HTTP Basic Authentication Header**, select **Supply**
9. Click **Save**.
10. Create a junction for each of the JBoss Enterprise Application Platform application paths.

## Session Management

When using Forms authentication, JBoss session cookies are not destroyed when the user is logged out of IBM Security Access Manager. This makes it possible for another user to reuse the same browser page and login to IBM Security Access Manager, creating a new session, while maintaining original user's JBoss session for junction applications.

- If the cookies generated by the JBoss application are not required at the client browser, proceed to *Using Managed Cookie List*.
- If the cookies generated by the JBoss application are required at the client browser, proceed to *Modifying template pages*.

### Using Managed Cookie List

1. Open the Web Gateway Appliance Web Console.
2. Select **Secure Web Settings** → Reverse Proxy
3. Select the preferred instance
4. Click **Edit**.
5. On the **Junction** tab:
6. In the **Managed Cookie List** text box, add a comma-separated list of cookies used by JBoss server applications. For example: `JSESSIONID`
7. Deploy the pending changes.
8. Restart the reverse proxy instance.

Proceed to *Configuring JBoss Enterprise Application Platform* on page 27.

### Modifying template pages

1. Open the Web Gateway Appliance Web Console.
2. Select **Secure Web Settings** → Reverse Proxy
3. Select the preferred instance.
4. Select **Manage** → **Management Root**
5. Expand **Management** → **C**
6. Select **login.html**.
  - a. Select **File** → **Open**
  - b. Insert the contents of the `sample/cookie-handler.js` file inside the `<HEAD>` tags. Modify the section in **bold** below for your specific cookie names and paths.

For example:

```
<script type="text/javascript">
  function delete_cookie(name, path) {
    // Expire the cookie
    var cookie_string = name + "; expires=Thu, 01 Jan 1970
00:00:00 UTC";
    if (path != null) {
      cookie_string += "; path=" + path;
    }
    document.cookie = cookie_string;
  }
  if ("%OLDSESSION%" == "1") {
    delete_cookie("JSESSIONID", "/");
  }
</script>
```

c. Click **Save**.

7. Repeat Step 6 above for **logout.html** also.
8. Deploy the pending changes.
9. Restart the reverse proxy instance.

Proceed to *Configuring JBoss Enterprise Application Platform* on page 27.

## WebSEAL Configuration

It is recommended that Forms authentication is used so that single sign-out can be achieved with JBoss Enterprise Application Platform. To enable Forms Authentication and disable Basic Authentication:

1. Stop the WebSEAL server.
2. Edit the `<PDWEB_HOME>/etc/webseald-<instance-name>.conf` file.  
For example: `/opt/pdweb/etc/webseald-default.conf`
3. Locate the **[forms]** stanza.
4. Update forms-auth to **https**.
5. Locate the **[ba]** stanza.
6. Update ba-auth to **none**.
7. Save the file.
8. Restart the WebSEAL instance.

## Junction Creation

The selection of the most appropriate junctions for each environment is outside the scope of this integration. See the IBM Security Access Manager Knowledge Center Junctions page for details of each junction type.

The following example uses junctions to eliminate the requirement of path filtering. The junction name must exactly match the Context Root for JBoss Enterprise Application Platform. For example, **/jbossso**.

1. Using the pdadmin command-line utility, determine the WebSEAL server name. For example:

```
pdadmin sec_master> server list <instance-name>-webseald-
<webseal-host-name>
```

- Using the pdadmin command-line utility, create a junction to the JBoss Enterprise Application Platform Gateway URL.

```
pdadmin sec_master> server task instance-webseald-server_name
create -t ssl -c iv_creds,iv_user -h <JBoss_EAP_HOSTNAME> -p
<JBoss_EAP_PORT> -v webseal_fqdn:port_no -b supply -f /jbossso
```

For example:

```
pdadmin sec_master> server task jboss-webseald-isam create -t
ssl -c iv_creds,iv_user -h jboss.ibm.com -p 443 -v
webseal.ibm.com:443 -b supply -f /jbossso
```

- Create a junction for each of the JBoss Enterprise Application Platform application paths.

## Session Management

When using forms authentication, sessions can be terminated using the `/pkmslogout` feature or through session timeout or expiry. When the user is logged out of IBM Security Access Manager, the JBoss session cookies are not destroyed, making it possible for another user to use the same browser window, login to IBM Security Access Manager to create a new session, maintaining the original user's JBoss session for junction applications.

If the cookies by the JBoss application are not required at the client browser, proceed to *Using Managed Cookie List*.

If the cookies by the JBoss application are required at the client browser, proceed to *Modifying Template Pages*.

### Using Managed Cookie List

- Edit the `<PDWEB_HOME>/etc/webseald-<instance-name>.conf` file. For example: `/opt/pdweb/etc/webseald-default.conf`
- Locate the **[junction]** stanza.
- Update `managed-cookie-list` to add a comma-separated list of cookies used by the JBoss server applications.

For example:

```
[junction]
managed-cookie-list = JSESSIONID
```

- Save and close the file.
- Restart the WebSEAL server.

Proceed to *Configuring JBoss Enterprise Application Platform* on page 27.

## Modifying Template Pages

- Locate the WebSEAL management page directory of your reverse proxy instance. For example, `/opt/pdweb/www-default/lib/html/C/`
- Edit **Login.html**
  - Insert the contents of `sample/cookie-handler.js` file inside the **<HEAD>** tags. Modify the section in **bold** below for your specific cookie names and paths.

For example:

```
<script type="text/javascript">
    function delete_cookie(name, path) {
```

```

        // Expire the cookie
        var cookie_string = name + "="; expires=Thu, 01 Jan 1970
00:00:00                UTC";
        if (path != null) {
            cookie_string += "; path=" + path;
        }
        document.cookie = cookie_string;
    }
    if ("%OLDSESSION%" == "1") {
        delete_cookie("JSESSIONID", "/");
    }
</script>

```

b. Save and close the file.

### 3. Edit **logout.html**

a. Insert the contents of `sample/cookie-handler.js` file inside the **<HEAD>** tags. Modify the section in **bold** below for your specific cookie names and paths.

For example:

```

<script type="text/javascript">
    function delete_cookie(name, path) {
        // Expire the cookie
        var cookie_string = name + "="; expires=Thu, 01 Jan 1970
00:00:00                UTC";
        if (path != null) {
            cookie_string += "; path=" + path;
        }
        document.cookie = cookie_string;
    }
    if ("%OLDSESSION%" == "1") {
        delete_cookie("JSESSIONID", "/");
    }
</script>

```

b. Save and close the file.

### 4. Restart the WebSEAL instance.

Proceed to *Configuring JBoss Enterprise Application Platform* on page 27.

## IBM Security Access Manager for Mobile Configuration (optional)

IBM Security Access Manager for Mobile can be used to evaluate Context-Based Access (CBA), also known as Risk-Based Access (RBA) decisions for requests from IBM Security Access Manager for Web.

The web reverse proxy acts as a Policy Enforcement Point (PEP), which will call the runtime security service's Policy Decision Point (PDP) using an Authorization Service to evaluate policy and provide an access decision. The web reverse proxy can enforce obligations or additional authentication constraints based on context data including request parameters.

The configuration in this chapter configures the sample application `SecTest.ear` which demonstrates a money transfer request, which has an obligation to step up the user's authentication level with a One-Time Password (OTP).

### Mobile Authorization Decision Point Configuration

Ensure you configure Web Gateway Appliance to use the Mobile Appliance by executing the **isamcfg** tool. Refer to the IBM Knowledge Centre for more information on this configuration.

### Configuring Web Reverse Proxy

For the sample application, the HTTP POST parameters must be forwarded by the reverse proxy instance to the mobile appliance.

1. Open the Web Gateway Appliance Web Console.
2. Select **Secure Web Settings** → **Reverse Proxy**.
3. Select the preferred instance.
4. Select **Manage** → **Configuration** → **Edit Configuration File**.
5. Locate the **[azn-decision-info]** stanza.
6. Assign the sample application's four post parameters, **amount**, **fromAccount**, **toAccount** and **action** URNs.

For example:

```
[azn-decision-info]
urn:ibm:security:jboss:amount = post-data:amount
urn:ibm:security:jboss:fromAccount = post-data:fromAccount
urn:ibm:security:jboss:toAccount = post-data:toAccount
urn:ibm:security:jboss:action = post-data:action
```

7. Locate the **[user-attribute-definitions]** stanza.
8. Assign the datatype of the **amount** parameter to double.

For example:

```
[user-attribute-definitions]
urn:ibm:security:jboss:amount.datatype = double
```

**Note:** An unspecified datatype will default to String. An unspecified category will default to **Environment**.

9. Click **Save**.
10. Deploy the pending changes.
11. Restart the reverse proxy instance.



## Creating the Policy Attributes

Create custom attributes for the sample application POST data.

1. Open the Mobile Gateway Appliance Web Console.
2. Select **Secure Mobile Settings** → **Attributes**.
3. Click **New Attribute**.
4. Add the **transferAmount** attribute:
  - a. Name: JBoss Transfer Amount
  - b. Identifier: urn:ibm:security:jboss:amount
  - c. Description: The transfer amount for the JBoss integration sample application.
  - d. Issuer: <blank>
  - e. Category: Environment
  - f. Date Type: Double
  - g. Matcher: exact\_match
  - h. Type: Policy
  - i. Storage Domain: None
5. Click **Save**.
6. Add the **fromAccount** attribute:
  - a. Name: JBoss From Account
  - b. Identifier: urn:ibm:security:jboss:fromAccount
  - c. Description: The account to perform a transfer from in the JBoss integration sample application
  - d. Issuer: <blank>
  - e. Category: Environment
  - f. Date Type: Double
  - g. Matcher: exact\_match
  - h. Type: Policy
  - i. Storage Domain: None
7. Click **Save**.
8. Add the **toAccount** attribute:
  - a. Name: JBoss To Account
  - b. Identifier: urn:ibm:security:jboss:toAccount
  - c. Description: The account to perform a transfer to in the JBoss integration sample application
  - d. Issuer: <blank>
  - e. Category: Environment
  - f. Date Type: Double
  - g. Matcher: exact\_match
  - h. Type: Policy
  - i. Storage Domain: None
9. Click **Save**.
10. Add the **Action** attribute:
  - a. Name: JBoss Action
  - b. Identifier: urn:ibm:security:jboss:action
  - c. Description: The transaction being performed in the JBoss integration sample application
  - d. Issuer: <blank>
  - e. Category: Environment
  - f. Date Type: String
  - g. Matcher: exact\_match
  - h. Type: Policy
  - i. Storage Domain: None
11. Click **Save**.

12. Deploy the pending changes.

## Policy Authoring

Policy must be authored in order to enforce an HMAC-based One-Time Password (HOTP) or Time-sensitive HOTP (TOTP) obligation based on the value of the **amount** parameter when the **action** type is "Transfer".

In this example, the user will be required to step up their authentication level when the transfer amount is \$100 or more.

1. Open the Mobile Gateway Appliance Web console.
2. Select **Secure Mobile Settings** → **Access Control**.
3. Select the **Policies** tab.
4. Click **Create Policy**.
5. Add the **Access Policy**
  - a. Name: JBoss Transfer Setup
  - b. Description: Force the user to step-up when they try to transfer more than \$100 in one transaction
  - c. Select **Precedence** → **First**.
  - d. Click **Add Rule**.
  - e. Create the rule by selecting the attributes and comparing to the given value.
 

```

1. If JBoss Action is present and
JBoss Action = "Transfer" and
JBoss Transfer Amount is present and
JBoss Transfer Amount >= 100
Then Permit with Authentication HOTP One-time Password
          
```
  - f. Click **OK**.
  - g. Select the drop down for **Add Rule**, then **Unconditional Rule**. Set this to *Permit*.
  - h. Click **OK**.
6. Click **Save**.
7. Deploy the pending changes.

## Attaching the Policy

The access policy can now be attached to the protected resource, which triggers when a request for that object is intercepted by IBM Security Access Manager. The policy is then evaluated and the appropriate action taken.

1. Open the Mobile Gateway Appliance Web console.
2. Select **Secure Mobile Settings** → **Access Control**.
3. Select the **Resources** tab.
4. Select `<webseal-host-name>-<instance-name>`
5. Click **Add Resource**.
6. Click **Browse**.
7. Select `/jbossso`.
8. Click **OK**.
9. Select `<webseal-host-name>-<instance-name>/jbossso`
10. Click **Attach**.
11. Select **Policies**.
12. **Check the JBoss Transfer Stepup policy.**
13. Click **OK**.
14. Click **Publish**.

## Configuring JBoss Enterprise Application Platform for Single Sign On

Login modules can be chained to allow users not authenticated by IBM Security Access Manager to access protected applications.

You can chain login modules to maintain direct access applications, so that the applications can be accessed by users who are not authenticated by IBM Security Access Manager.

The following examples use IBM Security Access Manager Login Module, which restricts access to only IBM Security Access Manager authenticated users.

You can configure additional login modules and chain them together. Use the *flag* property to specify either: *required*, *requisite*, *sufficient* or *optional*, depending on your requirements. Typically, *sufficient* is the most suitable for chaining login modules. For example:

```
<login-module code="com.ibm.security.websso.TAMLoginModule" flag="required"
module="com.ibm.security.websso">
```

For more information about the *flag* property for login modules, see the JBoss Enterprise Application Platform Security Domain Schema documentation.

The `<security-domain>` determines which login module is used to provide authentication for web applications. You can use either of the following configurations:

- Add a security-domain and update your web applications to reference the new domain.
- Modify an existing security-domain for already deployed applications to use the IBM Security Access Manager Login Module.

You can modify the `<security-domain name="other">` entry to control the default behaviour. The JBoss Enterprise Application Platform server uses this security-domain entry in the following situations:

- The web application does not specify a security-domain.
- The web application specifies a security-domain that does not have a matching `<security-domain>` entry in the security subsystem.

Generally, it is best to use a new security-domain specifically for IBM Security Access Manager. However, when you are migrating existing applications, you can reduce the number of changes that are required in the web application configuration files by modifying an existing security-domain.

Multiple web applications can reference the same security-domain instance.

Add a new `<security-domain>` section for the Login Module in the file `<JBOSS_HOME>\standalone\configuration\standalone.xml`. Alternatively, you can update the `<authentication>` block of an existing custom security-domain.

## Installing and Configuring the IBM Security Access Manager Login Module

Complete the following steps on the JBoss Enterprise Application Platform server:

- Complete all of the steps in *Preparing the Environment* on page 12.
- Complete all of the steps in *IBM Security Access Manager Single Sign-On to JBoss Enterprise Application Platform* on page 14.

- Complete all of the steps in *IBM Security Access Manager Configuration* on page 19.
- Optionally, complete the steps in *IBM Security Access Manager for Mobile Configuration (optional)* on page 24.

Each application can either reuse the authorization client that was created for the sample application or create a new authorization client.

You can configure server wide single sign on, or individual applications. Complete the configuration in either:

- *Configuring Application Server Wide Single Sign-On;*  
or;
- *Configuring Individual Applications for Single Sign-On*

## Configuring Application Server Wide Single Sign-On

The sample application `SecTest.ear` included with this integration can be used to validate the success of your integration. No additional configuration of the application is required; you need only add the Login Module configuration.

The following steps detail how to deploy and configure this sample deployment for Single Sign-On (SSO) with IBM Security Access Manager. This example is for the JBoss Enterprise Application Platform instance that is running as a stand-alone server.

1. Add the required `<authentication>` configuration to the `<security-domains>` section of the file `<JBOSS_HOME>\standalone\configuration\standalone.xml`. Modify the name of the `<security-domain>` to a named match, for example `SecTest` for the sample application, or other as a catchall for unmatched names.

### Full Mode Configuration

If you completed the steps in *Installing and configuring the IBM Security Access Manager Login Module in Full Mode Configuration* on page 14 add all parameters for the Login Module in the `<authentication>` stanza.

```
<security-domain name="SecTest">
  <authentication>
    <login-module code="com.ibm.security.websso.TAMLoginModule"
      flag="required" module="com.ibm.security.websso">
      <module-option name="pdconfig"
        value="file:///<JBOSS_HOME>/standalone/configuration/jboss-
          sso.properties"/>
      <module-option name="idType" value="iv-creds"/>
      <module-option name="loginID" value="jboss-sso-user"/>
      <module-option name="buildRolesFrom" value="PDPrincipal"/>
      <module-option name="reqHdrList" value="iv-user"/>
    </login-module>
  </authentication>
</security-domain>
```

### Standalone Mode Configuration

If you completed the steps in *Installing and configuring the IBM Security Access Manager Login Module in Standalone Mode Configuration* on page 18 disable the trust validation for the Login Module in the `<authentication>` stanza.

```
<security-domain name="SecTest">
  <authentication>
    <login-module code="com.ibm.security.websso.TAMLoginModule"
      flag="required" module="com.ibm.security.websso">
      <module-option name="ssoPwdExpiry" value="-1"/>
    </login-module>
  </authentication>
</security-domain>
```

Header based trust validation mechanisms including `reqHdrList`, `checkViaHeader` and `viaDepth` may be used in Standalone Mode configuration.

Refer to *TAM Login Module Configuration Options* on page 43 for configuration options.

2. Copy the `com` directory from the integration package to `<JBOSS_HOME>\modules\`.
3. Copy the `SecTest.ear` file from the installation package sample directory to `<JBOSS_HOME>\standalone\deployments`.

## Configuring Individual Applications for Single Sign-On

You can configure a new or existing web application for JBoss for IBM Security Access Manager Single Sign-On (SSO). In *Configuring Application Server Wide Single Sign-On* on page 28, the sample application is already configured and no changes to the deployment files of the sample application are required.

When you are migrating existing applications, you must change the application deployment files either on the server, in the application or both. The required changes depend on the current deployment and which values can be modified.

You can reduce the overall change management process by making server configuration changes in `standalone.xml` file instead of application file changes in `web.xml` and `jboss-web.xml`.

For new applications, the following implementation approach can minimize the required changes:

- In the `web.xml` file, use the BASIC authentication type.
- In the `standalone.xml` file, configure a security-domain on the JBoss server to use the IBM Security Access Manager Login Module and reference this security-domain in your application.

This implementation approach requires an update in the JBoss server only.

The following steps are specific to a JBoss Enterprise Application Platform instance that is running as a standalone server. However, these steps are also valid for domain deployments when the paths are updated accordingly.

You must update each Java application that you want to configure for IBM Security Access Manager Single Sign-On to use Basic Authentication and a security-domain configured for IBM Security Access Manager Login Module.

1. Add the required `<authentication>` configuration to the `<security-domains>` section of the file `<JBOSS_HOME>\standalone\configuration\standalone.xml`. Modify the name of the `<security-domain>` to a named match, for example `ISAMWebSSO`.

### Full Mode Configuration example

If you completed the steps in *Installing and configuring the IBM Security Access Manager Login Module in Full Mode Configuration* on page 14 add all parameters for the Login Module in the `<authentication>` stanza.

```

<security-domain name="ISAMWebSSO">
  <authentication>
    <login-module code="com.ibm.security.websso.TAMLoginModule"
flag="required" module="com.ibm.security.websso">
      <module-option name="pdconfig"
value="file:///<JBOSS_HOME>/standalone/configuration/jboss-
sso.properties"/>
      <module-option name="idType" value="iv-creds"/>
      <module-option name="loginID" value="jboss-sso-user"/>
      <module-option name="buildRolesFrom" value="PDPrincipal"/>
      <module-option name="reqHdrList" value="iv-user"/>
    </login-module>
  </authentication>
</security-domain>

```

### Standalone Mode Configuration example

If you completed the steps in *Installing and configuring the IBM Security Access Manager Login Module in Standalone Mode Configuration* on page 18 disable the trust validation for the Login Module in the `<authentication>` stanza.

```

<security-domain name="ISAMWebSSO">
  <authentication>
    <login-module code="com.ibm.security.websso.TAMLoginModule"
flag="required" module="com.ibm.security.websso">
      <module-option name="ssoPwdExpiry" value="-1"/>
    </login-module>
  </authentication>
</security-domain>

```

Header based trust validation mechanisms including `reqHdrList`, `checkViaHeader` and `viaDepth` may be used in Standalone Mode configuration.

Refer to *TAM Login Module Configuration Options* on page 43 for configuration options.

2. Enable the BASIC `<auth-method>` in the `<login-config>` of the `web.xml` file for the application. For example:

```

<web-app>
  ...
  <login-config>
    <auth-method>BASIC</auth-method>
    <realm-name>IBM Security Access Manager</realm-name>
  </login-config>
  ...
</web-app>

```

In this configuration, trust can be established by validating the user ID from the BA header and the WebSEAL dummy password, which is specified when you create the trusted user.

3. Ensure that `<security-domain>` of the `jboss-web.xml` file for application corresponds to the `<security-domain>` that is configured to use the IBM Security Access Manager Login Module. For example:

```

<jboss-web>
  ...
  <security-domain>ISAMWebSSO</security-domain>

```

```
...
</jboss-web>
```

If no match is found for the security-domain, or the domain is not specified, the Login Module that is configured for the `other` security-domain in the JBoss server configuration is called.

## Configuring Java Web Applications for Single Sign On

Ensure the Java Web Application correctly handles single sign out and any class dependencies are included in `module.xml`.

1. Applications may provide a logout function by using `session.invalidate()` to terminate a user session. However, this does not log the user out of their WebSEAL session. When an application is using IBM Security Access Manager, the user is automatically signed on to the application again. The user must instead be redirected to IBM Security Access Manager *pkmslogout* page after logging out from the JBoss application.

The sample application in the integration includes a `logout.jsp` file that can be used to log the user out of JBoss and redirect to WebSEAL's *pkmslogout* page. To enable Single Sign-Off, modify any logout pages for the application to perform a redirect to `"/../pkmslogout"` after performing any required logout tasks. **Note:** This configuration is for a Standard Junction. If using Forms Authentication, the URL can be simplified to `"/pkmslogout"`.

For example:

```
<%
    if(session!=null) {
        session.invalidate();
    }
    response.sendRedirect("/../pkmslogout");
%>
```

Inside a Java servlet request handler:

```
if(request.getSession(false)!=null) {
    request.getSession(false).invalidate();
}
response.sendRedirect("/../pkmslogout");
```

2. Ensure that any dependencies for the web application are resolved by referencing the appropriate modules on `module.xml`.  
For example, the sample application references HTTP Servlet Request classes, which are resolved by **com.ibm.security.websso** module as it references the **javax.servlet.api** module to satisfy the dependencies.
3. **Note:** This step is included for completeness. It is not related to the login module security domain.

Add any required module dependencies to `jboss-deployment-structure.xml` file for the application. For example:

```
<jboss-deployment-structure xmlns="urn:jboss:deployment-
structure:1.1">
    <ear-subdeployments-isolated>false</ear-subdeployments-isolated>
    <deployment>
```



```
<dependencies>
<module name="com.ibm.security.websso" slot="main"/>
</dependencies>
</deployment>
</jboss-deployment-structure>
```

## Mapping IBM Security Access Manager Groups to Roles in Java Web Application

You must create IBM Security Access Manager groups corresponding to all of the role definitions in the `web.xml` file for the application. At runtime, the Login Module adds roles to the JBoss Principal with the same role-name as the group names for which the authenticated user is a member.

IBM Security Access Manager Access Control Lists (ACLs) and Protected Object Policies (POPs) can still be attached in the WebSEAL object space. You can use these settings to provide fine grained access control or apply extra security constraints beyond the roles, without modifying the application directly.

For example, an application `web.xml` contains the following roles:

```
<security-role>
  <role-name>SecAdmins</role-name>
</security-role>
<security-role>
  <role-name>SecUsers</role-name>
</security-role>
```

This application requires the corresponding IBM Security Access Manager groups:

```
pdadmin sec_master> group create SecAdmins cn=secadmins,dc=iswga SecAdmins
pdadmin sec_master> group create SecUsers cn=secusers,dc=iswga SecUsers
```

The required users must be added to the groups:

```
pdadmin sec_master> group modify SecAdmins add secadmin
pdadmin sec_master> group modify SecUsers add secadmin
pdadmin sec_master> group modify SecUsers add secuser
```

## Configuring IBM Security Access Manager for the Sample Application SecTest.ear

You must configure IBM Security Access Manager instance that you are integrating with the JBoss Enterprise Application Platform server to provide the necessary HTTP headers for trust validation and user identity assertion.

### Mapping IBM Security Access Manager Groups to Roles

The IBM Security Access Manager Login Module adds roles to the JBoss Principal with the same role name as the IBM Security Access Manager group name.

Execute the following commands to create the required users and groups in IBM Security Access Manager:



```
pdadmin sec_master> user create secadmin cn=secadmin,dc=iswga secadmin
password
pdadmin sec_master> user modify secadmin account-valid yes
pdadmin sec_master> user create secuser cn=secuser,dc=iswga secuser
password
pdadmin sec_master> user modify secuser account-valid yes
pdadmin sec_master> user create getuser cn=getuser,dc=iswga getuser
password
pdadmin sec_master> user modify getuser account-valid yes
pdadmin sec_master> user create postuser cn=postuser,dc=iswga postuser
password
pdadmin sec_master> user modify postuser account-valid yes
pdadmin sec_master> group create SecAdmins cn=secadmins,dc=iswga SecAdmins
pdadmin sec_master> group modify SecAdmins add secadmin
pdadmin sec_master> group create SecUsers cn=secusers,dc=iswga SecUsers
pdadmin sec_master> group modify SecUsers add secadmin
pdadmin sec_master> group modify SecUsers add secuser
pdadmin sec_master> group create Getters cn=getters,dc=iswga Getters
pdadmin sec_master> group modify Getters add getuser
pdadmin sec_master> group create Posters cn=posters,dc=iswga Posters
pdadmin sec_master> group modify Posters add postuser
```

## Starting the JBoss Server

The integration steps for the JBoss server are now complete. Start the JBoss server to complete the integration with IBM Security Access Manager. The sample application is automatically deployed.

Run either `standalone.bat` or `standalone.sh` in `<JBOSS_HOME>\bin` to start the JBoss Enterprise Application Platform standalone server.

## Validating the Integration

To confirm that your IBM Security Access Manager integration with JBoss Enterprise Application Platform has been successfully configured, attempt to access the JBoss application through the IBM Security Access Manager instance.

`http[s]://<webseal.server.fqdn>:<port>/jbosssso/SecTestWeb`

For example:

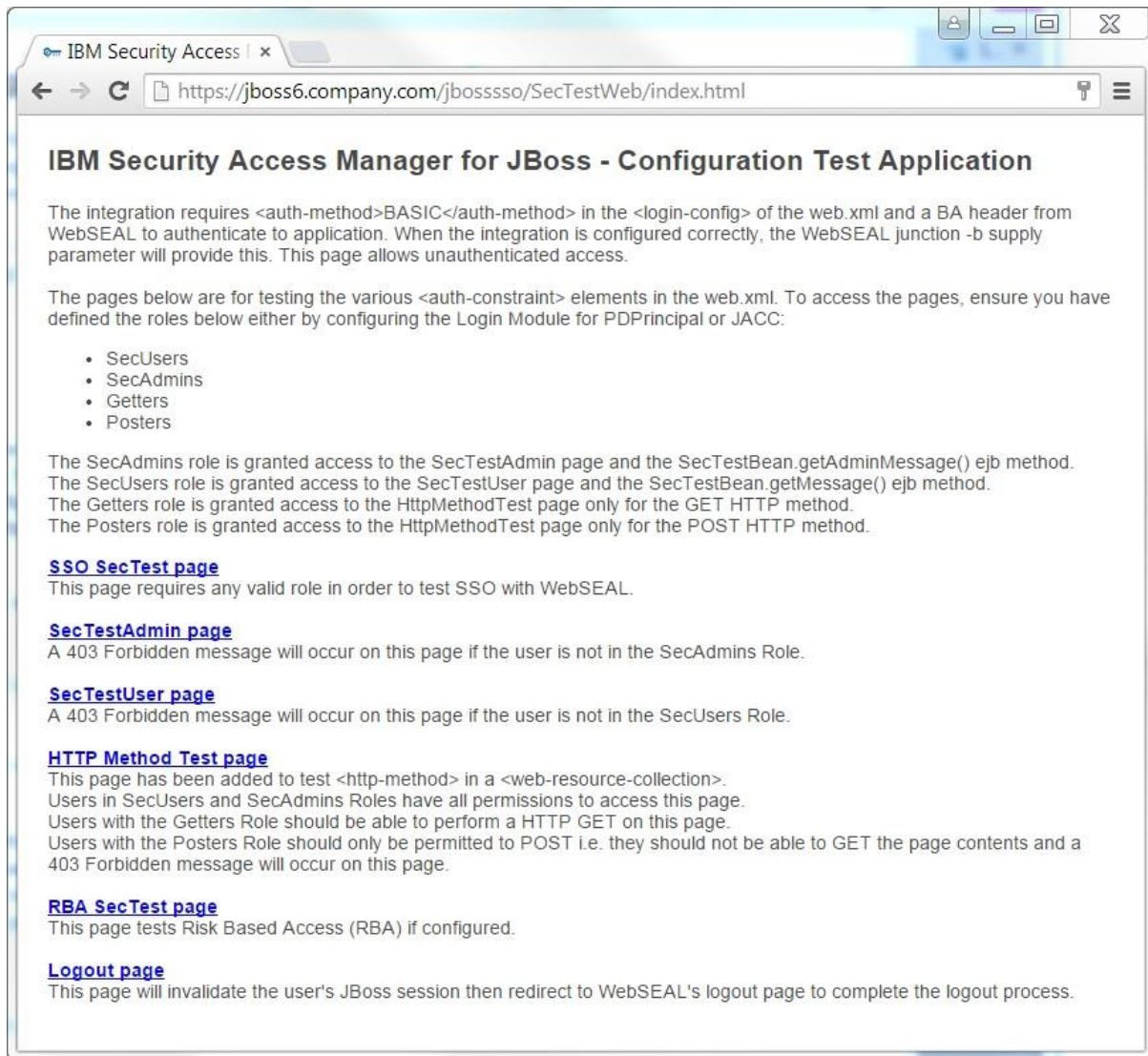
`https://webseal.ibm.com:443/jbosssso/SecTestWeb`

The following welcome page is displayed when a user successfully signs on the application. Follow the instructions on the Test Application welcome page to validate the access permissions for each user.

Ensure that the correct access is granted to all parts of the application. This access is based on the authenticated user and IBM Security Access Manager group to role mapping.

Ensure that the correct access is granted to all parts of the application based on the following factors:

- The authenticated user.
- The IBM Security Access Manager group to role mapping.
- Any additional ACL or POP assignments.



## IBM Security Access Manager Authorization (JACC) for JBoss Enterprise Application Platform

This scenario can only be used when *Installing and configuring the IBM Security Access Manager Login Module in Full Mode Configuration* on page 14.

This scenario provides fine-grained authorization of J2EE objects, such as EJBs, evaluated by IBM Security Access Manager. The Java Authorization Contract and Containers (JACC) standard is the mechanism through which authorization decisions are passed from JBoss Enterprise Application Platform to the IBM Security Access Manager JACC Policy provider.

Delegated authorization for applications can be configured for individual applications or application server wide and both configuration options are detailed in this chapter.

The IBM Security Access Manager JACC Policy provider can be integrated with any login module that correctly authenticates a Subject to the J2EE container.

### Access Manager JACC Policy Provider Installation and Configuration

1. Complete all steps in *Configuring JBoss Enterprise Application Platform for Single Sign On* on page 27 .
2. Copy the `com` folder from the integration package to `<JBOSS_HOME>\modules` if you have not yet done so.
3. Copy the contents of the `conf` folder from the integration package to `<JBOSS_HOME>\bin`.
4. Configure a new IBM Security Access Manager authorization client for JACC Policy Provider, used to store and retrieve the policy, authorize EJB permissions and populate Roles.

This example command should be executed on a single line:

```
"<JAVA_HOME>\jre\bin\java" -Dpd.cfg.home="<JAVA_HOME>\jre" -classpath  
"<ISAM_HOME>\java\export\pdjrte\PD.jar" com.tivoli.pd.jcfg.SvrSslCfg  
-action config -admin_id sec_master -admin_pwd <password> -appsvr_id  
jboss-jacc -port 7202 -mode local -policysvr  
<POLICY_SERVER_HOSTNAME>:7135:1 -authzsvr  
<AUTHZ_SERVER_HOSTNAME>:7136:1 -host <JBOSS_EAP_HOSTNAME> -cfg_file  
<JBOSS_HOME>\standalone\configuration\jboss-jacc.properties -key_file  
<JBOSS_HOME>\standalone\configuration\jboss-jacc.ks -dbdir  
<JBOSS_HOME>\standalone\configuration\ -domain Default
```

5. Grant membership to the `iv-admin` group to the JACC authorization client user account created when executing the `com.tivoli.pd.jcfg.SvrSslCfg` Java program.

```
pdadmin sec_master> group modify iv-admin add jboss-  
jacc/<JBOSS_EAP_HOSTNAME>
```

6. Open `<JBOSS_HOME>\bin\amwas.properties` for editing.
7. Update the `com.tivoli.pd.as.rbpf.AmasSession.LoggingURL` property to the full (Java representation) URL file path of `amwas.pdjog.properties` file.

```
com.tivoli.pd.as.rbpf.AmasSession.LoggingURL=file:///<JBOSS_HOME>/bin/  
/amwas.pdjlog.properties
```

- Update the `com.tivoli.pd.as.rbpf.AmasSession.CfgURL` property to the full (Java representation) URL file path of `jboss-jacc.properties` file that was created when executing the `com.tivoli.pd.jcfg.SvrSslCfg` Java program.

```
com.tivoli.pd.as.rbpf.AmasSession.CfgURL=file:///<JBOSS_HOME>/standalone/configuration/jboss-jacc.properties
```

- Save and close the file.

## JBoss Enterprise Application Platform Configuration

Configured the JBoss Enterprise Application Platform server to load the IBM Security Access Manager Policy Provider instead of system default.

- Open the file  
`<JBOSS_HOME>\modules\system\layers\base\org\jboss\as\security\main\module.xml`

- Under the `<dependencies>` section, insert the IBM Security Access Manager login module.

```
<module name="com.ibm.security.websso" export="true"/>
```

- Save and close the file.
- Edit the JBoss launch script to set the IBM Security Access Manager as the JVM JACC Policy Provider. This will be either `standalone.conf` or `domain.conf` depending on your configuration, and will end with `.bat` for Windows installations. For example, in a Windows standalone configuration open `<JBOSS_HOME>\bin\standalone.conf.bat` for editing.

- On a UNIX system, add the following lines to the end of the file. Ensure the `JAVA_OPTS` assignment is on a single line.

```
# ISAM JACC Provider
JAVA_OPTS="$JAVA_OPTS -
Djavax.security.jacc.policy.provider=com.ibm.security.websso.TA
MJACCJBossPolicy -
Djavax.security.jacc.PolicyConfigurationFactory.provider=com.ti
voli.pd.as.jacc.TAMPolicyConfigurationFactory"
```

- On a Windows system, add the following lines above `:JAVA_OPTS_SET`. Ensure the `set` command is on a single line.

```
rem # ISAM JACC Provider
set "JAVA_OPTS=%JAVA_OPTS% -
Djavax.security.jacc.policy.provider=com.ibm.security.websso.TA
MJACCJBossPolicy -
Djavax.security.jacc.PolicyConfigurationFactory.provider=com.ti
voli.pd.as.jacc.TAMPolicyConfigurationFactory"
```

- Save and close the file.

## Server-wide Authorization Configuration

1. Ensure that you have completed all configuration steps as described in *Configuring Application Server Wide Single Sign-On* on page 28.
2. Open the server configuration file located at  
`<JBOSS_HOME>\standalone\configuration\standalone.xml`.
3. Locate the security domain configured in *Configuring Application Server Wide Single Sign-On*.  
For example: `<security-domain name="SecTest">`.
4. Add an additional `<authorization>` section to enable JACC for the specified security domain.  

```
<authorization>
  <policy-module code="JACC" flag="required"/>
</authorization>
```
5. Under the `<login-module>` section, modify the `buildRolesFrom` module option to JACC. For example:  

```
<module-option name="buildRolesFrom" value="JACC"/>
```
6. Save and close the file.

Proceed to *IBM Security Access Manager Configuration* on page 38.

## Individual Application Authorization Configuration

1. Ensure that you have completed all configuration steps *Configuring Individual Applications for Single Sign-On* on page 29.
2. Open the server configuration file located at  
`<JBOSS_HOME>\standalone\configuration\standalone.xml`.
3. Locate the security domain configured in *Configuring Individual Applications for Single Sign-On*.  
For example, `<security-domain name="ISAMWebSSO">`.
4. Add an additional `<authorization>` section to enable JACC for the specified security domain.  

```
<authorization>
  <policy-module code="JACC" flag=" required"/>
</authorization>
```
5. Under the `<login-module>` section, modify the `buildRolesFrom` module option to JACC. For example:  

```
<module-option name="buildRolesFrom" value="JACC"/>
```
6. Save and close the file.

## 7. For each installed WAR component:

- a. Locate and open `jboss-web.xml` inside the deployment's `WEB-INF` folder.
- b. Ensure that the `<security-domain>` element under `<jboss-web>` references the security domain configured.

For example:

```
<security-domain>ISAMWebSSO</security-domain>
```

- c. Under `<jboss-web>` add the element  

```
<use-jboss-authorization>true</use-jboss-authorization>
```
- d. Save and close the file.

## 8. For each installed EJB component:

- a. Locate and open `jboss-ejb3.xml` inside the deployment's `META-INF` folder. If this does not exist, create it. A sample configuration is available in the JBoss Enterprise Application Platform Development Guide. If instead you have `jboss.xml`, you must update your deployment to use the new format. This update process is outside the scope of this guide.
- b. Insert or modify the `<s:security>` element to reference the security domain configured. Change the name in **bold** to that of the configured domain.

```
<jboss:ejb-jar>
...
  <assembly-descriptor>
    <s:security>
      <ejb-name>*</ejb-name>
      <s:security-domain>ISAMWebSSO</s:security-domain>
    </s:security>
  </assembly-descriptor>
...
</jboss:ejb-jar>
```

- c. Save and close the file.

Proceed to *IBM Security Access Manager Configuration* below.

## IBM Security Access Manager Configuration

When J2EE applications are deployed in an environment where container-level authorization is managed by IBM Security Access Manager, objects are created in IBM Security Access Manager object space to represent those objects. The objects created include Resources, representing the Web or EJB permissions for application and Roles, defined within the deployment descriptors of the resources. Only the Role objects require IBM Security Access Manager authorization policy to be applied as the Resources objects are automatically created during container start (deploy) and removed during container shutdown (undeploy).

The location and format of the object space in IBM Security Access Manager is:

```
/WebAppServer/deployedResources/Roles/role-name
```

Role membership is determined from the evaluation of an IBM Security Access Manager authorization decision on the object representing the J2EE role. ACLs must be created and attached to the Role object in the object space and the `[WebAppServer]` permission is checked against the ACL. Permissions for ACLs can be assigned to either an individual user or alternatively to a group, in which all members of the group will be granted the corresponding J2EE role. Refer to the IBM Security Access Manager documentation for ACL configuration syntax.

The JBoss Enterprise Application Platform must be started at least once with the IBM Security Access Manager JACC configuration applied, such that the Resources and Roles objects are created in the object space. During shutdown (undeploy) of the container, all existing Roles and associated authorization policies are maintained.

The following Web Application example demonstrates the required user, group and ACL configuration required to provide delegated authorization. You must perform the required IBM Security Access Manager configuration for each deployed application; however, you need to only configure the authorization policy once for Roles that are common between deployed J2EE applications.

## Web Application Example

Assume a web application requires a single role called `SecAdmins` to access `SampleApplication.jsp` defined in its `web.xml`.

```
<web-app>
...
  <security-constraint>
    <web-resource-collection>
      <web-resource-name>AuthColl1</web-resource-name>
      <description/>
      <url-pattern>/SampleApplication.jsp</url-pattern>
    </web-resource-collection>
    <auth-constraint>
      <description />
      <role-name>SecAdmins</role-name>
    </auth-constraint>
  </security-constraint>
  <security-role>
    <role-name>SecAdmins</role-name>
  </security-role>
</web-app>
```

1. Start the JBoss Enterprise Application Platform server. Ensure the server starts (deploys) successfully.
2. Create a user account who will log in to the target application.

```
pdadmin sec_master> user create secadmin cn=secadmin,dc=iswga
secadmin secadmin password
```

3. Create a group to represent membership of this J2EE role.

```
pdadmin sec_master> group create SecAdmins cn=secadmins,dc=iswga
SecAdmins
```

4. Create an ACL corresponding to this J2EE role.

```
pdadmin sec_master> acl create SecAdmins_role_acl
```

5. Add an ACL entry for the group created above. This command cannot be executed until after an instance of JBoss Enterprise Application Platform has been started, which will create the `WebAppServer` action group if it does not already exist.

```
pdadmin sec_master> acl modify SecAdmins_role_acl set group SecAdmins
T[WebAppServer]i
```

6. If appropriate, update the ACL to contain an entry for the `sec_master` user.



```
pdadmin sec_master> acl modify SecAdmins_role_acl set user sec_master  
T[WebAppServer]i
```

7. Update the ACL to contain an entry for the iv-admin group with the permissions TcmdbsvaBRl to allow it to be managed.

```
pdadmin sec_master> acl modify SecAdmins_role_acl set group iv-admin  
TcmdbsvaBRl
```

8. Attach the ACL to the J2EE Role definition in the IBM Security Access Manager object space.

```
pdadmin sec_master> acl attach  
/WebAppServer/deployedResources/Roles/SecAdmins SecAdmins_role_acl
```

9. Add the required users to the group specified in the ACL to grant them the corresponding J2EE role.

```
pdadmin sec_master> group modify SecAdmins add secadmin
```

10. Repeat steps 4 to 9 for each group required in the Web Application.  
For the sample `SecTest.ear` application the additional ACL creation, group entitlement and role mapping is required for:

- SecUsers
- Getters
- Posters

Users created in *Mapping IBM Security Access Manager Groups to Roles* on page 32 of *Configuring JBoss Enterprise Application Platform for Single Sign On* will already be assigned to the corresponding groups.

11. The application can now be accessed in JBoss Enterprise Application Platform with authorization decisions delegated to IBM Security Access Manager.

## Validating the Integration

To confirm that your IBM Security Access Manager integration with JBoss Enterprise Application Platform has been successfully configured, attempt to access the JBoss Enterprise Application Platform application through the IBM Security Access Manager instance.

```
http[s]://<webseal_server_fqdn>:<webseal_server_port>/jbossso/<Application  
Path>
```

For example:

```
https://webseal.ibm.com:443/jbossso/SecTestWeb
```

Ensure that the correct access is granted to all parts of the application based on the following factors:

- The authenticated user.
- The IBM Security Access Manager group to role mapping.
- Any additional ACL or POP assignments.



## Removing the Integration

To remove the integration between IBM Security Access Manager and JBoss Enterprise Application Platform, first you must restore all modified configuration files to their previous state.

You must also remove the IBM Security Access Manager Login Module before you proceed with the configuration removal of the Java components.

## Removing Integration for Standalone Mode Configuration

### Removing Configuration and Files

1. If deployed, remove the SecTest.ear sample application.
2. Remove redirects to `pkmslogout` in any configured applications.
3. Remove the `ibm` directory and its contents from `<JBOSS_HOME>\modules\com`.
4. Remove any additional `<security domain>` entries added to the JBoss Enterprise Application Platform server configuration file.
5. If the default security domain, `other`, was modified, return it back to its original state.
6. If JACC Authorization was configured,
  - a. Remove `amwas.pdjlog.properties` and `amwas.properties` from `<JBOSS_HOME>\bin`
  - b. Remove the `com.ibm.security.websso` dependency from `<JBOSS_HOME>\modules\system\layers\base\org\jboss\as\security\main\module.xml`
  - c. Remove the `JAVA_OPTS` line added in `standalone.conf[.bat]`.

## Removing Integration for Full Mode Configuration

1. Complete all steps as described in *Removing Integration for Standalone Mode Configuration* above.

## Removing the IBM Security Access Manager Configuration

Use the **pdadmin** tool to complete the following steps

1. Remove the user account that was created to validate the trusted connection between the IBM Security Access Manager web security component and the JBoss Enterprise Application Platform server.  

```
pdadmin sec_master> user delete -registry jboss-sso-user
```
2. Restore the default value of the **basicauth-dummy-passwd** parameter in the **[junction]** stanza of the IBM Security Access Manager web security component configuration file.
3. Restart the WebSEAL instance after this change.

## Removing the IBM Security Access Manager Environment

1. Unconfigure the authorization client for the Login Module:

```
"<JAVA_HOME>\jre\bin\java" -Dpd.cfg.home="<JAVA_HOME>\jre" -classpath  
"<ISAM_HOME>\java\export\pdjrte\PD.jar" com.tivoli.pd.jcfg.SvrSslCfg  
-action unconfig -admin_id sec_master -admin_pwd <password> -  
appsvr_id jboss-sso -policysvr <POLICY_SERVER_HOSTNAME>:7135:1 -host
```

```
<JBoss_EAP_HOSTNAME> -cfg_file
"<JBoss_HOME>\standalone\configuration\jboss-sso.properties"
```

2. If JACC was configured, unconfigure the authorization client for the Authorization Module:

```
"<JAVA_HOME>\jre\bin\java" -Dpd.cfg.home="<JAVA_HOME>\jre" -classpath
"<ISAM_HOME>\java\export\pdjrte\PD.jar" com.tivoli.pd.jcfg.SvrSslCfg
-action unconfig -admin_id sec_master -admin_pwd <password> -
appsvr_id jboss-jacc -policysvr <POLICY_SERVER_HOSTNAME>:7135:1 -host
<JBoss_EAP_HOSTNAME> -cfg_file
"<JBoss_HOME>\standalone\configuration\jboss-jacc.properties"
```

3. Unconfigure the Java Runtime Environment that is used by JBoss Enterprise Application Platform.

```
"<JAVA_HOME>\jre\bin\java" -Dpd.home="<ISAM_HOME>" -classpath
"<ISAM_HOME>\java\export\pdjrte\PD.jar" com.tivoli.pd.jcfg.PDJrteCfg
-action unconfig -java_home "<JAVA_HOME>\jre"
```

4. Restore the PD.jar from the initial installation of the IBM Security Access Manager Runtime for Java.
5. Uninstall the IBM Security Access Manager Runtime for Java.
6. Uninstall the IBM Java Runtime Environment.

## Removing the IBM Security Access Manager Configuration

Use the **pdadmin** tool to complete the following steps:

1. Remove the WebSEAL junction.
- ```
pdadmin sec_master> server task <default-webseald-fqdn> delete
/jbosssso
```
2. Remove all other users and groups created.
  3. Remove any ACLs created during authorization configuration.
  4. Restore the `basicauth-dummy-passwd` stanza entry.
  5. Restore the required authentication mechanism.
  6. Restart the WebSEAL instance.

## TAM Login Module Configuration Options

| Module Option  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Required                                            |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| pdconfig       | The full (Java representation) URL file path to the properties file created by the <code>com.tivoli.pd.jcfg.SvrSslCfg</code> Java configuration tool.<br><br>The file contains the necessary configuration to allow the authorization client to contact the IBM Security Access Manager servers.                                                                                                                                                                                                                                                    | Full Mode configuration.                            |
| idType         | The HTTP request header that contains the user authenticated by IBM Security Access Manager web security component.<br>Supported values are <code>iv-creds</code> and <code>iv-user</code> .<br><br>Use of the <code>iv-creds</code> HTTP request header is preferable.                                                                                                                                                                                                                                                                             | Full Mode configuration.                            |
| loginID        | The user ID of the account that you created to validate the trusted connection between IBM Security Access Manager web security component and JBoss Enterprise Application Platform server.                                                                                                                                                                                                                                                                                                                                                         | Full Mode configuration.                            |
| buildRolesFrom | The IBM Security Access Manager Login Module object that assigns roles for the authenticated user.<br>For single sign-on, set this value to <code>PDPrincipal</code> .<br><br>If <code>PDPrincipal</code> is not specified, users are authenticated to JBoss Enterprise Application Server container, but no roles are assigned.                                                                                                                                                                                                                    | Optional<br>Ignored in Standalone mode.             |
| reqHdrList     | A comma-delimited list of additional HTTP request headers that must be present in the request to validate the trusted connection in addition to the <code>loginID</code> .<br>For example, <code>iv-server-name</code> , or <code>iv-user</code> might be added if <code>-c iv-creds,iv-user</code> was specified during WebSEAL junction creation. The values of the HTTP request headers are not validated, only the existence of the header name. The <code>idType</code> header is added to this list at run time if it is not already present. | Optional                                            |
| checkViaHeader | Enforce trusted host validation for the incoming request. Valid values are <code>true</code> and <code>false</code> . The default value is <code>false</code> .<br><br>If set to <code>true</code> , only validation requests from a host contained in the <code>hostnames</code> property on a port contained in the <code>ports</code> property are serviced – all other requests are ignored.                                                                                                                                                    | Optional                                            |
| hostnames      | A comma-delimited list of hostnames from which connections will be acted upon. The example value of <code>webseal1.ibm.com,webseal2.ibm.com</code> would only service authorization requests from these two hosts on <code>ports</code> specified in the <code>ports</code> property.                                                                                                                                                                                                                                                               | Only required if <code>checkViaHeader</code> is set |
| ports          | A comma-delimited list of ports from which connections will be acted upon. The example value of <code>80,443</code> would only service authorization requests from these two ports on hosts specified in the <code>hostnames</code> property.                                                                                                                                                                                                                                                                                                       | Only required if <code>checkViaHeader</code> is set |
| ignoreProxy    | This property is used to determine whether or not hosts advertised as proxies in the Via header are checked for trusted host validation. If set to <code>false</code> , proxies are validated for hostname/port trust. If set to <code>true</code> , these entries in the Via header are ignored.                                                                                                                                                                                                                                                   | Only required if <code>checkViaHeader</code> is set |

| Module Option | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Required                                   |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
|               | Valid values are <code>true</code> and <code>false</code> . The default value is <code>false</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                            |
| viaDepth      | <p>A positive integer that specifies the number of source hosts in the Via header to check for trust. By default, every host in the Via header is checked, and if any host is not trusted, trust cannot be established. The <code>viaDepth</code> property is used when only some of the hosts in the Via header have to be trusted. The setting indicates the number of hosts that are required to be trusted.</p> <p>The <code>viaDepth</code> property is set to 0 by default, which means all of the hosts in the Via header are checked for trust.</p>                                                                                                                                                                                                                                                                                                | Optional                                   |
| ssoPwdExpiry  | <p>After trust is established for a request, the single sign-on user password is cached, eliminating the need to re-authenticate the single sign-on user with IBM Security Access Manager for every request. You can modify the cache timeout period by setting the single sign-on password expiry property to the required time in seconds.</p> <ul style="list-style-type: none"><li>• By default, single sign-on user trust is re-established every 600 seconds.</li><li>• When set to 0, an initial single sign-on user trust is established and the cached password never expires.</li><li>• When set to -1, no single sign-on user trust is established. Use this value for Standalone Mode Configuration as described in <i>Installing and configuring the IBM Security Access Manager Login Module in Standalone Mode Configuration</i>.</li></ul> | Optional.<br>Required for Standalone mode. |

## Troubleshooting

---

If you experience any problems during integration, examine the following notes for help in identifying problems. This chapter describes common configuration problems that are encountered during deployment and outlines possible resolutions.

This chapter also provide details on obtaining trace log files for the Login Module.

### Symptom 1

When you are starting the JBoss Enterprise Application Platform server instance, the following error is produces

```
ERROR [org.jboss.msc.service.fail] (MSC service thread 1-2)
MSC000001: Failed to start service
jboss.module.service."deployment.SecTest.ear".main:
org.jboss.msc.service.StartException in service
jboss.module.service."deployment.SecTest.ear ".main: JBAS018759:
Failed to load module: deployment.SecTest.ear :main
...
Caused by: org.jboss.modules.ModuleNotFoundException: Module
com.ibm.security.websso:main is not found in local module loader
@5cac6a45 (roots: /usr/local/EAP-<version>/jboss-eap-
<version>/modules)
```

### Solution

Ensure that the `com` directory supplied with this integration is copied in the

`<JBOSS_HOME>\modules` directory and

`<JBOSS_HOME>\modules\com\ibm\security\websso\main` contains the following files:

- `module.xml`
- `rbpf.jar`
- `AMJACCPProvider.jar`
- `TAMJBoss3.1.jar`

### Symptom 2

When you attempt to start the JBoss Enterprise Application Platform server instance, the following error is produced:

```
ERROR [org.jboss.as.controller.management-operation] JBAS014613:
Operation ("add") failed - address: ([
  ("subsystem" => "security"),
  ("security-domain" => "other")
]) - failure description: "JBAS014803: Duplicate resource [
  (\\"subsystem\\" => \\"security\\"),
  (\\"security-domain\\" => \\"other\\")
] "
```

### Solution

Ensure that there are only one security domain names "**other**" in the configuration file for the JBoss Enterprise Application Platform server. All security domain names in this file must be unique. If you want to modify the default security behaviour, you must replace the entry instead of creating one with the same name.

### Symptom 3

When you are starting the JBoss Enterprise Application Platform server instance, a message similar to the following is displayed.

```
JBAS014775:      New      missing/unsatisfied      dependencies:      service
jboss.security.security-domain.SecTest (missing) dependents: [service
jboss.web.deployment.defaulthost./SecTest.realm]
```

#### Solution

Ensure the `<security-domain>` specified in the `jboss-web.xml` for your application matches the `<security-domain>` specified in `standalone.xml`.

### Symptom 4

The following error is displayed when you attempt to use the custom Login Module.

```
HPDAC1373E aznAPI -- User registry authenticate failed.
...
Login failure: javax.security.auth.login.LoginException: Exception
occurred extracting PDPrincipal from Servlet request.
java.lang.Exception: Basic Authentication failed.
```

#### Solution

Make sure that the `basicauth-dummy-passwd` parameter in the `[junction]` stanza of the IBM Security Access Manager WebSEAL configuration file matches the password of the IBM Security Access Manager user that is used to establish a trusted connection. You must restart the WebSEAL instance after this change.

For more information, see [Configuring Trust Validation](#) on page 16.

### Symptom 5

The following error is displayed when you attempt to use the custom Login Module.

```
HPDAC1354E aznAPI -- User's password has expired.
...
Login failure: javax.security.auth.login.LoginException: Exception
occurred extracting PDPrincipal from Servlet request.
java.lang.Exception: Basic Authentication failed.
```

#### Solution

Make sure that the `password-valid` attribute of the JBoss Enterprise Application Platform user is set to `yes`.

For more information, see [Configuring Trust Validation](#) on page 16.

### Symptom 6

The following error is displayed when you attempt to use the custom Login Module

```
HPDAC1364E aznAPI -- Account Login Disabled.
...
Login failure: javax.security.auth.login.LoginException: Exception
```

```
occurred extracting PDPrincipal from Servlet request.  
java.lang.Exception: Basic Authentication failed.
```

### Solution

Make sure that the **account-valid** attribute of the JBoss Enterprise Application Platform user is set to yes.

For more information, see [Configuring Trust Validation](#) on page 16.

### Symptom 7

The following error is displayed in the browser when you attempt to use the custom Login Module.

```
Unexpected Authentication Challenge  
  
Access Manager WebSEAL received an unexpected authentication  
challenge from a junction Web server.
```

### Solution

Ensure that the WebSEAL instance is set to supply the HTTP Basic Authentication Header.

For more information, see [Configuring Trust Validation](#) on page 16.

### Symptom 8

The following error is displayed when you attempt to use the custom Login Module.

```
HPDIA0202W An unknown user name was presented to Access Manager.  
...  
Login Failure: javax.security.auth.login.LoginException: Exception  
occurred extracting PDPrincipal from Servlet request  
java.lang.Exception: Basic Authentication failed.
```

### Solution

Make sure the value of the **loginID** module option in `standalone.xml` exactly matches the user name of the IBM Security Access Manager user that is used to establish a trusted connection. For more information about configuring `standalone.xml` file, see *Installing and configuring the IBM Security Access Manager Login Module in Full Mode Configuration* on page 14

### Symptom 9

The following error is displayed when you attempt to use the custom Login Module

```
Wrappered Exception:  
  
javax.net.ssl.SSLHandshakeException: Error signing certificate verify  
[ HPDJA0116E Cannot contact server. ]  
  
...  
Caused by: java.security.NoSuchAlgorithmException: SHA224withRSA  
Signature not available.
```

### Solution

This error occurs when you attempt to use the Login Module with Oracle Java 7/8 Runtime. Ensure that the Java Runtime has been configured to use the IBM JCE Provider. For more information, see *IBM Security Access Manager 8 and 9 Runtime for Java prerequisites for Oracle* on page 12.

### Symptom 10

The following error is displayed when you attempt to use the custom Login Module

```
java.lang.IllegalStateException: java.lang.NoClassDefFoundError:  
com.ibm.misc.Debug
```

```
...  
Caused by: java.lang.ClassNotFoundException: com.ibm.misc.Debug
```

### Solution

This error occurs when you attempt to use the Login Module with Oracle Java 7/8 Runtime, but `ibmpkcs.jar` is in the wrong location. Ensure that `ibmpkcs.jar` is placed under `<JAVA_HOME>\lib\ext`. For more information, see *IBM Security Access Manager 8 and 9 Runtime for Java prerequisites for Oracle* on page 12.

### Symptom 11

The following error is displayed when you attempt to use the custom Login Module

```
TAMLoginModule.getPdContext().  
Error encountered while creating PDAuthorizationContext:  
file:///usr/local/EAP-<version>/standalone/configuration/jboss-  
sso.properties  
...  
Login failure: javax.security.auth.login.LoginException: Exception  
occurred extracting PDPrincipal from Servlet request.  
java.lang.Exception: The Authorization context could not be created,  
check the location of SSO properties file for the urlConfig  
java.lang.NullPointerException
```

### Solution

Ensure that the following conditions are met:

- The `jboss-sso.properties` file exists.
- The value of the **pdconfig** module option in `standalone.xml` correctly specifies the location of the `jboss-sso.properties` file.
- **SvrSslCfg** has been used to configure the runtime correctly. If `jboss-sso.properties` exists but contains less than 50 lines of configuration properties, the runtime has been unconfigured with **SvrSslCfg** and must be reconfigured.

For more information, see *Configuring Application Server Wide Single Sign-On* on page 28 and *IBM Security Access Manager Single Sign-On to JBoss Enterprise Application Platform* on page 13.

### Symptom 12

When you are using the `com.tivoli.pd.jcfg.SvrSslCfg` Java command to create an authorization client, the following error is displayed.



```
HPDJJA0803E Database URL does not specify a directory.
```

**Solution**

Specify the **-dbdir** property when you are using the `com.tivoli.pd.jcfg.SvrSslCfg` Java command. ...-dbdir "<JBoss\_HOME>\standalone\configuration"

**Symptom 13**

When you are configuring the IBM Security Access Manager for Java, one of the following errors is displayed.

```
Exception in thread "main" java.lang.SecurityException: class
"com.tivoli.pd.jutil.c"'s signer information does not match signer
information of other classes in the same package
```

or

```
java.lang.NoClassDefFoundError: com/ibm/security/x509/X509CertImpl
```

**Solution**

Ensure that there are no old versions of the IBM `PD.jar` file in the classpath. Old versions include backed up versions such as `PD.jar.old`. If necessary, remove all files except for the correct `PD.jar` from `<ISAM_HOME>\java\export\pdjrte`.

**Symptom 14**

When you are configuring IBM Security Access Manager Runtime for Java, the following error is displayed.

```
Exception in thread "main" java.lang.NoClassDefFoundError:
com/tivoli/pd/jcfg/PDCheckJre
```

```
Exception in thread "main" java.lang.NoClassDefFoundError:
com/tivoli/pd/jcfg/PDJrteCfg
```

**Solution**

Ensure that `PD.jar` file that is provided in the integration package is at `<ISAM_HOME>\java\export\pdjrte\PD.jar`. This file must have the same file permissions as the original `PD.jar` file.

**Symptom 15**

When you are configuring the IBM Security Access Manager Runtime for Java, the following error is displayed.

```
The java class could not be loaded.
```

```
java.lang.UnsupportedClassVersionError: com/tivoli/pd/jcfg/PDJrteCfg
(Unsupported major.minor version X.Y)
```

**Solution**

This error occurs if the `PD.jar` file shipped with this integration package for an Oracle Java Runtime Environment is used in an IBM Java Runtime Environment. Ensure that your JBoss Enterprise Application Platform server is running on an Oracle JRE. The `JAVA_HOME` environment variable,

Java executable in the `PATH` and `-java_home` argument to the `com.tivoli.pd.jcfg.PDJrteCfg` Java command must all point to the same Java Runtime Environment.

## Symptom 16

When you are configuring the IBM Security Access Manager for Java, the following error is displayed.

```
Exception in thread "main" java.lang.NoClassDefFoundError:
com/tivoli/pd/jutil/AMFipsMode at
com.tivoli.pd.jcfg.PDJrteCfg.config(PDJrteCfg.java:1543)
```

### Solution

This error occurs if the `PD.jar` installed by the IBM Security Access Manager Runtime for Java is used in an Oracle Java Runtime Environment. Ensure that the `JAVA_HOME` environment variable, Java executable in the `PATH`, the `-java_home` argument to `com.tivoli.pd.jcfg.PDJrteCfg` Java command and the `PD.jar` in `<ISAM_HOME>\java\export\pdjrte` all point to the same Oracle Java Runtime Environment.

## Symptom 17

When you are configuring the IBM Security Access Manager Runtime for Java, the following error is displayed.

```
java.lang.reflect.InvocationTargetException
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.
java:39)
at sun.reflect.DelegatingMethodAccessorImpl.invoke
(DelegatingMethodAccessorImpl.java:25)
at java.lang.reflect.Method.invoke(Method.java:585)
at com.tivoli.pd.jcfg.PDJrteCfg.config(PDJrteCfg.java:1570)
```

### Solution

Ensure that the `-java_home` argument of the `com.tivoli.pd.jcfg.PDJrteCfg` Java command specifies the location of the Java Runtime Environment, not the Java Development Kit.

## Symptom 18

When you are configuring IBM Security Access Manager Runtime for Java for following error is displayed.

```
Access Manager License Fileset not correct
```

### Solution

Ensure the correct `PD.jar` file for the installed IBM Security Access Manager Runtime for Java is copied into `<ISAM_HOME>\java\export\pdjrte`. Also, ensure the `pd.home` parameter is correct. This parameter is specified during [IBM Security Access Manager Runtime for Java Installation and Configuration](#) on page 12.

If the problem persists and IBM Security Access Manager for Java version 8 is installed, backup the `PDlic.txt` in `<ISAM_HOME>\.configure\` and replace it with the one contained in `lib\ISAM8\`. Re-execute the IBM Security Access Manager Runtime for Java configuration steps. To remove this integration or upgrade the IBM Security Access Manager Runtime for Java version,

## Integration Guide

after unconfiguring the IBM Security Access Manager Runtime for Java, replace the `PDlic.txt` file with the backed up version.

### Symptom 19

When you are configuring the IBM Security Access Manager Runtime for Java, following error is displayed.

```
java.lang.reflect.InvocationTargetException
HPDBA0226I   The requested command is not supported by the server.
```

#### Solution

Ensure that the `-host` and `-port` arguments for the `com.tivoli.pd.jcfg.PDJrteCfg` Java command refer to the Policy server and not the Authorization server.

### Symptom 20

When you are using the `SvrSslCfg` tool, the following error is displayed:

```
HPDBF0234E
Unable to load pd.properties.
```

#### Solution

Ensure that the JRE has been configured to use the IBM Security Access Manager Runtime for Java. For more information, see *Installing and configuring the IBM Security Access Manager Login Module in Full Mode Configuration* on page 14.

### Symptom 21

When configuring Context – Based Access (CBA), also known as Risk – Based Access (RBA) and testing the integration, you receive the following message from WebSEAL:

```
Forbidden
The resource you have requested is secured by Access Manager WebSEAL.
```

#### Solution

Ensure that IBM Security Access Manager for Mobile has been configured for the current runtime. See *Mobile Authorization Decision Point Configuration* on page 24 for more information.

### Symptom 22

During startup (deploy) or shutdown (undeploy) of JBoss Enterprise Application Platform the following error is displayed on the console or in `<JBOSS_HOME>\standalone\log\server.log` file.

```
AWXRB0075E   An unexpected Tivoli Access Manager exception was caught
while attempting to create a root object space, /WebAppServer, with
the description This object was created by RBPF. The details are: 0x0
[ HPDAC1050E   Operation is not authorized.]
```

#### Solution

*Integration Guide*

Ensure `iv-admin` is added to the authorization client user account created by the `com.tivoli.pd.jcfg.SvrSslCfg` Java command for JACC Policy Provider.

```
pdadmin sec_master> group modify iv-admin add jboss-  
jacc/<JBOSS_EAP_HOSTNAME>
```

### Symptom 23

When creating an authorization client using the `com.tivoli.pd.jcfg.SvrSslCfg` Java command, the following error message is displayed:

```
HPDJA0803E Database URL does not specify a directory.
```

#### Solution

Specify the `-dbdir` property when executing `com.tivoli.pd.jcfg.SvrSslCfg` Java command.

```
... -dbdir <JBOSS_HOME>\standalone\configuration
```

### Symptom 24

The JBoss Enterprise Application Platform does not start correctly and displays:

```
Instantiated: name=JaccPolicyProvider state=Described  
  
com.tivoli.pd.as.jacc.util.JACCException: AWXJR0006E The file,  
<JBOSS_HOME>\bin\amwas.properties, was not found.  
at  
com.ibm.tivoli.integration.am.jboss.TAMJACCJBossPolicy.init(Unknown  
Source)  
at  
com.ibm.tivoli.integration.am.jboss.TAMJACCJBossPolicy.<init>(Unknown  
Source)
```

#### Solution

Ensure the configuration file `amwas.properties` and `amwas.pdjlog.properties` are copied to the named directory in the error message (`<JBOSS_HOME>\bin`) and `amwas.properties` has been configured with the URL file paths for the entries

`com.tivoli.pd.as.rbpf.AmasSession.LoggingURL` and  
`com.tivoli.pd.as.rbpf.AmasSession.CfgURL`.

### Symptom 25

The JBoss Enterprise Application Platform does not start correctly and displays

```
Failed to start service jboss.security.bootstrap:  
org.jboss.msc.service.StartException in service  
jboss.security.bootstrap: JBAS013308: Unable to start  
theSecurityBootstrapService service  
  
Caused by: java.lang.IllegalArgumentException
```

#### Solution

This error indicates a parsing error inside the JACC configuration file. Ensure the configuration file `amwas.properties` has been configured with valid URLs for the following properties:

```
com.tivoli.pd.as.rbpf.AmasSession.LoggingURL
com.tivoli.pd.as.rbpf.AmasSession.CfgURL.
```

 and

## Symptom 26

The JBoss Enterprise Application Platform does not start correctly and displays

```
Failed to start service
jboss.deployment.unit."<APPLICATION>".jboss.security.jacc:
org.jboss.msc.service.StartException in service
jboss.deployment.unit."<APPLICATION>".jboss.security.jacc: Failed to
start service

Caused by: java.lang.NoClassDefFoundError:
com.ibm.websphere.security.auth.CredentialDestroyedException
```

### Solution

This error is indicative of a prior error encountered during the initialization of the JACC Provider. Find the previous error in the server log file and follow the troubleshooting steps for that error.

## Symptom 27

When deploying sample application, JBoss Enterprise Application Platform does not start correctly and displays

```
JBAS014775:      New missing/unsatisfied dependencies:

service jboss.security.security-domain.SecTest (missing) dependents:

[service jboss.web.deployment.default-host./SecTestWeb.realm, service
jboss.deployment.subunit."SecTest.ear"."SecTestEJB.jar".component.Sec
Test.CREATE]
```

### Solution

Ensure that the name of the security-domain in `<JBOSS_HOME>\standalone\configuration\standalone.xml` matches the configured security domain for the deployed application. For the sample application, it will be `SecTest`.

## Symptom 28

The JBoss Enterprise Application Platform does not start correctly and displays

```
failed handling operation rollback --
java.util.concurrent.TimeoutException
```

Additionally, `<ISAM_HOME>\log\msg__<JBOSS_EAP_HOSTNAME>.log` contains the error message:

```
HPDBA0222E    The TCP/IP host information could not be determined from
the server hostname.  Ensure that the server hostname is correct.
```

### Solution

This error occurs when the IBM Security Access Manager instance cannot resolve the hostname of the JBoss Enterprise Application Platform server. Ensure that the `-host` flag in `SvrSslCfg` is entered correctly and that this hostname resolves to the IP address of the application server. If required, add the host to the hosts file of the IBM Security Access Manager instance.

## Collecting Support Data

This section describes the specific data needed for problem determination for JBoss Enterprise Application Platform integration issues.

### Collecting Data for IBM Security Access Manager for JBoss Enterprise Application Platform

This section describes the specific “Must Gather” data needed for problem determination for JBoss Enterprise Application Platform integration issues. The following files should be gathered along with the relevant IBM Security Access Manager data (see [Collecting Data for ISAM: Read first for all IBM Security Access Manager products](#)) and submitted to support to aid in problem determination.

- The JBoss server log file. In a standalone installation, it will be located at `<JBOSS_HOME>\standalone\log\server.log`.
- The JBoss server configuration file. In a standalone installation, it will be located at `<JBOSS_HOME>\standalone\configuration\standalone.xml`.
- The application configuration files, located at `<APPLICATION_DEPLOYMENT>/WEB_INF/`. Both `web.xml` and `jboss-web.xml` files are required.
- The Tivoli Policy Director error log file. This will be located at `<ISAM_HOME>\log\msg__amj_error1.log`.

### Enable IBM Security Access Manager Login Module Trace

To enable trace for the Login Module in JBoss Enterprise Application Platform

1. Login to JBoss Enterprise Application Platform Administration Console.
2. Navigate to the logging subsystem configuration. If your JBoss Enterprise Application Platform server is running in a standalone configuration:
  - a. Navigate to the Subsystem pane by clicking **Configuration** in the upper left hand corner.
  - b. Expand **Core** in the subsystems pane on the left side.
  - c. Click **Logging**.
3. Select the **Log Categories** tab and click **Add**.
4. Specify the log category name as `com.ibm.security.websso.TAMLoginModule` and select a log level of at least **FINE**.
5. If you are using the IBM Security Access Manager JACC Provider, add `com.ibm.security.websso.TAMJACCJBossPolicy` and select a log level of at least **FINE**.
6. Click **Save**.

The trace is written to the server log file at `<JBOSS_HOME>\standalone\log\server.log` for standalone server instance.

For more information about logging in JBoss Enterprise Application Platform search for the Logging Subsystem details in JBoss Enterprise Application Platform 6: Administration and Configuration Guide.

## JACC Provider Caching

The JACC module that is part of this integration implements caches, to improve the performance if authorization decisions. Because of these caches, unexpected results may be obtained after changing a user's group membership or security policy (ACLs etc.), within the lifetime of the cache entries. It is advisable to reduce cache lifetimes during development and unit testing and increase them again during system testing, performance testing and production environments.

### Role Principal Cache

This cache stores principal objects, as well as the relationship between users and the roles they possess.

The default lifetime of entries in the principal cache is 10 minutes and is controlled by the `com.tivoli.pd.as.cache.DynamicRoleCache.PrincipalLifeTime` parameter in the `amwas.properties` file.

The default lifetime of entries in the principal-role cache is 20 minutes, and is controlled by the `com.tivoli.pd.as.cache.DynamicRoleCache.RoleLifetime` parameter in `amwas.properties` file.

### Role – Resource Cache

This cache stores the relationship between roles and the resources they have access to. The default lifetime entries in this cache is 20 minutes, and is controlled by the `com.tivoli.pd.as.cache.ObjectCache.ResourceLifetime` parameter in `amwas.properties` file.

## Notices

---

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785 U.S.A.*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan*

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Corporation  
224A/101  
11400 Burnet Road*



*Austin, TX 78758 U.S.A.*

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© IBM 2015. Portions of this code are derived from IBM Corp. Sample Programs. ©Copyright IBM Corp 2010, 2015. All rights reserved.

If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

## Trademarks

---

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and trademark information; at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.



Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Other company, product, and service names may be trademarks or service marks of others.