

Web Tools Platform (WTP) 3.2

for the Helios Simultaneous Release Review

Full Release Review Materials

June 11, 2010

Prepared by David Williams and sub-project leads

Table of Contents

Introduction and Purpose.....	2
WTP 3.2 Goals and Plans.....	4
Features.....	5
Common Tools.....	5
Server Tools.....	5
JavaScript Development Tools (JSDT).....	5
Source Editing.....	6
Web Service Tools.....	6
Java EE Tools.....	6
EJB Tools.....	7
JSF Tools.....	8
Dali.....	9
Non-Code Aspects.....	10
Developer and API Documentation.....	10
End-User Documentation and Examples.....	10
APIs.....	11
Architectural Issues.....	12
Tool Usability.....	13
End-of-Life.....	14
Quality (Bugzilla).....	15
Focused Quality Activities.....	15
Bugzilla statistics.....	15
Standards.....	16
W3C and OASIS standards.....	16
JCP standards.....	16
UI Usability.....	17
Schedule.....	18
Communities.....	19
Committers and Contributors.....	19
End-User Community.....	20
Adopter Community.....	20
IP Issues.....	21
IP Checks.....	21
IP Log and Documentation.....	21
JSDT Graduation.....	22
Helios Summary Simultaneous Release Compliance.....	24

Introduction and Purpose

This document is to fulfill the requirements of the [Eclipse Release Review](#)¹ for WTP 3.2 planned for release at the end of June, 2010.

Communication Channel: wtp-dev mailing list (See <https://dev.eclipse.org/mailman/listinfo>).

History

The Eclipse Web Tools Platform Project was originally proposed in 2004 by ObjectWeb, IBM and others. The Eclipse Foundation creation review was in June, 2004 with full time development since October, 2004. The original code contributions were from IBM and Eteration (“ObjectWeb Lomboz”). Since then several other large contributors have joined the effort, including SAS, Oracle and SAP, and others and several new subprojects formed, such as Dali and JSF.

Previous Releases

- WTP 0.7 July, 2005 and subsequent 0.7.1
- WTP 1.0 December 2005 and subsequent 1.0.1, 1.0.2, 1.0.3
- WTP 1.5 June, 2006 and subsequent 1.5.1, 1.5.2, 1.5.3, 1.5.4, 1.5.5
- WTP 2.0 June, 2007 and subsequent 2.0.1, 2.0.2
- WTP 3.0 June, 2008 and subsequent 3.0.1, 3.0.2, 3.0.3, 3.0.4, 3.0.5
- WTP 3.1 June, 2009 and subsequent 3.1.1, 3.1.2

Project Organization

Below is the list of current sub-projects and project leads as of June, 2010, that are part of this release.

Project	Lead
Common: tools and infrastructure not directly related to web tools, but required by Web Tools Platform	Carl Anderson, IBM
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, Java EE 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.	Angel Vera, IBM
JavaScript Development Tools (JSDT): Tools for editing and debugging JavaScript.	Nitin Dahyabhai, IBM
Source Editing: xml, dtd, xsd, xsl, sse infrastructure, html, css, jsp	Nitin Dahyabhai, IBM

¹ http://www.eclipse.org/projects/dev_process/release-review.php

Project	Lead
Web Services: Web services wizards and frameworks, Axis1 & Axis2 support, Web Services Explorer, WSDL Editor, JAX-WS support.	Keith Chong, IBM

PMC Organization

Our Project Management Committee, as of June, 2010, 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 3.2 Goals and Plans

Our Requirements Planning, Process, and Documentation is managed by Raghu Srinivasan, with substantial participation from each project lead.

We provide a standard-format [WTP Project Plan](#) which is updated every milestone as items are delivered or deferred, as part of our iterative development cycle.

Detailed requirements, plans, and progress are tracked (mostly) via Bugzilla, with 'plan' added as a keyword.

Themes and high-level requirements are coordinated through Eclipse Requirements Council and Planning Council representation.

Features

We document all our [new and noteworthy items](#)² for each milestone, for every release. The following summarizes the functionality provided by each sub-project with emphasis on what's new this release.

Common Tools

- **New this release: Deployment Assembly Framework**
A framework for Component Deployment that defines resource and archive dependency mappings. It also includes an extensible property sheet for specifying domain-specific deployment mappings.
- **Faceted Project Framework**
Provides a technique and UI for modeling server and runtime capabilities to associate with projects. This release moved some provisional API to be API, and usability improvements made in the UI for selecting facets for projects.
- **Validation Framework**
Provides preferences and APIs to provide domain specific validation for resources and projects. Provided official API this release, instead of only provisional.
- **Snippets View**
Offers a more visual mechanism for patterned text reuse through drag-and-drop.
- **Internet Cache**

Server Tools

- Support for Application Servers of several types (Java EE, HTTPD)
- Configure, publish, start/stop, debug
- Supports deploy, debug, project restart on Java EE runtimes
- Generic server adapter
- XML based configuration files for quick setup
- Includes JOnAS, WebLogic, WebSphere, Oracle Application Server, JBoss
- Custom (Java) server adapter for total control
- Includes Tomcat (included), Geronimo adapters (downloadable)

JavaScript Development Tools (JSDT)

- JavaScript editing
- Nature, project configuration and conversion tools
- Validation for JavaScript sources and Client-side JavaScript within web pages
- Integration with Common Navigator
- **New this release: JavaScript debug framework**

² <http://www.eclipse.org/webtools/development/news/main.php>

Source Editing

- Web Language Tools
 - HTML source editor
 - CSS source editor
 - JSP editor including syntax highlighting, integrated content assist for HTML, CSS, Java, JSF, EL, custom tags, and JSR-45 compliant debugging
- XML Language Tools
 - XML design and source editor
 - XSD editor - Graphical and source editor
 - DTD source editor
 - XSL source editor, XPath selection view, and XSLT debugging.
- Structured Source Editing (SSE) framework

Web Service Tools

New this release:

- JAX-WS support
- JAX-RS support

Continuing to support and improve:

- Extensible Web Service Wizards
- Creation of servlet based Web service using Axis1 and Axis2 runtimes
- Extension points for finding, creating and testing Web service
- Integrated into Java EE Navigator
- JAX-RPC codegen
- JSR 109 deployment ready
- Web services Ant tasks
- Web Services Explorer - UDDI, WSDL and WSIL pages
- WSDL Editor - graphical and source modes, integrated with XSD Editor
- WS-I Test Tools - Validate WSDL and SOAP for WS-I compliance
- Models for Web services deployment descriptors
- Service policies

Java EE Tools

New this release:

- Java EE 6 Support, including
 - Servlet 3.0 (web fragment) support
 - EJB 3.1
 - Connector 1.6

Continuing to support and improve:

- New wizards for creating Web artifacts: Servlet Filters and Application Lifecycle Listeners
- Usability enhancements in the toolbar of the Java EE perspective
- EAR 5 Bundled Libraries support
- Java EE Deployment Descriptor nodes in the Project Navigator view
- Improved Java EE classpath management/UI
- Ability to read Java EE models from binary archives
- Java EE Core Models and Model Provider Framework
- Natures and Builders
- Java EE Views and Navigators
- Java EE Projects and Modules
- Support for WAR, EJB-JAR, EAR, etc
- Java EE Navigator view
- Ability to target on different servers
- New wizard page in Java EE project wizards for configuring Java facets
- Better compliance with the UI guidelines for the Dynamic Web and EAR project wizards
- Improved default layout of the Java EE perspective
- Java EE preferences page introduced
- Avoid having the .JETEmitter system project in workspace by default
- Easier creation of new Servlets from existing Servlets and JSP pages
- Easier creation of new Filters from existing Servlets and JSP pages

EJB Tools

New this release:

- EJB 3.1 support.

Continuing to support and improve

- Wizards for creating EJB 3.0 artifacts: Session Beans and Message-Driven Beans

JSF Tools

New this release:

- JSF 2.0 Support (Java EE 6)

Continuing to support and improve:

- Web Page Editor
 - Multi-page Editor
 - Design View
 - Visual rendering of JSF, JSP and HTML tags
 - Support for JSF RI components
 - Support for Apache MyFaces Trinidad components
 - Extensibility framework to simplify adding support for other component libraries
 - Source View
 - Enhanced Source Editor gives content assists, both syntax and semantic validations, Hyperlink, Hover help
 - Preview Page
- JSF Validation
- Faces Configuration Model, Editor and Wizards
 - Multi-page Editor
 - Graphical diagram editor for navigational rules
 - EMF model of the application configuration resource file
- Support for JSF 1.1 and 1.2 versions

Dali

New and Improved this release

- Extensive JPA 2.0 support (Java EE 6)
- EclipseLink 2.x support
- JAXB Class and Schema generation

Continuing to support:

- Comprehensive, rich UI and validation for JPA metadata
- Provisional API for JPA model
- Extensible persistence.xml Editor
- Project Explorer Content for JPA projects
- New XML Mapping File wizard
- New Entity Wizard
- WTP (Facet) integration
- DTP Integration for DB metadata
- XML Mapping Descriptor editing support (JPA1.0 orm.xml)
- Code completion for JPA annotation values
- Annotated Java and ORM XML context based defaulting

Non-Code Aspects

Developer and API Documentation

Project-wide architectural overview (on website)

- Describes nature and relationship of components
- Presentations and tutorials provide drill down on selected topics (such as server definition)

Component overview (on website)

- Describes operation of an individual component and relationships among its parts, lifecycle issues, and other emergent properties of component

JavaDoc Package documentation

- Describes contents and inter-relationship of package contents

Per-file JavaDoc

- Conventional JavaDoc guidelines apply; scope is the class/interface being documented and its immediate surface area

Extension point documentation

- Provided with and as part of the API and JavaDoc documentation.

End-User Documentation and Examples

- Tutorials and presentation materials available on our [WTP Project Community website](http://www.eclipse.org/webtools)³. Some major additions were added this release and provided as re-usable education materials to teach web development.
- Contributed to conference presentations (EclipseCon, JavaOne, EclipseWorld), articles, etc.
- Two known books: Pro Eclipse JST, Eclipse Web Tools Platform.
- WTP downloads and installations includes end-user documentation. Additionally, this same documentation will be available on the internet, via an Eclipse info-center provided by the Eclipse Foundation.

³ <http://www.eclipse.org/webtools>

APIs

In general we provide APIs according to [Eclipse Quality API standards](#)⁴. But, we are aware that we (WTP) still do not provide enough APIs and still have too large a “provisional debt”.

We have published an [API Policy](#)⁵ document that describes how we protect some non-API so that adopters can invest with some assurance of continuity, but also detail the limits to that policy, so that eventually we can provide complete API.

4 http://www.eclipse.org/projects/dev_process/eclipse-quality.php

5 http://wiki.eclipse.org/WTP_API_Policy

Architectural Issues

- Tim Deboer manages WTP Architecture for this release, and has periodically held [work-group meetings](#)⁶ to discuss and decide specific architectural.
- We still have a few cases where we rely on non-API from the Eclipse Platform and JDT, this will be addressed in future releases.

6 http://wiki.eclipse.org/WTP_Architecture_Working_Group

Tool Usability

WTP provides a lot of functionality to web developers as indicated by the number of downloads and the traffic on the webtools newsgroup. It does emphasize Java based web development, even though it has some tools for pure HTML, CSS, XML and JavaScript development.

All WTP subprojects have “Ease of Use” in their plan for this release to address usability issues.

Several WTP subprojects made [UI walk-throughs](#) in front of the UI Best Practices Working Group. The feedback was useful for improving the usability in this release.

Another way we contribute to the overall usefulness experience to Eclipse end-users is to participate in the EPP packaging project, by “owning” the Eclipse IDE for Java EE Developers package. We have improved the usability of this package this release, by adding a custom welcome page which links to more information and samples.

End-of-Life

No end-of-life changes this release.

Quality (Bugzilla)

Neil Hauge manages the general, overall measurement and monitoring of our quality and bug handling.

Focused Quality Activities

In order to make sure we focused on quality of particular types, that are important to our project's health, we devoted part of each of our weekly status meetings to discussing and reducing our bug backlog in specific areas.

- First we focused on older bugs that had a severity of “Major” or higher to ensure that these potentially serious issues were being addressed, or at least correctly categorized.
- Next we focused on “bugs with patches attached” to make sure we were being responsive to non-committers efforts to improve parts of WTP that was important enough to them to contribute a patch.
- Finally we focused on reviewing newer bugs that had a severity of “Major” or higher to ensure that these potentially serious issues were being addressed, or at least correctly categorized.

Bugzilla statistics

The statistics in this table reflect overall Bugzilla entries (since project inception) with current snapshot, obtained on May 25, 2010 (4 weeks before release).

Total bugs	23939
Total Resolved/Closed bugs	0
Total Open bugs	0
Blocker/Critical	1
Major	136

The statistics in this table reflect activity since the previous release (for the period May 29th, 2009 to May 25, 2010)

Bugs opened	3434
Bugs resolved	3668
Fixed	2529
Invalid	145
Wontfix	244
Duplicate	377
Worksforme	344
Not Eclipse	29

We interpret these statistics as demonstrating an alive-and-well project: many bugs open, many resolved, approximately 70 per week. Our backlog decreased by 9% as a result of resolving more bugs than were entered over the past year. This is a nice trend that we would like to see continue into the next release.

Standards

W3C and OASIS standards

- HTML 4.01, XHTML 1.0 / 1.1, XML Catalog 1.0, CSS 2.0, ECMAScript 262
- XML 1.0, XSD 1.0, WSDL 1.1, WS-I Basic Profile 1.1
- SOAP 1.1, WS-I Attachment Profile 1.0.
- SOAP 1.2
- XML Catalog 1.1

JCP standards

- J2EE 1.2 / 1.3 / 1.4: Servlet, JSP, EJB, JAX-RPC, JSR045, JSR109, JSR921
- Java EE 5
- Java EE 6
- JDBC 2.1
- JSR 220: EJB 3.0 JSR
- JSR 317: JPA 2.0
- JSR 127: JSF 1.1
- JSR 252: JSF 1.2
- JSR 127: JSF 1.1
- JSR 252: JSF 1.2
- JSR 314: JSF 2.0
- JSR 318: EJB 3.1
- JSR 315: Servlet 3.0
- JSR 311: JAX-RS 1.1
- JSR 322: Connector 1.6
- JSR-224 JAX-WS
- JSR-181 Web Services Metadata for the Java Platform
- JSR-222 JAXB

UI Usability

We are familiar with, and follow, the [Eclipse User Interface Guidelines](#) .⁷

We do have some contributors that often open bugs for Accessibility and National Language issues so we can fix those bugs before end-users encounter them.

We participate in UI walk through, from time to time .

⁷ http://wiki.eclipse.org/index.php/User_Interface_Guidelines

Schedule

WTP followed the simultaneous release schedule, and delivered all milestones on time.

We plan to participate in simultaneous maintenance as well.

WTP may elect to deliver additional maintenance, tech preview milestones, or off-cycle releases in addition, if adopter or user requests warrant.

In addition to providing predictable milestones and releases, we also frequently “step up” to building and testing with our prerequisite software (usually weekly) so that we can find bugs early and get fixes in the platform and other prerequisite on behalf of the whole release train.

Communities

Committers and Contributors

Many active committers from several companies (including individuals)

- Committer elections and removals have followed charter principles
- Continuing to recruit additional contributors (organizations and individuals)
- Many, diverse, Contributing Organizations (current and previous): Eteration, Exadel, IBM, Innoopract, JBoss, ObjectWeb, Oracle, SAS, Thales, University of Karlsruhe, SAP, Sybase

Open communications via mailing lists and newsgroups

- Mailing lists: PMC, wtp-dev, wtp-releng, JSF, Dali,
- Meetings, meeting, meetings – numerous, open, and documented
- Weekly PMC, weekly dev status, requirements as needed, bi-weekly architecture
- PMC minutes available on website:
http://www.eclipse.org/webtools/development/index_pmc_call_notes.php
- Weekly status telecon minutes available on website:
http://wiki.eclipse.org/WTP_Development_Status_Meetings

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

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

Coordination/cooperation with other Eclipse projects

- Platform, JDT, DTP, TPTP, EMF, GEF, PDT

Leverage other open source technologies in the project

- Participate in Orbit

End-User Community

Substantial WTP download activity for milestones and releases

- WTP is one of the most popular downloads

Substantial website content

- Download links, New & Noteworthy, mailing lists, presentation DB
- Tutorials, documentation, presentation summary,

Evangelism and outreach in the market & broader community

- Website lists WTP events (conferences, etc.)
- Multiple commercial implementations
- Working with Eclipse and industry press to promote WTP
- Presence at EclipseCon, Eclipse World, others

Adopter Community

Many, known commercial adopters

- IBM – Rational Application Developer
- Oracle Corporation - Oracle Enterprise Pack For Eclipse (OEPE)
- Eteration – Lombos
- Genuitec – MyEclipse
- Innoopract – Yoxos
- Exadel – Exadel Studio
- Jboss – JBoss Developer Studio
- SAP – NetWeaver Studio

Many Server Adapters available, another sign of diversity:

Open Source

- Apache Tomcat
- Apache Geronimo
- JBoss
- Jetty
- ObjectWeb JONAS
- Glassfish

Commercial

- IBM WebSphere
- Pramati Server
- Oracle WebLogic Application Server
- SAP NetWeaver Application Server Java

IP Issues

IP Checks

Project Leads have all 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 most feature is set to "Eclipse Web Tools Platform" (Note: many were left as the previous "Eclipse.org" so as not to change too much unnecessarily, but will be corrected over time.
- The "copyright" field of each feature is set to 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

Submitted separately

- A list of third party software distributed with WTP, including information on the license and a link to the WTP CQ.
- The name of every committer for this release
- The name of every non-committer who contributed code via Bugzilla entries, with bug numbers.

Our IP Log contains a detailed description of our dependencies on third party software that is not re-distributed with WTP. In summary:

- Users can install their own Application Servers, where server adapters have been provided.
- EJB 2.1 developers can install Xdoclet for "old style" annotations
- Axis2 developers will need to install their own Axis2 runtime, if it is not part of their application server.
- JSF developers need to provide a JSF runtime and component libraries, if it is not part of their application server.
- JPA developers need to provide their own JPA runtime, if it is not part of their application server.

JSDT Graduation

This page is to fulfill the requirements of the [Eclipse Graduation Review](#) for WTP JSDT Graduation simultaneous with its June 2010 release. The JSDT Project is a special case, compared to many incubating projects, but is, technically, in the incubation phase, so this page is to document the reason it is ready for graduation, just so there is no doubt or ambiguity. Other information about JSDT, that is concerned solely with “release materials” are mixed in with the other sections of this Release Document.

It is a special case because the bulk of JSDT has existed for several releases, in the Source Editing project. One part that is “new” in WTP – the debugging framework – has existed for more than a year in the e4 incubating project. In addition to having established code, all the committers involved have been committers for many years at Eclipse, so they already “know the ropes”, follow “the Eclipse Way”, etc. See [Criteria for Graduating](#):

A working and extensible code base with exemplary tools

- JavaScript editing
- Nature, project configuration and conversion tools
- Validation for JavaScript sources and Client-side JavaScript within web pages
- Integration with Common Navigator
- New this release: JavaScript debug framework

Active communities

Committers and Contributors. There a number of active committers and contributors. While current committers are all from one company, there are many active contributors from other companies, so the project is diverse and expected to remain so and grow over time.

End-User Community. There are many active users, based on bugs that are open, and questions to mailing lists and newsgroups.

Adopter Community. There are several active adopters. Some with interest in extending the editor and integrating with IDEs, and some that are working on using the debug framework to debug other JavaScript runtimes (the exemplary implementation that ships with WTP is for Mozilla, but it is possible to extend the debug framework to debug JavaScript running in browsers).

Operating fully in the open using open source rules of engagement

The project effectively uses all the standard processes as the rest of WTP does; Bugzilla, Wiki, mailing lists, newsgroups.

The project "gets the Eclipse way".

In addition to adhering to standard Eclipse Development Process, JSDT participates in the larger Eclipse community. For example, tutorials and presentations at [EclipseCon](#). Interacting with other Eclipse Projects (such as PHP) to make sure compatible and “plays well with others”. Providing [new and noteworthy items](#) for each milestone.

Conclusion

In short, the project is a credit to Eclipse and is functioning well within the Eclipse community.

Helios Summary Simultaneous Release Compliance

- Project: webtools
- Date: 2010-05-28T13:24:42-04:00

Basic Requirements

- ◆ [Planning Url](#)
- ◆ [Ip Documentation IP URL](#)
- ◆ [Release Review PMC Executive Summary](#)

Common Repository Requirements

Communication And Availability

- ◆ Communication And Availability
- ◆ Communication And Availability Releng Contacts: David Williams

API

- ◆ Bugzilla Api Clean
- ◆ [Url Api Reports](#)
- ◆ [List Bugzilla Api Deviations](#)
- ◆ Message Bundles Used Throughout
- ◆ Standard Versioning Used
- ◆ Using Osgi Bundle Format
- ◆ Using Bundle Required Execution Environment
- ◆ Using Signing
- ◆ Used Jarred Bundles
- ◆ Used Reuse Orbit
- ◆ Used Optimize P2 Repositories
- ◆ Provide P2 Repository

Capabilities

- ◆ [Url Capabilities Description](#)
- ◆ [Url Discover Strategy](#)
- ◆ Used Support Translations
- ◆ Used ICU4J
- ◆ Used Branding
- ◆ No Unresolved Bugs
- ◆ Used Standard License Form

Eclipse Citizenship

Engage Community

- ◆ Engage Community
- ◆ [Link To Supporting Documentation](#)

Usability

- ◆ Following Usability Guidelines
- ◆ [Link To Supporting Documentation](#)

Performance

- ◆ Measurable Performance Criteria

◆Link To Supporting Documentation: unfortunately, we do not have criteria, and do no dedicated testing.

Test Localization

◆Use Babel Pseudo Translation

◆[Link To Supporting Documentation](#)

Enable Use With All Languages

◆Enable Use With All Languages

◆Link To Supporting Documentation: adopters do various enablement testing, and open bugs if problems found.

Builds

◆Builds

◆[Link To Supporting Documentation](#)

Ramp Down Planned And Defined

◆Ramp Down Planned And Defined

◆[Link To Supporting Documentation](#)

Accessibility

◆Accessibility

◆[Link To Supporting Documentation](#)

Unit Tests

◆Unit Tests

◆[Link To Supporting Documentation](#)

API Policy Defined And Documented

◆Api Policy Defined And Documented

◆[Link To Supporting Documentation](#)

Retention Policy

◆Retention Policy

◆[Link To Supporting Documentation](#)

Project Metrics

◆Project Metrics

◆Link To Supporting Documentation: 200 bundles, 20,000 unit tests, 2 million lines of code.

Key

- ◆Completed
- ◆No Data
- ◆Not Completed
- ◆Completed, with exception