



ID-WSF 2.0 Tools & Libraries & Firefox SAMLv2 ECP Extension

Prepared by Asa Hardcastle, Zenn New Media

revised: March 25, 2008

Presentation Overview

1. Current Projects

2. The Website: openLiberty.org

3. ID-WSF 2.0 Client Library

overview

clients to date

other code of note

5 step start

sample code

4. FireFox ECP Plugin

5. ZXID is now part of OpenLiberty!

6. Conor Cahill's WSC and WSP Are now part of OpenLiberty!

7. Currently in progress or near future

8. Contact Information



Current Projects (March 2008)

ID-WSF 2.0 Client Library

Open Source Web Service Client (WSC) implementations with the intention of being WSC feature complete and interoperability tested for the WSC and WSC LUAD Liberty Profiles.

Identity Governance Framework (IGF)

IGF will help to enable identity-consuming applications to bind governance policies (consent and constraints) to the identity data they receive and ensure those policies are enforced whenever any other IGF-enabled application tries to access that data at a later time.

FireFox ECP Plugin

An XUL and ECMAScript (Javascript) FireFox extension implementing the UserAgent Enhanced Client or Proxy (ECP) SAMLv2 profile.

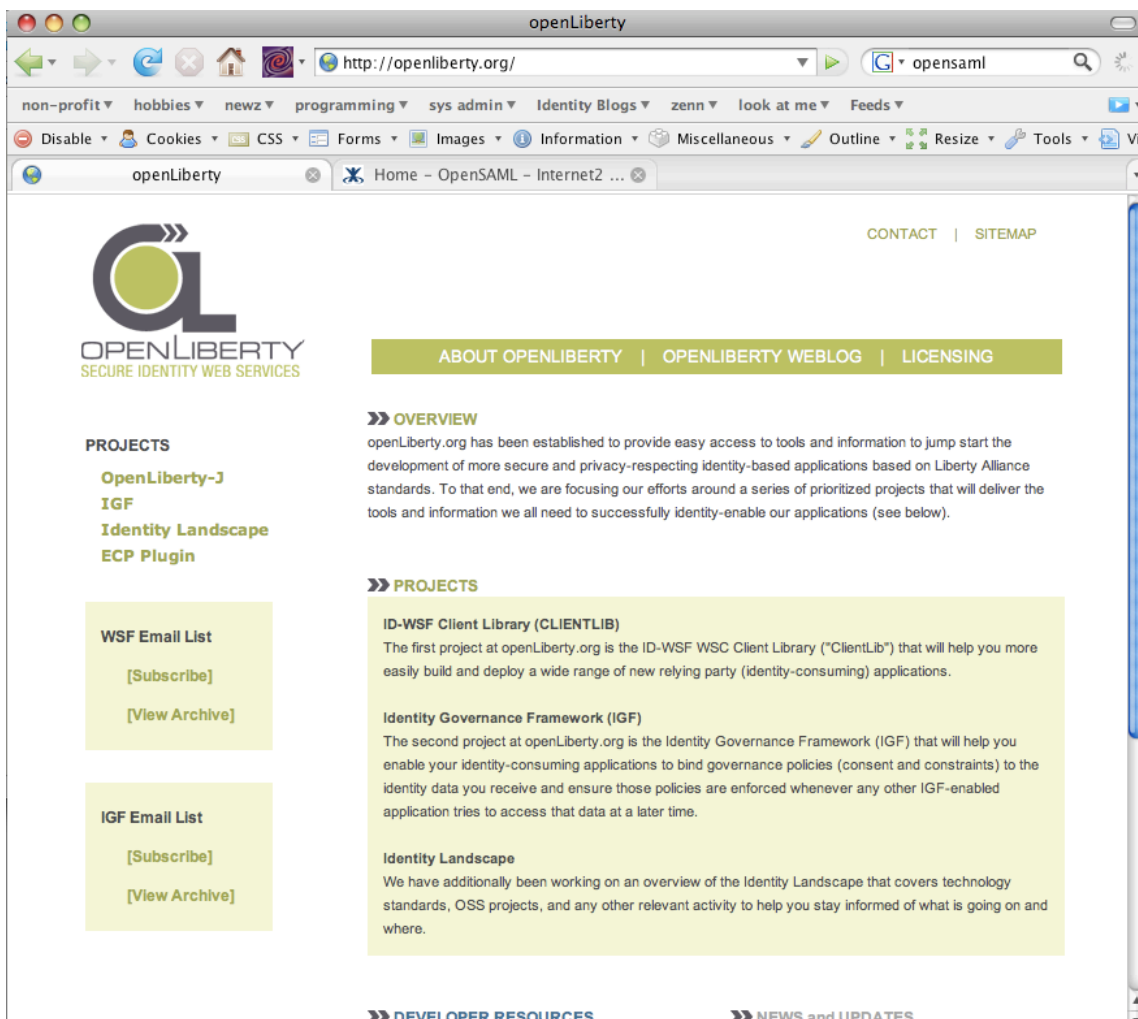
Identity Landscape

A map of existing technologies with a goal of finding a unifying mechanism to characterize these technologies and systems, and discuss social, regulatory and business aspects of them. In addition, known open source projects and commercial products that are relevant in the identity landscape.

The Website: *openLiberty.org*

Tools and Communication

The website is the primary vehicle for collaboration, communication, documentation, and project organization. Along the left side is a list of projects. There is a unified web log for news and updates relevant to OpenLiberty. A shared wiki serves as a repository for documentation. Source code is currently hosted in SVN and CVS on sourceforge.net



openLiberty

http://openliberty.org/

non-profit ▾ hobbies ▾ newz ▾ programming ▾ sys admin ▾ Identity Blogs ▾ zenn ▾ look at me ▾ Feeds ▾

Disable ▾ Cookies ▾ CSS ▾ Forms ▾ Images ▾ Information ▾ Miscellaneous ▾ Outline ▾ Resize ▾ Tools ▾ Vie

openLiberty

CONTACT | SITEMAP

OPENLIBERTY
SECURE IDENTITY WEB SERVICES

ABOUT OPENLIBERTY | OPENLIBERTY WEBLOG | LICENSING

» OVERVIEW

openLiberty.org has been established to provide easy access to tools and information to jump start the development of more secure and privacy-respecting identity-based applications based on Liberty Alliance standards. To that end, we are focusing our efforts around a series of prioritized projects that will deliver the tools and information we all need to successfully identity-enable our applications (see below).

» PROJECTS

ID-WSF Client Library (CLIENTLIB)
The first project at openLiberty.org is the ID-WSF WSC Client Library ("ClientLib") that will help you more easily build and deploy a wide range of new relying party (identity-consuming) applications.

Identity Governance Framework (IGF)
The second project at openLiberty.org is the Identity Governance Framework (IGF) that will help you enable your identity-consuming applications to bind governance policies (consent and constraints) to the identity data you receive and ensure those policies are enforced whenever any other IGF-enabled application tries to access that data at a later time.

Identity Landscape
We have additionally been working on an overview of the Identity Landscape that covers technology standards, OSS projects, and any other relevant activity to help you stay informed of what is going on and where.

» DEVELOPER RESOURCES

» NEWS and UPDATES

PROJECTS

OpenLiberty-J
IGF
Identity Landscape
ECP Plugin

WSF Email List
[Subscribe]
[View Archive]

IGF Email List
[Subscribe]
[View Archive]

ID-WSF 2.0 Client Library: overview

Java (J2SE 1.5)

The client library is 100% java. The packages currently fall into two categories, XML Tooling, which models the XSD files and specifications, and WSC which contains service clients, SOAP, and WSF Message implementation.

Apache 2 Licence

Open Source and under the extremely flexible Apache 2 license (as are all projects on OpenLiberty) allowing for maximum use for commercial and non-commercial products.

OpenSAML basis



The Client library heavily leverages the work done by Internet2. Java XML Tooling, Java SAML2, and Java WS libraries form the basis for the xmltooling layer. The advantages of this relationship include easy integration with Shibboleth, OpenSAML signature and encryption, and the proper handling of SAMLv2 Assertions.

Simplicity Without Sacrifice

The ID-WSF Client Library has been built from the ground up with the intention of serving as both an extremely simple way for a Service Provider (SP) to participate in an ID-WSF 2.0 environment and also a thorough implementation offering as much access to low level features as the Specification provides.

Beta Available Now!

A Beta release has been posted which is ready for interested parties to test and begin the consideration for use. The code has not been formally tested, although portions have been tested against Symlabs Federated Identity Suite and Conor Cahill's Open Source WSP.

ID-WSF 2.0 Client Library: clients to date

Discovery Service (DS) Client

The disco client has been tested with both Symlab's Federated Identity Suite and Conor Cahill's open source WSP.

XML Tooling: XSD Complete

Service Client: Complete

Authentication Service (AS) Client

The AS client has been tested with both Symlab's Federated Identity Suite and Conor Cahill's open source WSP. It is currently capable of handling authentication using SASL with both PLAIN and CRAM-MD5 Mechanisms. The point of entry for the library is currently through the AS Client which expects a Discovery Service Endpoint Reference (Disco EPR).

XML Tooling: XSD Complete

Service Client: PLAIN and CRAM-MD5 SASL

Personal Profile Service (PP) Client

The PP client is built on top of DST 2.1 and the DST 2.1 reference implementation. This is a non-standard implementation as the specification is for DST 1.1. However, with a few minor modifications this implementation was tested showing initial success with Symlab's PP.

XML Tooling: XSD Complete

Service Client: Complete

People Service (PS) Client

The PP client is built on top of DST 2.1 and the DST 2.1 reference implementation. This is a non-standard implementation as the specification is for DST 1.1. However, with a few minor modifications this implementation was tested showing initial success with Symlab's PP.

XML Tooling: XSD Complete

Service Client: In Progress, Stubbed and partially implemented.

Directory Access Protocol (ID-SIS-DAP) Client

The PP client is built on top of DST 2.1 and the DST 2.1 reference implementation. This is a non-standard implementation as the specification is for DST 1.1. However, with a few minor modifications this implementation was tested showing initial success with Symlab's PP.

XML Tooling: XSD Complete

Service Client: Partially Complete



ID-WSF 2.0 Client Library: other code of note

Data Services Template (DST 2.1)

The XML Tooling for the DST is complete and DST reference implementation is complete for Query and Modify. The Personal Profile Service Client uses the DST and DST ref. ID-SIS-DAP utilizes the DST.

WSFMessage

All messages are created and invoked through this class. It handles EndpointUpdate, RedirectRequest, and the construction of the SOAP Headers specified in Liberty SOAP Bindings.

WSFMessageSigner

This class handles the proper signing of the body and specific header elements within the WSFMessage when SAMLv2 security mechanism is used.

IMS, Subscriptions & Notifications, Utility 2.0, some WS-security, some WS-Addressing

These XML Tooling components are heavily leveraged throughout the library.

Liberty SOAP Bindings

All ID-WSF Messaging is over the SOAP transport and utilizes a number of header bindings which contain security tokens,

ID-WSF 2.0 Client Library: 5 Step Start

(Note: available on website in more detail)

1. Usage Requirements

Java IDE (eclipse, netbeans), J2SE 1.5, and Subversion (svn).

2. Get the OpenSAML Libraries

You will need three java libraries developed by Internet2 through subversion repo access.

- Java XML Tooling
- Java SAML2
- Java WS

3. Get the ID-WSF 2.0 ClientLib

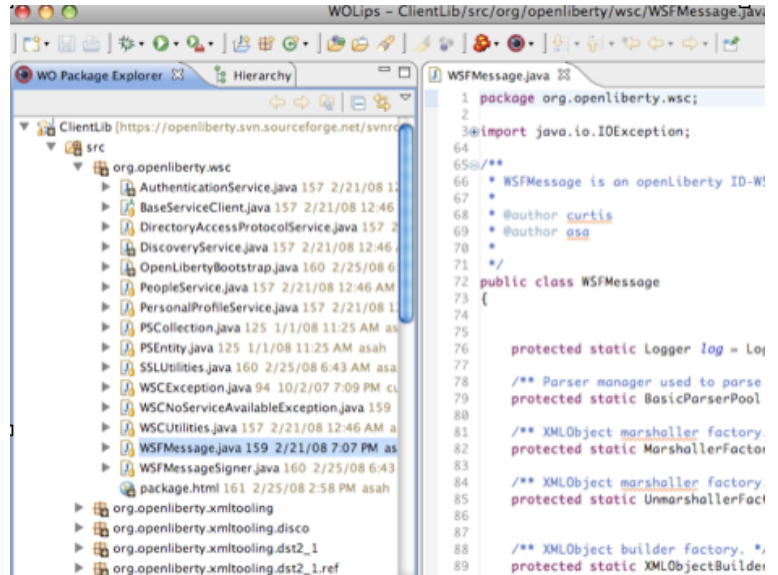
Again, this requires access to subversion, source available on sourceforge.

4. Import into Eclipse (or your IDE)

The projects you have just checked out will need to be imported into your IDE. Or, you may have used your IDE to check out the projects.

5. Run the test

Once you've dealt with all of your classpath issues, you are ready to run a test that runs through some Authentication Service (AS) queries, extracts a Discovery Endpoint Reference and then makes a series of Discovery Service (DS) queries. This will be fun! All of the tests are live and connect to Conor Cahill's ID-WSF 2.0 WSP.



ID-WSF 2.0 Client Library: sample code

Using the WSCUtilities class

Delivering on simplicity, the WSCUtilities class contains methods which combine a large number of steps into a single call.

STEP 1: Set Up default AS

```
// Setup for the Default Authentication Service
OpenLibertyBootstrap.setAuthenticationServiceUrl(SERVICE_URL);
OpenLibertyBootstrap.setAuthUsername(USERNAME);
OpenLibertyBootstrap.setAuthPassword(PASSWORD);
SSLUtilities.loadX509Certificate(PATH_TO_CERT);
```

STEP 2: Authenticate and Discovery Bootstrap

```
// Get an Endpoint Reference for the Discovery Service,
// using PLAIN authentication
EndpointReference epr;
DiscoveryService ds;

epr = WSCUtilities.bootstrapDiscoveryEPRFromDefaultAS(
    AuthenticationService.AuthMechanism.PLAIN);

// Query the discovery service for the Discovery Service
// (Self referential test)
ds = (DiscoveryService)WSCUtilities.clientForDiscoveryEPR(
    discoEPR,
    DiscoveryService.WSFServiceType.DISCOVERY_SERVICE);
```

Advanced Access

Not only do you have deeper access to the methods within each service client, but you have complete access to the XML Tooling library. This can be used to create objects and requests from scratch. An increasing number of these classes have constructors that take a simple set of arguments. It is also easy to follow the design pattern of the xml tooling packages or the Service Clients and create your own client.

Firefox ECP Extension

Firefox

Because Firefox runs on many platforms, is highly standards compliant, and has a very powerful yet simple extensions framework it was chosen as the first UserAgent to be enabled with an ECP plugin.

ECMAScript (Javascript) and XUL - Easily Extensible

The role of the SP is well defined in ECP, the role of the IdP is a little more loosely defined as there may be any number of authentication requirements. Javascript and XUL are easy to modify and the choice was made in the hope that modifications would be encouraged.

Shibboleth and/or Symlabs Federated Identity Suite Integration

The version 1 release will be certified to work with Shibboleth and/or Symlabs FIS



Available June 30th 2008!

ZXID is Now Part of OpenLiberty!

ZXID Project, five outputs

ZXID consists of C libraries. Some of these libraries are generated from *schema grammar* descriptions using a tool called `xsd2sg.pl`, part of Plaindoc distribution. Other libraries that express flows and processing rules are hand-written. The language bindings, other than C, are generated automatically using *swig(1)*.

libzxid

A C library for supporting SAML 2.0, including federated Single Sign-On (SSO)

zxid

A C program that implements a SAML Service Provider (SP) as a CGI script

Net::SAML

A Perl module wrapping `libzxid`. Also `zxid.pl`, that implements SP in `mod_perl` environment, is supplied.

php_zxid

A PHP extension that wraps `libzxid`. Also supplied: `zxid.php` that implements SP in `mod_php` environment.

libzxidjni.so

A Java JNI extension that wraps `libzxid`. Also supplied: `zxid.java` that implements SP as a CGI script. `zxidhlo.java` demonstrates use under servlet engine, e.g. Tomcat.

Aims of ZXID Project

ZXID aims at *full stack* implementation of all federated identity management and identity web services protocols. Initial goal is supporting SP role, followed by ID-WSF WSC and IdP roles. We aim at supporting US GSA E-Auth profile.

ZXID is light weight, has a small foot print, and is implemented in C. It is suitable for both high performance and embedded applications. Scripting languages are supported using SWIG, including Perl, PHP and Java. The "full stack" nature of ZXID means it's self contained and has minimal external library dependencies (see downloads).

More Information

Contact: Sampo Kellomäki (sampo@iki.fi)

Project Site: <http://www.openliberty.org>

Conor Cahill's ID-WSF OSS is Now Part of OpenLiberty!

Client Toolkit

The client toolkit is a C++ library that supports the Liberty ID-WSF 1.0 and 2.0 framework protocols. In addition to the base ID-WSF framework, the toolkit also includes client modules for interacting with the following services:

- *Liberty ID-WSF Authentication Service*
- *Liberty ID-WSF Discovery Service*
- *Liberty ID-WSF Provisioning Service*
- *Demo Radio Service (non-Liberty)*
- *Demo Media Service (non-Liberty)*
- *Demo Registration Service (non-Liberty)*
- *Demo Provisioning Service (non-Liberty)*

Server Toolkit

The server toolkit is a Java implementation. The following service components are included

- Liberty ID-WSF Authentication Service (very basic, only supports Username/Password validation)
- Liberty ID-WSF Discovery Service - fairly complete
- Liberty ID-WSF Provisioning Service - substantial portions, but NOT complete (activate/deactivate missing, polling/notification not supported)
- Demo Media Service (non-Liberty)
- Demo Registration Service (non-Liberty)
- Demo Provisioning Service (non-Liberty)
- Early Beginings (non-functional) Liberty Identity Mapping Service and and Liberty People Service

More Information

Contact: Conor Cahill

Project Site: <http://www.cahillhome.com/OpenSource/>



Currently in progress or near future

Publicly available testing environment

Sample ID-WSF 2.0 Application

Official Liberty Interoperable

Q3 2008 final release of WSC Code

More Service Clients!



Contact Information

Asa Hardcastle

Skype: subsystem7

email: asa.openliberty@zenn.net

cell phone: 413 429 1044