Web Tools Platform Incubator
XSL Graduation and Move Review

Graduation and Move Review Materials
March 18, 2008
Prepared by David Williams, David Carver, and Doug Satchwell

Table of Contents

Introduction and Purpose

This document is to fulfill the requirements of the Eclipse Graduation Review for WTP XSL Graduation and Move to WTP Source Editing, planned for March, 2008.

The communication channel for the review is the webtools newsgroup. Use the newsgroup to post any comments or to request the review to be held on a conference call.

History

The Eclipse XSL component was originally proposed in November of 2007. See the original mailing list post and the original XSL proposal. It had a pre-1.0 release in August, 2008.

Project and Component Organization

The XSL component is one of the components of the WTP Incubating Project, which itself is a sub-project of the Web Tools Platform project.

WTP

Below is the list of current WTP projects and project leads as of March, 2009.

Project	Lead
Common: tools and infrastructure not directly related to web tools, but required by Web Tools Platform	Konstantin Komissarchik, Oracle
Dali (JPA Tools): infrastructure and tools for JPA applications	Neil Hauge, Oracle
EJB Tools: EJB creation wizards, preferences, future annotation tools	Kaloyan Raev, SAP
Java EE Tools: Common Project Infrastructure, JEE models, preferences, classpath model, publish api, refactoring	Chuck Bridgham, IBM
JSF Tools: infrastructure and tools for Java Server Faces.	Raghu Srinivasan, Oracle
Server Tools: tools and infrastructure to define and interact with servers.	Tim Deboer, IBM
Source Editing: xml, dtd, xsd (and sse infrastructure) html, css, javascript, jsp	Nitin Dahyabhai, IBM
Web Services: Web services wizards and frameworks, Axis1 & Axis2 support, Web Servcies Explorer, WSDL Editor	Keith Chong, IBM
Release Engineering: contains the code and scripts to do builds, various tests, API scans, etc.	David Williams, IBM
WTP Incubator: a general purpose incubation project other WTP Projects to use when incubation is desired.	David Williams, IBM
ATF (incubating): infrastructure and tools for AJAX	Philippe Ombredanne, nexB
Datatools (RDB): tools for working with databases. Primarily moved to DTP, but the quiescent sub-project of WTP is doing 1.5 maintenance	Larry Dunnell, IBM

WTP PMC Organization

Our Project Management Committee, as of March, 2009, is made up of 6 members, each having a WTP-wide management role, in addition to what ever project-specific roles they have. In the execution of their tasks within these roles, the PMC members will form groups, organize meetings, etc., to accomplish their goals. In other words, they don't do all the work ... they just manage it!

Member	Role
David Williams, IBM	PMC Lead, and Planning Role
Tim Deboer, IBM	Architecture
Neil Hauge, Oracle	Quality
Kaloyan Raev, SAP	User Experience
Raghu Srinivasan, Oracle	Requirements
Naci Dai, Eteration	Education

WTP Incubating Project

The WTP Incubating project was initially formed with all the WTP Project leads as seed committers, so they could monitor, mentor, and govern the incubating components. Each incubating component must be aligned with one of the main sub-projects in WTP and in this case, XSL is affiliated with the Source Editing subproject. This affiliation primarily is required to make sure our incubating projects stay in scope of WTP, but secondarily to decide in advance where its likely home would be once the component graduates.

XSL Component

While technically, all WTP Incubating committers are committers on any sub-component of the incubating project, in practice, there are component teams that do the work. For the XSL component, the team comprises:

Doug Satchwell	Lead
David Carver	

Specific CVS Code involved

From the `/cvsroot/webtools` repository, the xsl 'incubator' modules would move to the 'sourceediting' directory. Specifically:

1. incubator/sourceediting/documentation/org.eclipse.wst.xsl.doc
2. incubator/sourceediting/documentation/org.eclipse.wst.xsl.sdk.documentation
3. incubator/sourceediting/features/org.eclipse.wst.xsl.feature
4. incubator/sourceediting/features/org.eclipse.wst.xsl_sdk.feature
5. incubator/sourceediting/features/org.eclipse.wst.xsl_tests.feature
6. incubator/sourceediting/plugins/org.eclipse.wst.xml.xpath.core
7. incubator/sourceediting/plugins/org.eclipse.wst.xml.xpath.ui
8. incubator/sourceediting/plugins/org.eclipse.wst.xml.xpath2.processor
9. incubator/sourceediting/plugins/org.eclipse.wst.xsl
10. incubator/sourceediting/plugins/org.eclipse.wst.xsl.core
11. incubator/sourceediting/plugins/org.eclipse.wst.xsl.debug.ui
12. incubator/sourceediting/plugins/org.eclipse.wst.xsl.jaxp.debug
13. incubator/sourceediting/plugins/org.eclipse.wst.xsl.jaxp.debug.ui
14. incubator/sourceediting/plugins/org.eclipse.wst.xsl.jaxp.launching
15. incubator/sourceediting/plugins/org.eclipse.wst.xsl.launching
16. incubator/sourceediting/plugins/org.eclipse.wst.xsl.saxon
17. incubator/sourceediting/plugins/org.eclipse.wst.xsl.ui
18. incubator/sourceediting/plugins/org.eclipse.wst.xsl.xalan
19. incubator/sourceediting/tests/org.eclipse.wst.xml.xpath.core.tests
20. incubator/sourceediting/tests/org.eclipse.wst.xml.xpath2.processor.tests
21. incubator/sourceediting/tests/org.eclipse.wst.xsl.core.tests
22. incubator/sourceediting/tests/org.eclipse.wst.xsl.launching.tests
23. incubator/sourceediting/tests/org.eclipse.wst.xsl.tests
24. incubator/sourceediting/tests/org.eclipse.wst.xsl.ui.tests

would move, repectively, to

1. sourceediting/documentation/org.eclipse.wst.xsl.doc
2. sourceediting/documentation/org.eclipse.wst.xsl.sdk.documentation
3. sourceediting/features/org.eclipse.wst.xsl.feature
4. sourceediting/features/org.eclipse.wst.xsl_sdk.feature
5. sourceediting/features/org.eclipse.wst.xsl_tests.feature
6. sourceediting/plugins/org.eclipse.wst.xml.xpath.core
7. sourceediting/plugins/org.eclipse.wst.xml.xpath.ui
8. sourceediting/plugins/org.eclipse.wst.xml.xpath2.processor
9. sourceediting/plugins/org.eclipse.wst.xsl
10. sourceediting/plugins/org.eclipse.wst.xsl.core
11. sourceediting/plugins/org.eclipse.wst.xsl.debug.ui
12. sourceediting/plugins/org.eclipse.wst.xsl.jaxp.debug
13. sourceediting/plugins/org.eclipse.wst.xsl.jaxp.debug.ui
14. sourceediting/plugins/org.eclipse.wst.xsl.jaxp.launching
15. sourceediting/plugins/org.eclipse.wst.xsl.launching
16. sourceediting/plugins/org.eclipse.wst.xsl.saxon
17. sourceediting/plugins/org.eclipse.wst.xsl.ui
18. sourceediting/plugins/org.eclipse.wst.xsl.xalan
19. sourceediting/tests/org.eclipse.wst.xml.xpath.core.tests
20. sourceediting/tests/org.eclipse.wst.xml.xpath2.processor.tests
21. sourceediting/tests/org.eclipse.wst.xsl.core.tests
22. sourceediting/tests/org.eclipse.wst.xsl.launching.tests
23. sourceediting/tests/org.eclipse.wst.xsl.tests
24. sourceediting/tests/org.eclipse.wst.xsl.ui.tests

Characteristics of a Project ready to graduate and how XSL meets those criteria

Taken from Criteria for Graduating

A working and demonstrable code base with extensible frameworks and exemplary tools

- XSLT Debugging using Xalan 2.5.1 or higher – Allows for the debugging of XSL files using the eclipse debug framework. Users can set break points and view variables and nodes.
- XML Perspective – Working with XML only projects is not necessarily the same as working on Web Projects or JEE projects. The XML perspective provides the views and necessary shortcuts that are most commonly used with XML projects.
- XSLT Launch and Debug configurations – The user has the ability to create Launch and Debug scenarios for XSL. This allows for easier execution of XSL transformations. Users have the ability to pass varables, execute transformations in a pipeline (i.e. execute XSL 1, XSL 2, XSL 3, XSL 4, passing the output of the first transformation to the second).
- XSLT Aware XML Editor – the XSL Tools extends the existing WTP XML editor to provide XSL specific functionality. It provides awareness of XSL 1.0 or XSL 2.0 stylesheets and loads the appropriate grammar for validation. Content assistance is available for select and test attributes for Xpath expression completion.
- XSLT and Xpath validation – The XSL Tools leverages the WTP 3.0 Validation framework to provide as you type validation as well as Batch validation using the existing Validation Framework and builders.
- Xpath View - This allows the user to run an Xpath against the currently active XML document. If the document changes, or the currently active editor changes, the Xpath is re-run.
- XSLT New File Wizard – Allows for the creation of XSL specific files based on XSL Templates defined in the Preference pages. Users can add new XSL Templates that can be made available to the wizard.
- XSLT Processor Invoker and Debugger Extension Points – Adopters can add their own specific XSLT processors and debugging support. By implementing the XSLT debugging interfaces, adopters can add support for their own processors within XSL Tools. There is exemplary support for JAXP compliant processors, but it is also straightforward to add support for other processors and languages, such as XalanC for C++.
- Xinclude ANT Task – An xinclude ant task has been included to allow for the batch execution of Xinclude processing. Xinclude allows you to combine multiple XML files into one file. It uses and Xpath extension language called Xpointer. It is commonly used in DITA and DocBook authoring scenarios.
- Xpath Templates – The XSL Tools project includes templates for content assistance for Xpath 1.0 for axis, operations, and functions. It also includes place holders for Xpath 2.0 and EXSLT extensions.
- Developer and API Documentation
    - Developer SDK Documentation for API on Xpath View, and XSLT Debugging.
    - Adopters can use the standard org.eclipse.text.templates extension point to contribute additional templates to both XSLT and Xpath.
- End-User Documentation and Examples
    - The XSLT end user documentation is included for the all of the XSL Tools functionality.

- General Documentation is in place for Launching and Debugging Transformations

Active communities

Committers and Contributors

There a number of active committers from several companies (including individuals, not affiliated with a member company).

- Committer elections and removals have followed charter principles
- Continuing to recruit additional contributors (organizations and individuals)
- Open communications via mailing lists and newsgroups: wtp-pmc, wtp-dev, wtp-releng, wtp-incubating-dev.

Open and inclusive release planning and tracking processes

- Bugzilla used to request and track all defects, enhancements, and milestone plans
- Additional reports (defect summaries, test stats, etc) used to enhance planning / tracking

All contributions made directly to Eclipse CVS

- Nightly, weekly integration builds, and release builds available to the community

Leveraged other open source technologies in the project

- Built upon Eclipse Platform and WTP components plus some common engines from XSL open source community.

End-User Community

Bugs activity from end-user community.

Of the 236 bugs open, since inception, 29 of those were from user ids outside the immediate team and interested parties. While not a huge number, it is 12% which indicates a significant end-user community.

Steady download activity for each milestone.

- While the absolute numbers are not accurate (since many downloads and installs are done from mirror sites, which are not counted) we take the relatively steady download activity to indicate a moderate and sustained interest in this XSL component [these numbers are from the pre-1.0 release time period]:

Milestone	Downloads
M8	543
M7	509
M6	316
M5	527
M4	400

Substantial website content

- Download links, New & Noteworthy, mailing lists, presentation DB
- Tutorials, documentation, presentation summary,
  - Doug Satchwell has written an article and submitted to the Eclipse Corner for possible publication.
  - David Carver has written an article on Authoring with Eclipse that uses XSL Tools for the publishing portion of DocBook related content.

Evangelism and outreach in the market & broader community

- David Carver an XSL Tools Committer has several blog postings available at his blog.
  - Http://intellectualcramps.blogspot.com/

- David Carver attend EclipseCon 2008 and discussed with various parties their about the XSL Tools project.
- David Carver presented an XSL Tools and XSLT Tutorial at EclipseCon 2009.
- David Carver presented XML Development Tools at EclipseCon 2009 including information on XSL Tools.

Adopter Community

- STAR - Standards for Technology in Automotive Retail is an XML Business to Business Standards Organization developing B2B standards for the Automotive, Marine, Powersports, and Heavy Duty truck industries. STAR has a member XML IDE. STAR will be releasing it's latest version of the STAR Workbench in October which will use Eclipse 3.4, Web Standard Tools 3.0, and the XSL Tools incubator to provide a XML IDE to it's members.
- XMLOpt - is an XML editing and processing environment. It supports XML, DTD, XML Schema, XSLT, and Docbook editing. Also included is support for the XML database eXist. XMLOpt is based on the Eclipse 3.4 platform. Currently only Linux distribution available

Adopters are encouraged to publicly state their support and usage on the XSL Tools Adopters page. http://wiki.eclipse.org/XSLT_Project/Adopters

Operating fully in the open using open source rules of engagement

Open and transparent Bugzilla with a described and documented bug process.

Use Eclipse Bugzilla

Open and transparent project schedules.

Schedules have been published on the development wiki

The project has working policies and procedures for developing, specifying, testing, and getting feedback on APIs.

In general we provide APIs according to Eclipse Quality API standards.

The project decision making processes are published, and all project decisions are being made in public.

Discussions on newsgroups, mailing lists, and wiki

The project "gets the Eclipse way".

Abides by the base Eclipse Development Process.

Adheres to the Eclipse IP Policy and each Committer is following the committer responsibilities and due diligence rules.

Participates in the larger Eclipse community. For example, tutorials at EclipseCon or other conferences, writing articles, participating in the Reviews of other projects, etc.

Works with other projects in its enclosing top-level Project.

The project is a credit to Eclipse and is functioning well within the Eclipse community, such as document all our new and noteworthy items for each milestone

Dependencies and interactions with other projects.

Follows W3C and OASIS standards

- XSLT 1.0, XSLT 2.0
- Xinclude 1.0
- Xpath 1.0, Xpath 2.0

Familiar with, and follows, the Eclipse User Interface Guidelines .

We have participated in a UI walk through, prior to this release.

- XSL Tools User Preferences Walkthrough - July 23, 2008

IP Issues

IP Checks

Component Lead has proof-read, double checked and confirmed the following:

- About files and use licenses are in place as per the Guidelines to Legal Documentation.
- All contributions (code, documentation, images, etc) has been committed by individuals who are either Members of the Foundation, or have signed the appropriate Committer Agreement. In either case, these are individuals who have signed, and are abiding by, the Eclipse IP Policy.
- All significant contributions have been reviewed by the Foundation's legal staff. Include references to the IPZilla numbers of all clearances.
- All non-Committer code contributions, including third-party libraries, have been documented in the release and reviewed by the Foundation's legal staff. Include references to the IPZilla numbers of all clearances.
- All Contribution Questionnaires have been completed
- The "provider" field of each feature is set to "Eclipse.org"
- The "copyright" field of each feature is set to the copyright owner (the Eclipse Foundation is rarely the copyright owner).
- Any third-party logos or trademarks included in the distribution (icons, help file logos, etc) have been licensed under the EPL.
- Any fonts or similar third-party images included in the distribution (e.g. in PDF or EPS files) have been licensed under the EPL.

IP Log and Documentation

Our IP Log is complete and has been reviewed by Eclipse Legal. See Appendix 1.

Committers in XSL Component moving to Source Editing Project

As voted on via mailing list, as per Move Review].

Name	Email	cvs id
David Carver	d_a_carver{at}yahoo.com	dacarver
Doug Satchwell	doug.satchwell{at}ymail.com	dsatchwel

Appendix 1

IP Log for XSL Incubating Components to graduate and move to Source Editing Project

This IP Log was generated dynamically from automatically collected project information and then hand edited to be specific to the XSL Component.

Third-Party Code

CQ	Third-Party Code	License	Use
1837	Webtools Incubtor XSL Tool	Eclipse Public License	
1904	xalan.jar Version: 2.7.0 (PB CQ1761)	Apache License, 2.0	unmodified binary
1932	BCEL.Jar Version: 5.2 (PB CQ88)	Apache License, 2.0	unmodified binary
1934	java_cup runtime Version: 10k (PB CQ1414)	Java Cup License (MIT Style)	unmodified binary
1985	xalan 2.7.1 top level jar Version: 2.7.1 (minus nested jar files)	Apache License, 2.0	unmodified source & binary
2142	XSL Tooling support for XIncludes and Generation Output	Eclipse Public License	
2232	xinclude schema Version: 1.0	W3C License	unmodified source & binary
2877	xalan 2.7.1 top level jar Version: 2.7.1 (minus nested jar files) (using Orbit CQ1985)	Apache License, 2.0	
2878	Xerces Version: 2.9.0 (excludes nested jar files) (using Orbit CQ2095)	Apache License, 2.0	
2879	Apache Commons Logging Version: 1.0.4 (using Orbit CQ1945)	Apache License, 2.0	
2971	PsycoPath XPath 2.0 Processor	Eclipse Public License	

No pre-req dependencies

Pending Contribution Questionnaires

These "pending" items are not required for release. We can incorporate them whenever they are able to have a complete review and do not need them for function (they are test related).

CQ	Third-Party Code	License	Use
3155	Junit W3C Tests for XPath 2.0	Eclipse Public License	
2975	W3C XQuery 1.0 Version: 1.0.2	W3C License	

Committers

Past and Present Active	
Name	**Organization**
David Carver	STAR Organization
Doug Satchwell	

Contributors and Their Contributions

Bug	Size	Description
Nik Matyushev (gmail.com)		
195262	3.3K	[xslt][editor]Need more XPath-centric double-click strategy in XML editor for attributes proposed patch (fist iteration)
Torsten Stolpmann (verit.de)		
217919	2.6K	[xslt][doc] Need ability to read xincludes and execute through ant Updated Packages

Works-with relationships

XSLT Engines

We provide one XSLT engine (Xalan 1.0) that works out of the box, but users can choose to install additional XSL engines, such as Saxon 2.0 if they desire. There's no requirement to install it, but naturally every runtime (and version) offers it's own special features.