



SAML 1.1 Java Toolkit 2.0

User's Guide

© 2004 Ping Identity Corporation. All rights reserved

Part Number 3001-022
Version 1.00
10/22/2004

Ping Identity Corporation
1400 16th St. Suite 220
Denver, CO 80202
U.S.A.
Phone: 303.468.2900
<http://www.pingidentity.com>

Contents

	About This Guide	1
	SourceID Overview	1
	Intended Audience	1
	Support	1
	Text Conventions	2
	Introduction	3
	System Requirements	3
Chapter 1	Installation	5
	Installing the Toolkit and Demo Application	5
	Viewing API Documentation	6
Chapter 2	Using the Demo Application	7
	Introduction	7
	Performing Single Sign-On	7
	Using SAML Protocol Queries	8
Chapter 3	Toolkit Architecture	9
Chapter 4	Integrating the Toolkit	11
	Building and Packaging the Toolkit	11

About This Guide

SourceID Overview

SourceID is an open source project for enabling identity federation and cross-boundary security. SourceID focuses on ease of integration and deployment within existing Web applications, products, or services. In addition, SourceID provides a high level of developer functionality and customization and is designed to shield the integrator and enterprise from needing to understand the complexities of federation, or the rapidly evolving federation standards.

Intended Audience

This guide is written for individuals who have basic knowledge of Ant, JBoss, Java, and the Security Assertion Markup Language (SAML) specifications. For information about SAML, see the OASIS [Technical Overview](#) of version 1.1 available on the organization's Web site:

www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

Support

User and developer support documentation and mailing lists are located on the SourceID Web site at www.sourceid.org.

Text Conventions

This document uses some or all of the typographical conventions identified below.

Table 1: Text Convention Definitions

Convention	Description
Bold	Indicates first use of product specific terminology defined in the glossary and identifies the beginning of procedures and examples.
Fixed Width	Indicates text you must type exactly as shown in the instructions. Also used to represent program code, file names, and directory paths.
Fixed Width Bold	Used to identify event handlers, function calls, and API objects.
Blue text	Used in online documents to indicate hyperlinks.
<i>Italic</i>	Used for emphasis and to identify document titles.
Teal text	Used for parameters and attributes in online documents.
Sans serif	Identifies GUI text as shown on a screen. Example: "Print Document dialog"
Sans serif bold	Identifies menu items or buttons. Example: Click Start > Programs > Photoshop

Introduction

Two components are included in this distribution: the SAML 1.1 Java Toolkit framework and a demonstration application. The demonstration application shows how to integrate the Toolkit framework into an existing application.

This release of SAML 1.1 Java Toolkit supports the core SAML profiles:

- Single Sign-on (SSO) (Artifact & POST)
- SAML SOAP Protocol Binding

System Requirements

The following are required to install SAML 1.1 Java Toolkit:

- Windows XP or Linux (kernel 2.4.2+)
- Sun Java JDK 1.4.2
- JBoss 3.2.5
- Ant 1.6.2 (or higher)

Installation

Installing the Toolkit and Demo Application

If you have not already done so, go to the SourceID Web site (<http://www.sourceid.org/download.do>) and download the SAML-1.1 Java Toolkit 2.0 zip file.



Note: This procedure assumes that you have JBoss, the JDK, and Ant installed on your system (see “[System Requirements](#)” on page 3).

- 1 Unzip the source archive into a work directory.
- 2 Open the `SAML_1.1_Java_Toolkit_2.0` directory.
- 3 Turn off the read-only property for the file `build.local.properties`.
- 4 Edit `build.local.properties` to set the `jboss.dir` property to the directory where JBoss is installed.
- 5 Copy the edited `build.local.properties` to the Infrastructure directory, replacing the existing file.
- 6 Run `ant demo-deploy` from the `SAML_1.1_Java_Toolkit_2.0` directory.

- 7 Optionally, disable the verbose logging of the embedded workflow engine by adding the following to the `${jboss.server.dir}/conf/log4j.xml` file above the configuration for the root logger:

```
<category name="org.obe">  
  <priority value="WARN"/>  
</category>
```

- 8 Start JBoss.
- 9 At this point, the Demo application should be deployed and ready to use. You can access the Demo application by going to:

```
http://localhost:8080/sourceid-saml-demo/
```

The page has links to Asserting Party (AP) and Relying Party (RP) login pages as well as a link to a page that demonstrates the SAML attribute request SOAP protocol binding. (See [“Using the Demo Application”](#) on page 9.)

Please note that default deployment of the SAML 1.1 Java Toolkit server acts as both AP and RP.

Viewing API Documentation

To generate Javadoc for the SAML 1.1 Java Toolkit:

- At a Command prompt run:

```
ant doc-public-toolkit-apis
```

from the `SAML_1.1_Java_Toolkit_2.0` directory.

To view the API documentation for the Toolkit adapter interfaces, use a Web browser to open:

```
SAML_1.1_Java_Toolkit_2.0/build/doc/index.html
```

For general information about the adapter interfaces, see [“Toolkit Architecture”](#) on page 11.

Using the Demo Application

Introduction

The demonstration application serves as a simple example of using and integrating the toolkit. It contains implementations of all the toolkit adapter interfaces and shows how to configure their usage (see [“Toolkit Architecture”](#) on page 11). It also provides examples of other toolkit functionality, including Simple Object Access Protocol (SOAP) user-attribute queries.

A Demo Application page is provided as a means of viewing the results of Toolkit functionality. After you have installed and deployed the demo (see [“Installation”](#) on page 7), use a Web browser to view the page at:

`http://localhost:8080/sourceid-saml-demo/`

Performing Single Sign-On

- 1 On the SAML Demo Application page, click AP (Asserting Party) site.
- 2 On the AP Web Site, enter `joe` for User name and `test` for Password.
- 3 Click **Login**.
- 4 At the bottom of the page, click either **Post** or **Artifact**.

You have now performed a single sign-on and are logged into the RP (Relying Party) Web Site. From here, you can retrieve all available attributes from the AP by clicking the related link on the RP Web Site.

To retrieve specific attributes by name or by assertion ID, return to the Demo Application page to reach the SAML Protocol Request/Response Demo Page (see the next section).

Using SAML Protocol Queries

To execute SOAP protocol queries:

- ▶ Click **SAML Protocol Request/Response Demo Page** on the demo page and following the procedures below in sequence.

To request all attributes:

- 1 On the SAML Protocol Query Page, enter:
`joe@mail.com` in the Subject Name ID field.
- 2 (Optional) Enter `email` in the Subject Name ID Qualifier field.
- 3 For Subject Name ID format, select:
`urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress`
- 4 Click **Request All Attributes**.

To request named attributes:

- 1 For Attribute Names, enter one or more attributes from the list of all attributes (see the procedure above).

You can use the browser Back and Forward buttons to copy and paste attribute names from the previous Test Results page.

- 2 Click **Request Named Attributes**.
- 3 From the Test Results Valid Assertions XML section, highlight and copy the Assertion ID value for the next procedure.

You will use the ID in the next procedure.

To request attributes by Assertion ID:

- 1 For Assertions IDs, paste the ID copied for the named-attributes test above.

If you want to test multiple Assertion IDs, repeat either of the two procedures above and copy and past additional IDs from the Test Results.

- 2 Click **Request Assertions by ID**.

Toolkit Architecture

From a developer and deployment standpoint, the SAML 1.1 Java Toolkit's adapter tier is a critical aspect of the application architecture. The adapter tier is a set of Java interfaces that allow developers to customize how data is stored and how interactions with the Web application are handled (see [Figure 1](#)). You implement these interfaces to adapt SourceID to your application environment.

From the `SAML_1.1_Java_Toolkit_2.0` directory, the adapter interfaces are located in:

```
src/java/org/sourceid/saml11/adapter/ap
```

and

```
src/java/org/sourceid/saml11/adapter/rp
```

The demo application provides an example of how to implement the adapter interfaces and configure their usage (see [“Using the Demo Application”](#) on page 9 and [“Integrating the Toolkit”](#) on page 13). From the `SAML_1.1_Java_Toolkit_2.0` directory, the implementations in the demo can be found in:

```
demo/src/java/org/sourceid/saml11demo/ap/adapter
```

and

```
demo/src/java/org/sourceid/saml11demo/rp/adapter
```

To select the implementation to use for a given adaptor, see the `sourceid-core-config.xml` file in `src/app/config`.

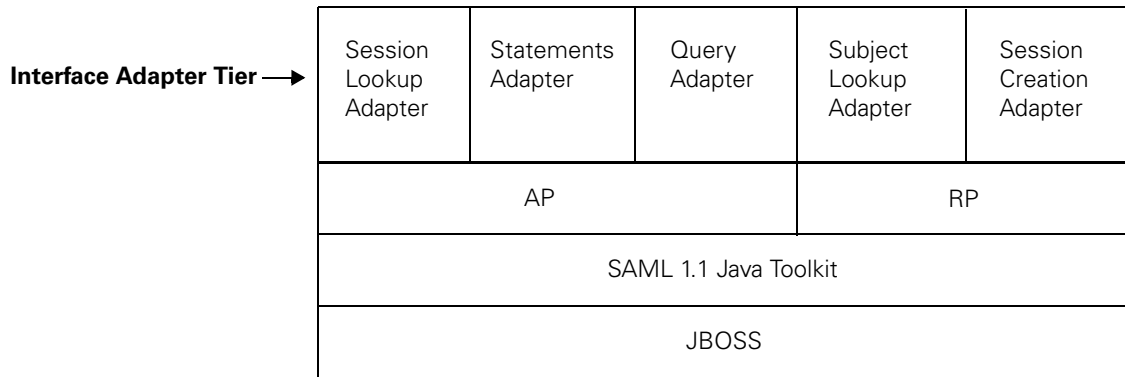


Figure 1 SAML 1.1 Java Toolkit High Level Architecture

Session Lookup Adapter — Provides a lookup mechanism for an end user's session, allowing the system to identify the user based on session information. Typical implementations of this adapter might look for some object or value on the user's session or check cookies. The returned map of identifying information is passed to the Statements Adapter.

Statements Adapter — Builds SAML statements about a user to be passed in an SSO assertion. Typical implementations of this adapter might look up the user into a back-end persistent store and construct the statements from data about the user. At least one authentication statement must be included.

Subject Lookup Adapter — Returns a Java map identifying information about the subject contained in incoming assertions. This adapter is used to identify the user locally, based on information in an assertion. The map, `subjectIdentifiers`, is passed to the Session Creation Adapter.

Session Creation Adapter — Creates a local session for the user identified by the `subjectIdentifiers` map. Typical implementations of this adapter might set a cookie or put an object on the HTTP session.

Query Adapter — Allows you to plug in application-specific logic in order to fulfill SAML query requests—used only for advanced SOAP queries. See the SAML specifications for more information about SAML query processing rules.

Integrating the Toolkit

Building and Packaging the Toolkit

You must build the SAML 1.1 Java Toolkit framework and integrate it into your application to incorporate SAML functionality. The following procedure will guide you through this process.

- 1 From the `SAML_1.1_Java_Toolkit_2.0` directory, run `ant dist` at a command prompt to build the SAML 1.1 Java Toolkit framework. This creates the `sourceid-saml.zip` file in `SAML_1.1_Java_Toolkit_2.0/dist`.
- 2 Extract the `sourceid-saml.zip` file into a working directory `{my_dir}`.
- 3 Copy the contents of `{my_dir}/lib` to the `WEB-INF/lib` directory of your application.



Note: If the `WEB-INF/lib` directory does not exist on the application, place the contents in the runtime class path of the application.

- 4 Merge the contents of `{my_dir}/template/web.xml` with the `web.xml` file for the application.

You may need to edit the files listed below to configure the distribution. They are located in `{my_dir}/template/config`.

```
sourceid-core-config.xml
sourceid-soap-auth.xml
```

Examples of edited versions of these files for the demo application are located in `SAML_1.1_Java_Toolkit_2.0/demo/src/config`.

- 5 Copy the `{my_dir}/template/jboss-web.xml` file to the `/WEB-INF` directory of the application.
- 6 Copy the contents of the `{my_dir}/template/config` directory into the `{jboss_home}/server/default/conf` directory of the application.
- 7 Copy the contents of the `{my_dir}/template/resource` directory to the class path `WEB-INF/classes` of your Web application.

One file may require editing:

`sourceid-log-messages.properties`

An example of an edited version of this file for the demo application is located in `SAML_1.1_Java_Toolkit_2.0/demo/src/resource`.

- 8 The demo Web interface provides some examples of how to invoke SourceID SAML functionality. The files described below are located in:

`SAML_1.1_Java_Toolkit_2.0/demo/src/web/WEB-INF/jsp/`

For Asserting Party implementations, the file `ap/mainpage.jsp` demonstrates links for calling the `InterSiteTransferServlet` to initiate SSO.

For Relying Party implementations, the following examples show how to use the API to make SAML Protocol requests via SOAP:

- The file `rp/mainpage.jsp` has a link to `RequestAttrsServlet`, which uses the API for an attribute query.
 - The JSPs in `rp/protocol/` interact with `Sam1ProtocolRequestServlet` to show other uses of the API for SOAP requests.
-

SourceID[™]

Ping Identity Corporation
1400 16th St. Suite 220
Denver, CO 80202
U.S.A.