Eclipse Lyo Project

Eclipse Lyo (tentative name) is a proposed open source project under the Eclipse Technology top-level project. This proposal is in the Project Proposal Phase (as defined in the Eclipse Development Process) and provides the project's intent and scope. We are soliciting additional participation and feedback from the Eclipse community regarding this proposal. Please provide any feedback via the Eclipse Proposals forum.

Abstract

The Eclipse Lyo project is focused on providing an SDK to enable adoption of OSLC specifications. OSLC (Open Services for Lifecycle Collaboration: http://open-services.net) is an open community dedicated to reducing barriers for lifecycle tool integration. The community authors specifications for exposing lifecycle artifacts through uniform (REST) interfaces and relying on Internet and Linked Data standards. OSLC's scope started with Application Lifecycle Management[1] and is expanding to include integrations across Product Lifecycle Management[2] and IT operations[3]. The proposed Eclipse Lyo project is designed to be a companion to the continuing specification efforts of the OSLC community. Its main purpose is to expand adoption of OSLC specifications and to enable the Eclipse community to easily build OSLC compliant tools.

Background

The OSLC community has collected initial components for an OSLC SDK in an open source project, the OSLC Tools Project, hosted on SourceForge.net (http://sourceforge.net/projects/oslc-tools/). The OSLC Tools codebase includes reference implementations for some OSLC specifications, a test suite for validating OSLC provider implementations, and a set of examples and code snippets that aid in OSLC client and provider implementations. Contributions were made under an Apache-like contributor license agreement, and the content is currently distributed under the terms of the Apache Software License 2.0. The project has four committers. The intention of this proposal is to move the OSLC Tools Project, in its entirety, to Eclipse and to evolve the SDK in support of OSLC specifications.

Scope

The focus of the proposed Eclipse Lyo project is to provide an SDK -- including candidate reference implementations, test suites, and code libraries -- for enabling OSLC implementations by tools in the ALM, PLM, and IT Operations lifecycle.

Out of Scope

Eclipse Lyo is not a project tasked with creating OSLC compliant tools. Instead, it is a project that will create an SDK for enabling others to create OSLC-based tool integrations. Further, the project is not for general purpose linked data technologies, though it will certainly rely on other OSS projects for these capabilities (e.g. RDF triple store).

Why Eclipse?

The Eclipse Foundation and open source community have a well-established governance model and set of IP policies. Furthermore, its broad industry membership and its focus on development tools have affinity with OSLC's scope and can help to further the Lyo project's goals.

The proposed Eclipse Lyo project complements other Eclipse projects like Mylyn and Orion. Mylyn is already aligned with OSLC and is creating client side implementations of OSLC specifications for bringing tasks into the developer IDE. Orion, with its web architecture and tooling, also has potential affinities as a client of OSLC implementations and may find benefit from the Eclipse Lyo project and OSLC SDK.

Initial Contribution

The initial contribution to the proposed Eclipse Lyo project will consist of the entire codebase that is currently part of the OSLC Tools Project at SourceForge. This includes:

- **Reference Implementations**: including Java EE based web applications that implement the OSLC lifecycle domain specifications for Change and Architecture Management. The main OSS dependencies are Geronimo Servlet API, Sesame/OpenRDF and the ANTLR parser generator.
- **Provider Test Suites**: Java-based test suites designed to test implementations of each OSLC lifecycle domain specification. The main OSS dependencies are JUnit, Apache HTTPClient, Jena and OAuth.net libraries.
- **Example OSLC resources.** Example OSLC resources in RDF/XML and other RDF format for educational and testing purposes.
- **Example OSLC provider and client implementations.** In addition to the reference implementations, the OSLC Project provides other example OSLC implementations, including a JAX-RS and Apache Wink based client and server (see table below).
- **Tutorial example code.** These educational examples are not technically part of the OSLC Tools Project yet, but may be contributed by the time a

move to Eclipse made such as the code to support OSLC Tutorial and Primer.

The current codebase is mostly Java code with dependencies on permissively licensed open source libraries. Moving forward the project will encourage contributions of code, candidate reference implementations, test suites, and examples for other platforms and languages.

The table below lists the current dependencies of the OSLC Tools Project codebase that forms the initial contribution:

Note: in the case of a dual license, we elected to obtain and use the code under the license indicated in **bold** text

Name	License	URL
Aduna Commons 2.6, 2,7, 1.3	BSD	https://repo.aduna-software.org/
Ant and Ant-launcher 1.7.1	ASL2	http://ant.apache.org
Antlr 2.7.7, 3.2	BSD	http://www.antlr.org/
Commons Codec 1.3	ASL2	http://commons.apache.org/codec
Commons Collections 3.2.1	ASL2	http://commons.apache.org/collections
Commons Fileupload 1.2.2	ASL2	http://commons.apache.org/fileupload/
Commons HttpClient 3.1	ASL2	http://hc.apache.org/httpclient-legacy/
Commons IO 1.2	ASL2	http://commons.apache.org/io/
Commons Lang 2.3	ASL2	http://commons.apache.org/lang
Commons Logging 1.1.1, 1.0.4	ASL2	http://commons.apache.org/logging
Commons Pool 1.5.3	ASL2	http://commons.apache.org/pool
Concurrent Jena 1.3.2	BSD	http://openjena.org/
Derby and Client 10.1.3.1	ASL2	http://derby.apache.org
Dojo 1.6	**BSD** / Academic Free License	http://dojotoolkit.org/
DOM4J 1.6.1	Apache 1.1-style	http://dom4j.sourceforge.net/
Geronimo J2EE API 1.1	ASL2	http://geronimo.apache.org
Geronimo JTA API 1.1.1	ASL2	http://geronimo.apache.org
Geronimo JMS API 1.1.1	ASL2	http://geronimo.apache.org
Geronimo JPA API 1.0	ASL2	http://geronimo.apache.org

Geronimo Servlet API 1.2	ASL2	http://geronimo.apache.org
HTTP Client 4.0.1	ASL2	http://hc.apache.org
HTTP Core 4.0.1	ASL2	http://hc.apache.org
ICU4J 3.4.4	MIT	http://site.icu-project.org/home
Jackson 1.5.2	ASL2	http://jackson.codehaus.org/
Java Beans Activation Framework 1.1	**CDDL** / GPL2	http://glassfish.java.net/javaee5/mail/
JAXB API 2.1 and JAXB-impl 2.1.4	**CDDL** / GPL2	http://jaxb.java.net/
Jena 2.6.3 (arq, iri, jena, json-jena)	BSD	http://jena.sourceforge.net
Jdom 1.1	Apache 1.1-style	http://www.jdom.org/
Jetty 6.1.24	ASL2	http://jetty.codehaus.org/jetty/
JSR 311 API	CDDL	http://jersey.java.net/
JUnit 4.8.1	CPL	http://junit.org/
Log4J 1.2.13	ASL2	http://logging.apache.org/log4j/
Lucene Core 2.3.1	ASL2	http://lucene.apache.org
Maven 3.0	ASL2	http://maven.apache.org
OAuth.net v20090617	ASL2	http://code.google.com/p/oauth/
POI 3.7	ASL2	http://poi.apache.org
OpenJPA 2.0	ASL2	http://openjpa.apache.org
OpenRDF Sesame 2.3.2	BSD	http://sourceforge.net/projects/sesame/
Serp 1.13.1	BSD	http://serp.sourceforge.net/
SLF4J 1.5.11	MIT	http://www.slf4j.org/
STAX API 1.0.1	ASL2	http://stax.codehaus.org
String Template 3.2	BSD	http://www.stringtemplate.org/
Wink 1.1.1	ASL2	http://wink.apache.org
Wink JSON4J 1.1.2	ASL2	http://wink.apache.org
WS commons util	ASL2	http://ws.apache.org/xmlrpc/
WSTX ASL 3.2	**ASL2** / LGPL	http://woodstox.codehaus.org
Xalan 2.6	ASL2	http://xml.apache.org/xalan-j/
Xerces 2.7.1	ASL2	http://xerces.apache.org

XML Parser APIs 1.3.02	ASL2	http://xerces.apache.org
XML-RPC 3.1.3	ASL2	http://ws.apache.org/xmlrpc/

Note that the dependencies above are declared in Maven pom.xml files and the actual JAR files that we depend on are not included in the OSLC Tools SVN repository.

The OSLC Tools Project has not yet done any “releases” of software and so has had no reason to distribute the above JAR files, but we do expect to create releases and distribute the above dependencies from the Eclipse Lyo project as parts of reference implementations, example bundles, and an SDK.

Legal Issues

The proposed initial contributions are currently made available under an Apache 2.0 license. As the proposed Eclipse Lyo project is formed, the contributions will need to be made under the Eclipse Public License. Due to the nature of the project and its materials (JavaScript, RDF documents and other files that are not source and not binary, as well as samples, test suites, and reference implementations that are intended for download and customization), dual licensing is required under a more permissive license, hence we would like to propose the Eclipse Distribution License (EDL).

Committers

- **Dave Johnson**, IBM (Committer and Project Lead). Works on the reference implementations, example implementations, and the OSLC Tutorial.
- **Steve Speicher**, IBM (Committer and Project Lead). Developed the OSLC Provider Test Suites, works on the reference implementations and other examples. .
- **Olivier Berger**, Institut Telecom (Committer). Committer to the project, has contributed to design discussions, sample code in PHP format, oversight and OSLC specification effort.
- **Jim Conallen**, IBM (Committer). Developed the initial version of the Reference Implementations including an OSLC-AM and OSLC-CM implementation.
- **Sofia Yeung**, Oracle (Committer). Committer to the project, has contributed to design discussions, oversight and OSLC specification effort.
- **Robert Elves**, Tasktop (Committer). Contributed to OSLC-CM specification and implemented Mylyn OSLC implementation.

Interested Parties

Accenture
Ericsson
General Motors
IBM
Oracle
Tasktop
Tieto

Changes to this Document

Date	Change
May 17,2011	Document created

References

1- http://en.wikipedia.org/wiki/Application_lifecycle_management
2- http://en.wikipedia.org/wiki/Product_lifecycle_management
3- http://en.wikipedia.org/wiki/IT_service_management