

Proposal to Refactor WTP into Smaller Eclipse Projects

August 7, 2007, David Williams, WTP PMC

Eclipse Public Review is scheduled for August 15, 2007, 10 AM Eastern. See [Eclipse Projects](#).

Feel free to ask questions, comment, and vote! in Bugzilla: [bug 199112](#).

Background and the reason for this proposal

The top-level [Web Tools Platform \(WTP\) Project](#) is currently divided into two large projects, JST and WST, each of which cover a large span of functionality, plus three smaller, more focused projects JSF and Dali (JPT) and ATF (still incubating). Our original [WTP charter](#) covers the WST and JST projects, the others being added later via creation reviews and graduation reviews.

Over a year ago, we anticipated that WST and JST were too large to effectively manage themselves as a group of knowledgeable, voting committers, so we divided them up into “[Component Committer Teams](#)”. Each of these teams actually worked on a collection of 4 to 8 components, some in WST and some in JST.

This has been working fine for us and our teams, but recently we ran into complications when we started to formalize this organization for the “voting portal” that the Eclipse Foundation created. In brief, some of these “component teams” cross both projects (WST and JST) and that cross-project case isn't really covered by the Eclipse by-laws, development process documents, and administrative processes. For example, see the [Eclipse Development Process](#), [Official Documents Pertaining to Projects](#), and [Governance Documents](#). In other words, the reality didn't match what was documented. Our current organization could, technically, be handled by the existing processes, but that would mean, that if someone, for example, is voted into joining the “Web Services Component Team” they are basically being voted into both the WST and the JST projects at the same time. While technically correct, this does not make much conceptual sense, and simply further dilutes the meaning of the term 'Eclipse Project'.

The Proposal

To solve this mismatch of reality and documentation, I'd like to propose to do away with WST and JST as Eclipse Projects and replace them with new projects designed along the lines of the component committer teams we already have been using. See [Appendix 1](#) for complete details, but the main list of projects and project leads are listed in [Table 1. "Project Leads"](#).

1. Project Leads

Project	Lead
Common: tools and infrastructure not directly related to web tools, but required by web tools	Konstantin Komissarchik, BEA
Server Tools: tools and infrastructure to define and interact with servers.	Tim Deboer, IBM
Source Editors: xml, dtd, xsd (and sse infrastructure) html, css, javascript, jsp	Nitin Dahyabhai, IBM
Web Services: wsdl, axis1, axis2, web services framework, web services explorer	Kathy Chan, IBM
Java EE Tools: Common Project Infrastructure, jee models, preferences, classpath model, publish api, refactoring	Chuck Bridgham, IBM
EJB Tools: EJB creation wizards, preferences, future annotation tools	Naci Dai, ertation
JSF Tools: infrastructure and tools for JSF	Raghu Srinivasan, Oracle
Dali (JPA Tools): infrastructure and tools for JPA applications	Neil Hauge, Oracle
ATF (incubating): infrastructure and tools for AJAX	Robert Goodman, IBM
Datatools (RDB) [a near obsolete project, just doing 1.5 maintenance]	Der Ping Chou, IBM
WTP Release Engineering: contains the code and scripts to do builds, various tests, API scans, etc.	David Williams, IBM

Frequently asked questions

1. Does this mean the WST and JST distinction is not important?

No, it is important architecturally ... and that architectural distinction will remain ... but, it does not reflect the team makeup and working relationships which we believe are important to project domains and definitions.

2. Is this refactoring changing the charter of WTP?

No, at least, not the scope of the charter, same code, same goals, just the people will be organized differently.

3. Will each of these projects have a separate, individually delivered 'product'?

No. We will continue to base our packaging based on “customer demand”, combining whatever features and plugins are required.

4. Does this mean the people that have been the “component team leads” will have more work once they they are “project leads”?

No, they have already been responsible for reporting to the PMC, being the focal point for their components teams, making sure bugs are triaged, making sure planning input is supplied, schedules met, etc. There might be some new activities on the surface ... such as they might now get email directly from the EMO about some IP questions, priorities, etc, but in reality that's no change. In the past, if a PMC member or previous Project Lead got such an email, we'd often have to go the Components Team Lead anyway to get the information or details. We'll continue using that back-and-forth communication to accomplish what ever is needed, by who ever is best to do it.

5. Will there be changes made in CVS to reflect these new projects?

No, especially nothing right away. As the projects continue to evolve, we will keep these projects in mind in case there are opportunities to improve things in a meaningful, non-disruptive way.

6. What about bundle and package names? Does this mean there will be a big literal refactoring of all the code to change all the package or bundle names to remove 'wst' and 'jst'.

Definitely not! We would not do anything that disruptive to our adopters! All bundles and package names would stay the same, and new bundles and packages names would likely follow the 'wst' and 'jst' pattern since that would probably be least confusing. One exception to this might be the use of “org.eclipse.common” for some new API's in the Common Project, but that is more related to [bug 167144](#) than to this proposal.

7. If someone is already a committer in one of the WTP Projects, do they have to become a committer to contribute to one of the other WTP Projects.

Yes and No. Yes, they have to be voted-in to become a committer to that other project by the members of that other project. The process and paper work would be less, however, since they would already be a committer in WTP. But, also, no, since they can always contribute code via patches. These patches would not be subject to the “250-line-triggered-mandatory-IP-review” since they are already committers to the same top-level project.

8. Do Project Leads automatically become members of the PMC?

No. PMC members are nominated and voted in by other PMC members, based on their perceived potential to contribute to WTP beyond one project and to help manage all of WTP, not just one project.

9. In the future, for plugins in WTP to move from one project to another, will another refactoring review be needed?

No. Small, technically motivated movement between projects of the same top-level project, would be handled by simply discussion/announcements on open lists, and formally documenting the agreement of Project Leads and the PMC. The PMC would still be responsible for notifying the Eclipse Foundation. And, of course, if anything was substantial or changed the meaning of any project, then we would apply for an appropriate Eclipse review.

Actions to do after approval of this proposal

1. Eclipse Foundation to update their master data base of projects and committers to reflect the new structure and memberships.
2. Eclipse Foundation to update any code or data necessary to hook us into the Eclipse Voting Portal.
3. Eclipse Foundation to double check the committer paper-work to be sure all committer legal paper work is in order with the new structure. In theory, I've heard, it is possible some committers have very specific paper work that says, for example, they are "permitted to commit to JST Project only". We don't think that's the case, but we should check, and if there is a case found, then work with them to get that paper work updated with correct project name(s).
4. WTP to update their landing web site to reflect the new Project structure. Old, fringe documents will be updated if they are updated for other things during normal development, so some documents may remain out-of-date.

Projects and components

[Table 2., Descriptive Components](#), is provided just to provide a better overview description of the projects. What ever components we have in terms of Bugzilla, etc. will remain as they are. Some projects in Eclipse allocate voting rights to within components of their Projects. For a hypothetical example, in our Source Editors Project, we could hypothetically have a separate team/group of committers for “JSP Editors” vs. “XML Editors” as they are separate components. But, we do not see a need for that level of granularity at this time. That would only be necessary if these Projects (and their components) grew to contain many more committers than they currently have).

2. Descriptive Components

Project	Components
Common	Faceted Project Framework Validation Framework Command Framework Snippets View
Server Tools	Non-Java Servers Java Servers
Source Editors	XML: xml, dtd, xsd (and sse infrastructure) Web: html, css, javascript Java: jsp
Web Services	wsdl web services framework axis1 axis2 web services explorer
Java EE Tools	
EJB Tools	
JSF	
Dali (JPA Tools)	
ATF (incubating)	
Datatools (RDB)	
WTP Release Engineering	

Appendix 1: People and Code

The following lists the people who are committers in each project, as well as the code that is in that project. Where possible, a feature is listed, which means all plugins in that feature are part of the project, except for those other cases where plugins are listed in a project.

1. Common

People	Code
Konstantin Komissarchik John Lanuti Nitin Dahyabhai Craig Salter Peter Moogk Chuck Bridgham Jason Sholl Lawrence Mandel Kate Price David Williams Naci Dai	org.eclipse.wst.common_core.feature org.eclipse.wst.common_sdk.feature org.eclipse.wst.common_tests.feature org.eclipse.wst.common_ui.feature org.eclipse.wst.common_userdoc.feature org.eclipse.jst.common_core.feature org.eclipse.jst.common_sdk.feature

2. Server Tools

People	Code
Tim DeBoer Larry Isaacs Gorkem Ercan Rob Frost Kate Price David Williams Naci Dai	org.eclipse.wst.server_core.feature org.eclipse.wst.server_sdk.feature org.eclipse.wst.server_tests.feature org.eclipse.wst.server_ui.feature org.eclipse.wst.server_userdoc.feature org.eclipse.jst.server_adapters.feature org.eclipse.jst.server_core.feature org.eclipse.jst.server_sdk.feature org.eclipse.jst.server_tests.feature org.eclipse.jst.server_ui.feature org.eclipse.jst.server_userdoc.feature

3. Source Editors

People	Code
Nitin Dahyabhai	org.eclipse.wst.xml_core.feature
Amy Wu	org.eclipse.wst.xml_sdk.feature
Craig Salter	org.eclipse.wst.xml_tests.feature
Keith Chong	org.eclipse.wst.xml_ui.feature
Kate Price	org.eclipse.wst.xml_userdoc.feature
Naci Dai	org.eclipse.wst.css.core
Valentin Baci	org.eclipse.wst.html.core
Lawrence Mandel	org.eclipse.wst.javascript.core
David Williams	org.eclipse.wst.css.ui
	org.eclipse.wst.html.ui
	org.eclipse.wst.javascript.ui
	org.eclipse.jst.jsp.core
	org.eclipse.jst.jsp.ui

4. Web Services

People	Code
Kathy Chan	org.eclipse.wst.ws_core.feature
Chris Brealey	org.eclipse.wst.ws_sdk.feature
Peter Moogk	org.eclipse.wst.ws_tests.feature
Craig Salter	org.eclipse.wst.ws_ui.feature
Richard Mah	org.eclipse.wst.ws_userdoc.feature
David Lauzon	org.eclipse.jst.ws.*
Kate Price	
David Williams	
Naci Dai	
Valentin Baci	
Lawrence Mandel	
Lahiru Sandakith	

5. JSF

People	Code
Raghunathan Srinivasan	org.eclipse.jst.jsf.*
Cameron Bateman	org.eclipse.jst.pagedesigner.*
Gerry Kessler	
Ian Trimble	
Jean Choi	

6. Java EE Tools

People	Code
Chuck Bridgham	org.eclipse.jst.j2ee
John Lanuti	org.eclipse.jst.j2ee.core
Jason Sholl	org.eclipse.jst.j2ee.jca
Naci Dai	org.eclipse.jst.j2ee.jca.ui
Rob Frost	org.eclipse.jst.j2ee.navigator.ui
Kate Price	org.eclipse.jst.j2ee.ui
David Williams	org.eclipse.jst.j2ee.web
Kaloyan Raev	org.eclipse.jst.j2ee.webservice
Carl Anderson	org.eclipse.jst.j2ee.webservice.ui
Neil Hauge	org.eclipse.jst.jee
	org.eclipse.jst.jee.ui
	org.eclipse.jst.jee.web
	org.eclipse.jst.servlet.ui
	org.eclipse.wst.web.*
	org.eclipse.wst.doc.isv
	org.eclipse.wst.doc.user
	org.eclipse.wst.doc.user.feature
	org.eclipse.wst.webtools.doc.user

7. EJB Tools

People	Code
Naci Dai	org.eclipse.jst.j2ee.ejb
Chuck Bridgham	org.eclipse.jst.jee.ejb
Jason Sholl	
Sinan Konya	org.eclipse.jst.ejb.ui
Carl Anderson	org.eclipse.jst.j2ee.ejb.annotation.model
Rob Frost	org.eclipse.jst.j2ee.ejb.annotations.emitter
Kate Price	org.eclipse.jst.j2ee.ejb.annotations.ui
David Williams	org.eclipse.jst.j2ee.ejb.annotations.xdoclet
Kaloyan Raev	

8. Dali (JPA Tools)

People	Code
Neil Hauge Shaun Smith Brian Vosburgh Dirk Le Roux Karen Moore Max Rydahl Andersen Paul Fullbright Tran Le	org.eclipse.jpt.*

9. ATF (incubating)

People	Code
Robert Goodman Javier Pedemonte Kevin Sawicki Leugim Bustelo Philip Berkland Philippe Ombredanne	org.eclipse.atf.*

10. Datatools (RDB)

People	Code
Der Ping Chou Shiela Sholars Lawrence Dunnell Dirk le Roux Kate Price David Williams	org.eclipse.wst.rdb_adapters.feature org.eclipse.wst.rdb_core.feature org.eclipse.wst.rdb_sdk.feature org.eclipse.wst.rdb_tests.feature org.eclipse.wst.rdb_ui.feature org.eclipse.wst.rdb_userdoc.feature

11.WTP Release Engineering

People	Code
David Williams Naci Dai Tran Le Raghunathan Srinivasan John Lanuti	downloadsites org.eclipse.wtp.releng.tests.feature releng.control releng.wtpbuilder org.eclipse.wtp.releng.tests org.eclipse.wtp.releng.tools org.eclipse.wtp.releng.webupdatesite

Appendix 2: High level components that span WTP

This is to document our WTP policy about these specialized parts of WTP. These areas span all of WTP, both conceptually and in terms of cvs permissions.

Releng Maps

There is one part of release engineering that is governed as an Eclipse Project (since it has code contributions and responsibilities for deliverables to WTP). There is another part for maps files, where the locations and versions for various components for various builds are specified. All of these map file locations will automatically have read/write access to every committer in WTP. The current CVS directories this applies to is releng, releng.dali, releng-jsf, releng-atf.

Website

Every committer in a WTP Project will automatically be given read/write access to the CVS parts of the website. But, there are additional people that can have write/read access to only the website area (but not other project areas) since they've made enough contributions to be voted-in for that specific area of work. These votes would be WTP-wide. The current CVS area for our website is /cvsroot/org.eclipse/www/webtools. The current list of additional people with commit rights to only the website area is short: Jess Garms and Bob Fraser.

Bugzilla

Everyone in a WTP Project will automatically be given committer access to all parts of WTP's Bugzilla, so they can move and triage bugs. In addition, there can be additional people with access to only Bugzilla, to allow them to do triage, even if they do not have other commit rights (we do not have any at the moment, but this seems to sometimes occur in Eclipse Projects in general, so just wanted to document the policy here).