



Eclipse Project 3.3 Release Review

Eclipse Project PMC

Highlights



- 3.3 new features:
 - Vista support, Java6 (JSR 269, JSR 199), UI usability/appearance enhancements, Early Access WPF port, Eclipse on the server, JNI launcher and customized splash, OSGi R4.1/JSR 291 support
- API quality:
 - High. 17 changes in porting guide.
 - Binary compatible for compliant plug-ins, 1 source incompatible change (JDT)
 - 129 new API (129) : Platform (64), JDT (50), PDE (5), Equinox (10)
 - 16 deprecated API: Platform (7), JDT (6), PDE (1), Equinox (2)
- End-of-life issues:
 - None
- IP Clearance and Licenses:
 - All licenses and about files are in place as per the Eclipse Development Process, the Due Diligence Process was followed for all contributions
- Community and Committer Diversity:
 - 91 committers, 74 active in past 9 months
 - Organizations: IBM (77), BEA (4), Consultants (3), Intel (2), Sun (1), Others (4)
 - Geographies: Canada (47), USA (22), Switzerland (10), France (6), Russia (2), Poland (1), Germany (1), Austria (1), UK (1).
 - Commits: IBM (94.15%), BEA (5.26%), Individuals (0.58%)
 - Consumed by all other Eclipse projects.

Themes and Plan Items



- **Components**
 - Target provisioning
 - Application Model
 - OSGi R4.1/JSR291 Spec
 - Server side support
 - Help content on the fly
 - User Assistance Content Componentization
- **Java**
 - Enhance launching support
 - Extend Clean Up
 - Compiler API (JSR199)
 - Support for Java6 debug
 - Pluggable annotation processing (JSR 269)
 - More refactorings
- **Consumability**
 - Remote help
 - Improve launching experience
 - Improve serviceability
 - Manage/share settings
 - Platform level proxy settings
 - GTK printing
 - Search based navigation
 - Performance focus
- **Vista**
 - Win32 version of SWT on Vista
 - Port SWT to WPF

Themes and Plan Items (cont'd)



- UI Evolution

- Background saving of editors
- Improve workbench usability
- Mozilla everywhere
- UI forms face-lift
- JFace enhancements
- Text editor productivity features
- Ubiquitous cheat sheets

- API

- Generalize editor annotation and ruler support
- Adopt new UI features
- Provide access to more native controls
- Custom widget API support
- Commands and key bindings
- JFace data binding
- Improve compare
- Graphics improvements
- Custom debugger integration
- Adopt the Eclipse File System

New and Noteworthy - Platform



- Eclipse on Vista
- Support for "Quick access" to UI functionality (ctrl-3)
- Undo support for resource, task and bookmark operations
- New default presentation with enhanced usability
- New Eclipse WPF port (early access)
- Improved workspace switching
- Improved resource selection dialog
- Working set usability improvements
- Hide-able window toolbar
- Working sets for the Project Explorer
- Patch wizard improvements
- Compare editor improvements
- Compare Structure in Outline view
- Long resource paths on Windows
- Open files with other editors
- Forms enhancements
- Keys preference page improvements
- Editor area drag and drop
- Properties view make over
- Console launcher for Windows
- Improved detection of Mozilla plug-ins
- Easier help navigation
- Help search term highlighting
- Categorized help search
- Print multiple help topics
- Cheat sheet improvements
- Import CVS projects directly into working sets

New and Noteworthy - Platform (cont'd)

- Export working sets as part of team project set
- History view search
- History view now has view history
- Spell check in Commit dialog
- Flat presentation in model-based synchronizations
- IDE application moved to new plug-in
- Advanced tooltips
- Custom tooltips for tree and table cells
- Contribute columns to vertical ruler
- Extensible hyperlink detection
- Remote help content support
- Plug in help in any format
- Improved dynamic user assistance content
- Added control of help/welcome content ordering
- Link to anything from context-sensitive help
- Compare supports File Buffers
- Printing support on GTK+
- New DateTime control
- Option to print line numbers added to StyledText
- Text SEARCH style
- Writing PNGs now supported
- Browser profiles on GTK+ and linux-motif
- Browser support added on PowerPC
- Native GTK Combo box
- New graphics line drawing capabilities
- SWT libraries automatically found
- Mozilla everywhere
- DragDetect API
- Drag over and drag under effects
- JavaXPCOM support
- OLE improvements and new OLEExample
- Reparenting support on Mac OSX
- Advanced graphics supports mirroring

New and Noteworthy – Platform (cont'd)

- Triple-click in text editors
- Double-click + mouse move to select words in editors
- Text drag and drop in text editors
- Scroll by pages using Ctrl + mouse wheel
- Convert tabs to spaces
- Smart Home/End
- Show invisible whitespace characters
- Recenter command
- Joining lines in text editors
- Resizable Find/Replace dialog
- Make hover sticky
- Configurable and extendable hyperlink detection
- Text editors restore caret position
- Spell checking in text editor
- Live annotate improvements
- Warning if editing a derived file
- Contribute columns to vertical ruler
- Run/Debug selection (or active editor) vs. launch last
- Run/Debug settings property page
 - manage launch configurations scoped by resource
- Support for mixed mode launching (e.g. profile and debug)
- Support for more than one launcher per application type and mode (e.g. two Java profilers in same workbench could co-exist)
- Console toolbar buttons to toggle console activation when output is written to stdout and stderr
- Support for pluggable detail panes in variable/register views
- JSch SSH2 Client Support
- Platform level proxy settings

New and Noteworthy - JDT



- New refactoring 'Introduce Parameter Object'
- New refactoring API for rename and move
- All refactorings are now scriptable
- Refactorings without save
- In-place rename refactoring
- More refactorings offered as quick assists: extract local, extract constants, inline locals, Convert local to field, Convert anonymous to member
- Colored labels in Java views
- Actions On Save: Organize Import, Format, Sort Members and Clean Up on Save
- Clean Up profiles
- Formatting, organize import sort members during clean up
- Working set improvements: Add new Java project to a working set, Assign projects to working sets
- More Java search options
- Workspace relative paths for Javadoc location
- New code formatter options: whitespace before operators, keep comments on first column, number of new lines between imports
- New compiler options: Don't report unused parameters when documented
- Deprecated and non-modifiable classpath variables
- Non-modifiable and disallowed attributes for classpath container entries
- Grouping of JARs and class folders in Package Explorer
- More undo support: undo delete and undo quick fixes
- Improved Javadoc comment checking
- Improved handling of duplicated fields and types
- Syntax coloring for brackets

New and Noteworthy - JDT (cont'd)



- Content Assist:
 - support for favorite static imports
 - proposes unresolved names
 - proposals on variables with unresolved type
 - improved completion on empty statement
 - improved inside catch clause
 - improved completion in annotations
- Perform additional optional actions on save
- Added template for adding JUnit 4 test methods
- Configurable Javadoc font
- Configurable background color for source hovers
- Class file editor shows disassembled code
- Shipping English spelling dictionaries
- No more "I don't see the full source" questions on the newsgroup
- J2SE-6 debugger support:
 - display object references
 - display all instances
 - force early return from methods
- Hyperlink stepping in debugger
- Access rules for Java system library API's when project bound to an execution environment
- Step through filters vs. step return from filtered locations
- Auto format for Java stack trace console
- Export logical structures (from user preferences)
- Support for Java 6 pluggable annotation processors (JSR269)
- Support for compiler API (JSR199)
- Improved handling of duplicate local variables
- Package/enclosing type pattern search
- Improved readable name of wildcard captures
- Filter method invoked via super types
- @Override for implementations of interface methods
- Declaration proposals for undefined local variables
- Warning for raw types activated by default
- Improved diagnostics for generic type casts
- Format by comment type
- Improved detection of unused private constructors
- Unused local types detection
- Improved null check detection (Null reference, Potential null reference, Redundant null check)
- Folders in JARs
- New code formatter options (wrapping binary expressions, exclude comments on first column from indentation)

New and Noteworthy - PDE



- New look for PDE editors
- Content assist in the plug-in manifest editor source pages
- Hyperlinking and Open Declarations in the plug-in manifest editor source pages
- Code folding and code formatting in the plug-in manifest editor
- Cheat sheet authoring
- Target Platform provisioning
- Self-hosting with multiple versions of the same plug-in
- Diagnostic tools in the Plug-in Dependencies view
- Stronger manifest file validation
- RCP tooling enhancements: Custom Splash screen templates and support to bundle JREs into an RCP product
- Enable the integration of additional OSGi frameworks in the PDE tooling
- New OSGi templates

New and Noteworthy - Equinox



- All plug-ins are now signed
- Custom splash screen support
- Native launching support
- OSGi R4.1 / JSR 291 reference implementation
- Adopted OSG MEG Application Model
- Support for Jetty embedded in Equinox
- Servlet Bridge support for running HTTP service in standard web app servers
- HTTP extension registry
- Help using embedded Jetty
- Improved PDE Build performance
- PDE Build against pre-built pieces downloaded from web
- Orbit (centralized bundling of third party libraries)



Deferred 3.3 Plan Items

- Components
 - API tools
 - Bundle/Module Development Tools
 - Incremental Plug-in Build
 - Improved support for provisioning
 - Bundle Management
- Consumability
 - Customization
 - Help keyword index
 - Ship Finer-grained Components
- Java
 - Add more class file targets for compiler
- Vista
 - Generalize the win32 version of SWT to win64
- UI Evolution
 - Improve multi-instance view management
 - Help webapp refresh

Performance of 3.3RC3 Relative to 3.2



<http://download.eclipse.org/downloads/drops/S-3.3RC3-200706011539/performance/performance.php>

3.3 Plug-in Changes from 3.2



Added Plug-ins (38)

- javax.servlet
- javax.servlet.jsp
- javax.servlet.jsp.source
- javax.servlet.source
- org.apache.ant.source
- org.apache.commons.el
- org.apache.commons.el.source
- org.apache.logging
- org.apache.logging.source
- org.apache.jasper
- org.apache.jasper.source
- org.apache.lucene.analysis
- org.apache.lucene.analysis.source
- org.apache.lucene.source
- org.eclipse.core.databinding
- org.eclipse.core.databinding.beans
- org.eclipse.core.net
- org.eclipse.cvs
- org.eclipse.cvs.source
- org.eclipse.equinox.app
- org.eclipse.equinox.http.jetty
- org.eclipse.equinox.http.registry
- org.eclipse.equinox.http.servlet

Added Plug-ins

- org.eclipse.equinox.jsp.jasper
- org.eclipse.equinox.jsp.jasper.registry
- org.eclipse.equinox.launcher
- org.eclipse.equinox.launcher.win32.win32.x86
- org.eclipse.jdt.apt.pluggable.core
- org.eclipse.jdt.compiler.apt
- org.eclipse.jdt.compiler.tool
- org.eclipse.jsch.core
- org.eclipse.jsch.ui
- org.eclipse.pde.ui.templates
- org.eclipse.ui.ide.application
- org.eclipse.ui.net
- org.eclipse.junit.source
- org.mortbay.jetty
- org.mortbay.jetty.source

Removed Plug-ins (0)

Unchanged Plug-ins (1)

- org.eclipse.core.boot

Non-Code Aspects



- The 3.3 release will contain updated User and ISV documentation
- Books authored by members of the Eclipse community:
 - Eclipse: Building Commercial-Quality Plug-ins (2nd Edition) (The Eclipse Series) (Paperback)
- Community is very active
 - Mailing lists and newsgroups have steady activity
 - Blogs dedicated to Eclipse are active e.g.
 - <http://www.planeteclipse.org>
 - Wiki content is growing
 - http://wiki.eclipse.org/index.php/Eclipse_Project

Non-Code Aspects



- **Internationalization**
 - Latin1 and Latin2 locales are supported in all operating environments
 - DBCS locales are supported on Windows, GTK, and Motif window systems
 - BIDI locales supported on Windows
 - GB18030-1 Chinese codepage standard is supported on Windows and GTK
- **Localization**
 - Tested for Localization (awaiting readiness of 'Eclipse Globalization Project')
- **Accessibility**
 - We are unaware of any non-compliance with section 508 accessibility standards in the user interface

Non-Code Aspects



- Articles, examples, and tutorials
 - New and updated articles and tutorials on eclipse.org (more than 5)
 - <http://www.eclipse.org/articles/Article-action-contribution/Contributing%20Actions%20to%20the%20Eclipse%20Workbench.html>
 - <http://www.eclipse.org/articles/article.php?file=Article-Unleashing-the-Power-of-Refactoring/index.html>
 - http://www.eclipse.org/articles/Article-JavaCodeManipulation_AST/index.html
 - <http://www.eclipse.org/articles/Article-CustomDrawingTableAndTreeItems/customDraw.htm>
 - <http://www.eclipse.org/articles/Article-Accessibility/accessibility.html>
 - Numerous Webinars and Podcasts
 - Some of the new/updated articles and tutorials were provided by the Eclipse community
 - Older articles need to be reviewed and updated for 3.3, if applicable

Platform Quality API



- API quality is a collaborative effort that involves the experience of the developers working on the Eclipse project, and feedback from consumers.
- API changes and proposed API additions are often broadcast to mailing lists to raise awareness of the changes and encourage discussion and feedback.
- The 3.3 migration guide identifies 17 changes:
 - http://dev.eclipse.org/viewcvs/index.cgi/org.eclipse.platform.doc.isv/porting/eclipse_3_3_porting_guide.html?view=co
 - For each, a description of the change, what code is affected, and the action that needs to be taken is described.
 - API changes between 3.2 and 3.3 are checked by component teams and checked again during API reviews, using an automated diff tool.
 - We are not aware of any API compliant plug-ins breaking as a result of these changes.
- The 3.3 migration guide also describes changes required to adopt mechanisms and APIs that are new in 3.3.
- The PMC is comfortable supporting the API that is in the Eclipse project 3.3

3.3 API – Platform



New

- New APIs have been added to the file buffer plug-in to deal with file stores.
- New extension points to for extendable hyperlink detection
- New extension point to contribute ruler columns to vertical ruler
- New API has been added to load a font from a file
- New API has been added to support line attributes
- New API has been added to draw selected text natively for StyledText and TextLayout
- New API has been added to support typed listeners (DragDetect, MenuDetect, MouseWheel)
- New API was added to support drag detect for custom controls
- New API was added to query the cursor that has been set in a control
- A DateTime widget was implemented
- New API was added to determine the text bounds of a tree or table item
- New API was added to support native search text controls
- New API to detect the hot state was added for custom draw trees and tables
- New API was added to specify that the Browser widget should be Mozilla, rather than the default
- New API was added to support cursor movement for the StyledText control and TextLayout
- New API was added to access JavaXPCOM from a Browser widget
- Verify support was added to CCombo
- Cut, copy paste was added to CCombo

New

- New API was added to support drag over and drop target effects in custom controls
- The built in Table and Tree drag effects became API
- New API was added to print line numbers in StyledText
- API to access Byte and Char types were added to OLE Variant
- New API to support the use of file buffers in compare editors
- New API to support validateEdit in compare editors
- New API to provide more flexibility for compare viewers (e.g navigation, asynchronous initialization, etc.)
- New API to support programmatic patch application
- Improved FileModificationValidator API to provide better support for Core level validateEdit calls
- New API to allow models to group related changes
- New History view API to improve view management and provide support for property change events
- New API to support the Flat presentation in the Synchronize view
- New API to provide support for comparing a single file in a compare editor
- New API to improve support for the local-history management of sub-file model elements (e.g Java Methods)
- New API to provide support for proxy setting management
- New API to provide support for using the JSch SSH2 client in Eclipse
- New file system API to support symbolic links and string attributes
- New API to associate a source id with a marker
- New IResource API to find the highest severity of the markers for a resource tree

3.3 API – Platform (cont'd)



New

- JFace Databindings Framework has been promoted to API
- Commands API expanded to support menus and improve handler support
- Support for column based viewers added to JFace.
- FieldAssist support expanded for image placement and colouring
- Workbench Image and Font support improvements
- Wizard page change notification expanded
- Resource undo support expanded
- FilteredResourcesSelectionDialog supplied as an API way to search for resources
- IURIEditorInput added to support EFS based editors
- EncodingFieldEditors expanded to allow for custom labelling
- FilteredItemsSelectionDialog added to support large element searches
- QuickAccess and ToggleCoolbar actions given API constants in ActionFactory
- FilteredTree opened up for greater customization
- IDragAndDropService service added to allow additions to drop support
- Intro added support for detection of new content
- MultiPageEditorPart expanded to allow searching and selection of editors
- Part property support added
- Progress supports setting of busy state and IJobRunnable
- IServiceScopes now supports lookup of a service based on context
- IFocusService now supports handler updates for focus changes
- IPersistableEditor allows for participation in the save/restore cycle of the workbench

New

- Filtering added to Saveable support to allow all Saveables in the Workbench to be saved
- Working sets added support for adaption of arbitrary elements into working sets
- NavigatorServices expanded to better support the CommonViewer sorter
- Extension Points
 - org.eclipse.ui.menus
 - org.eclipse.ui.preferenceTransfers
 - org.eclipse.ui.splashHandlers
 - org.eclipse.ui.statusHandlers

3.3 API – Platform (cont'd)



Deprecated

- Deprecated all file buffer API using IPath for performance reason, and replaced with more specific ones
- Deprecated the Platform Text's Assert class in favor of `org.eclipse.core.runtime.Assert`
- Old `IFileModificationValidator` API
- Team specific API for showing page based comparisons (replaced by flexibility added to Compare)
- `objectClass` attribute of `org.eclipse.ui.propertyPages`. Replaced by common expressions based the `enabledWhen` attribute.
- Resource API that did not reference the undo support in the copy and move actions in `org.eclipse.ui.actions` and resource dialogs in `org.eclipse.ui.dialogs`
- References to `ViewerSorter` deprecated in favour of `ViewerComparator` throughout the platform

3.3 API – JDT



New

- New API to access to new type wizard actions
- Added better API to access to the type hierarchy view
- New extension point:
`org.eclipse.jdt.ui.classpathAttributeConfiguration`
- New API to access all refactoring descriptors
- New API for rename and move refactorings
- New access to the refactoring script actions
- New API for adding filter actions to a search result view
- Enabled use of objects in 'iterate' and 'adapt' expressions by implementing `Iterable` and `ICountable`
- New API to allow expression references (reuse of expressions)
- New API to access the elements in the JUnit view
- New API for the JUnit launch delegate, launch shortcut and launch configuration tab
- Added test run listener API
- New API for access rule participants for execution environments
- New API to access reference and instance information from Java debugger
- New API for force return from methods in Java debugger
- Added methods on `org.eclipse.jdt.apr.core.util.AptConfig`, and new String constants in `org.eclipse.jdt.apr.core.util.AptPreferenceConstants`, to control Annotation processing while editing
- New search API to support patterns for package/enclosing type name while searching all types names
- Added to the batch compiler the ability to redefine the destination path on a source folder basis
- Added new flag on nature of searched element to specify both interfaces and annotations

New

- New API to query whether a method reference math is on a message sent from a super type of the declaring type
- Added API to create a Java element from an `IFile` using a specific project
- New API on search engine to return type handles when searching for all types
- New API to define the default behavior if a container initializer fails to initialize a container
- Added code assist API to get proposals that are required before a given proposal can be applied
- Added API to get the exceptions thrown by a method from a binding key
- Added API to validate names using specific source and compliance levels
- New formatter options to add blank lines between import groups and to insert a space before a parenthesized expression in throw clause
- New common interface for Java model's compilation unit and class file
- Added API to set the compiler's options relative to a given compiler's compliance
- Added API constants to represent the 1.7 compliance
- Added code assist API allows to get and set a list of favorite references
- New code assist API to get import statement proposals
- New API to flag a classpath variable as deprecated or read-only
- New API to generate text edits when sorting a compilation unit
- New extension point to register an annotation processor manager inside the Java Builder

3.3 API – JDT (cont'd)



New

- Added formatter options to clear blank lines in Javadoc comment, in block comment
- Added formatter options to format line comment, block comment, Javadoc comments
- New API to retrieve the bytes from a Java model's class file
- Added compiler options to report null dereference, report potential null dereference, report redundant null check
- New API interface to represent non-Java resources (files and folders) in a .jar file
- New API to retrieve the resources generated by the Java builder
- Added API to specify the problem requestor on a working copy owner
- Added API to handle incomplete DOM bindings
- New API to retrieve whether resolved bindings, statement recovery, or binding recovery was requested
- New API on classpath container initializer to specify whether the initializer supports a given attribute, or allow the modification of a given attribute
- Added formatter option to wrap before the binary operator
- Added formatter option to indent block comments or line comments that start on the first column
- New API on tool factory to specify whether the created formatter is formatting new or existing code

Deprecated

- Some PreferenceConstants (pushed down to Platform Text)
- Some problem ids (IProblem) that are no longer generated have been deprecated
- The formatter option that clears blank lines has been split into 2 options (clear blank lines in block comments and clear blank lines in Javadoc comments). See DefaultCodeFormatterConstants.
- The factory methods IClassFile#becomeWorkingCopy(...) to create a working copy with a problem requestor have been deprecated. One should define the problem requestor on the working copy owner instead.
- The Java convention methods that validate a name have been deprecated. One should use the version that take a source level and a compliance level instead. See JavaConventions.
- The following method on SearchEngine has been deprecated: searchAllTypeNames(char[], char[], int, int, IJavaSearchScope, TypeNameRequestor, int, IProgressMonitor)
 - One should use the version that take match rules instead.

Breaking changes

- - IClassFile#getType() no longer throws a JavaModelException. This only affects source compatibility.

3.3 API – PDE



New

- API for launch shortcuts (Eclipse Application, JUnit Plug-in, OSGi launchers)
- API for the JUnit Plug-in launcher delegate.
- API to access the plug-in model and the target environment settings
- New extension point
`org.eclipse.pde.ui.osgiFrameworks`
- New extension point
`org.eclipse.pde.ui.targetProvisioners`

Deprecated

- Upon creating the OSGi Framework launcher, `org.eclipse.pde.ui.launcher.EquinoxPluginsTab` and `org.eclipse.pde.ui.launcher.EquinoxSettingsTab` got deprecated. They got replaced by `org.eclipse.pde.ui.launcher.BundlesTab` and `org.eclipse.pde.ui.launcher.OSGiSettingsTab`

3.3 API – Equinox



New

- Added hook method to Jobs for cancelation
- Added SubMonitor progress monitor
- Query capabilities to the AdapterManager
- Added constants for WPF
- Enhanced bundle state manipulation API
- Added support of OSGi R4.1 and JSR 291
- Enhanced FileLocator.findEntries support
- Added convenience constructors to Status
- API to configure Job worker threads to be daemons
- Added StartupMonitor for interactive splash support

Deprecated

- Deprecated APIPlatform.getJobManager()
- Deprecated IPlatformRunnable

Tool Usability



- Eclipse is a superior IDE for Java tooling and plug-in development
- Many usability enhancements made in 3.3 to continue this tradition
 - New Presentation for enhanced usability
 - Quick Access (Control-3)
 - Remote help
 - Enhanced launching
 - Improve serviceability
(http://wiki.eclipse.org/index.php/Status_Handling_Best_Practices)
 - Better managing and sharing settings
 - Search based navigation
 - Save editors in background
 - Improved UI usability
 - Improved text editors productivity features
 - Performance
 - Much more...

Awards



Awards

- **2007 Java Magazin (German)**
 - Best Java Open Source Project, 1. Place 'Eclipse'
- **2006 Java Magazin (German)**
 - Best Java Open Source Project, 1. Place 'Eclipse'
 - Best IDE, 1. Place 'Eclipse - pure'
- **2006 Java Pro Magazine Readers' Choice Award**
 - Best IDE and Best Java Development Suite
- **2006 JAX Reader's Choice Award**
 - Eclipse awarded Best Open-Source Java Project
- **2005 LinuxQuestions.org Members Choice Awards**
 - Eclipse awarded IDE of the Year
- **2005 SOA Web Services Journal Readers' Choice Awards**
 - Eclipse awarded Best GUI for SOA
- **2005 Java Developer's Journal Readers' Choice Awards**
 - Eclipse awarded Best Java Application
 - Eclipse Rich Client Platform awarded Best Rich Client Platform
 - Eclipse IDE awarded Best Team Development Tool
 - Eclipse IDE awarded Most Innovative Java Product
 - SWT awarded Best Java Class Library
 - Eclipse awarded Best Java Debugging Tool

Architectural Issues



- New Eclipse Launcher starts Java directly instead of invoking Java executable
- Jars are signed
- Primary runtime is a 1.4 JRE. Complementary functionalities on 5.0 JRE (junit4, APT 5) and 6.0 JRE (APT 6, compiler API)
- Third party library bundles are now consumed from Orbit
- 38 new plug-ins

End of Life Issues



- When evolving API the Eclipse Platform will, whenever possible, deprecate the affected API methods and continue to keep them operational.
- Exceptions to this rule are in the 3.3 migration guide.

Bugzilla



- Between June 29, 2006 and June 9, 2007
 - More than 14,700 reports were created
 - Over 12,800 were resolved
 - Over 5,600 were resolved without changing code
 - invalid, duplicate, worksforme, etc...
- Current state is
 - 23 blockers, 90 critical
 - 0 P1, 136 P2
- 3.2 state was
 - 11 blockers, 80 critical
 - 0 P1, 158 P2

Bug resolution during 3.3

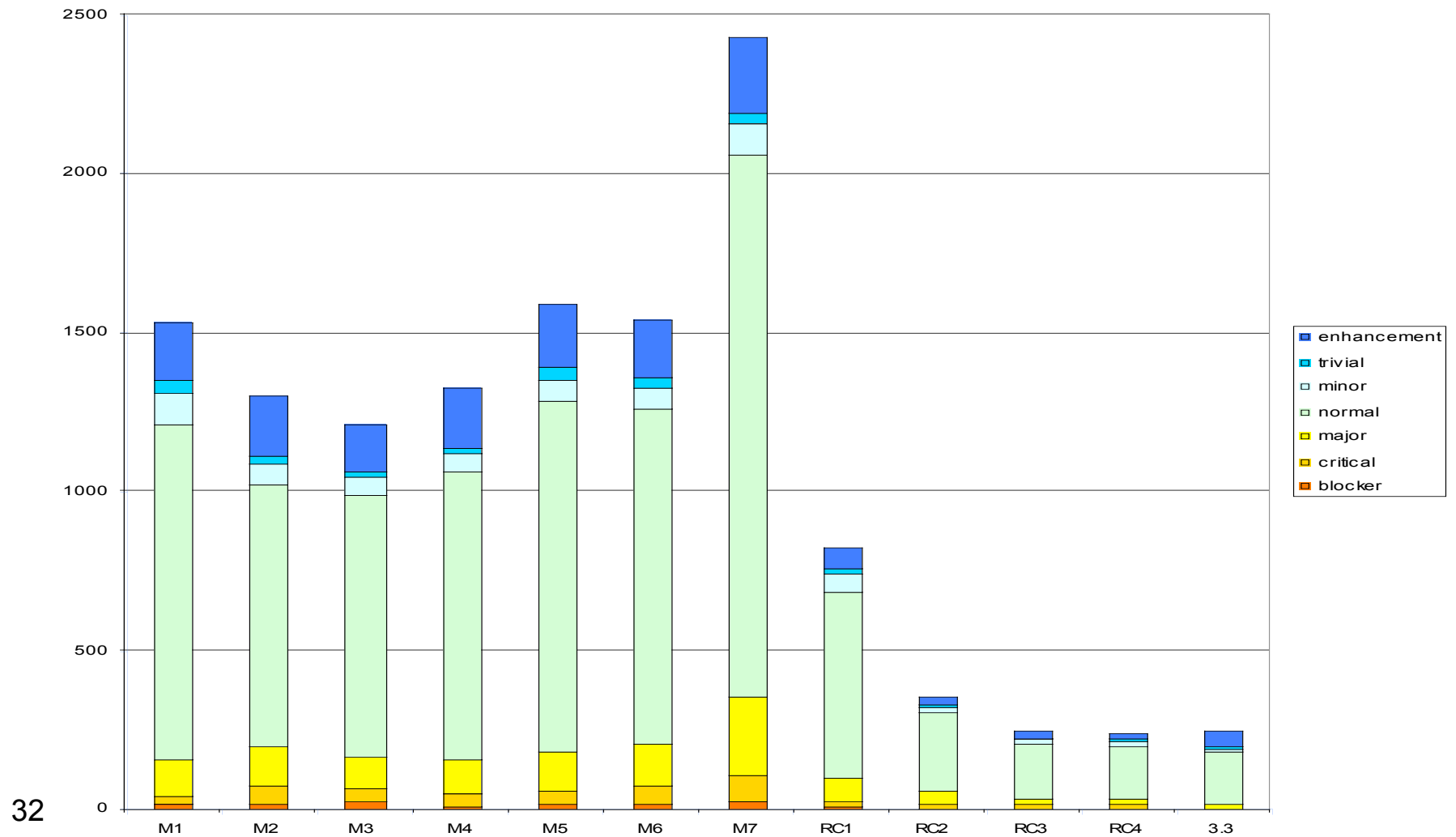


RESOLVED	M1	M2	M3	M4	M5	M6	M7	RC1	RC2	RC3	RC4	3.3	Total
blocker	21	19	29	12	18	22	29	9	5	5	4	0	173
critical	24	55	39	37	45	51	82	15	13	11	14	0	386
major	116	126	96	106	120	131	245	81	44	23	16	19	1123
normal	1052	822	825	911	1101	1057	1703	583	245	171	168	167	8805
minor	96	64	56	59	73	71	103	53	15	14	16	9	629
trivial	41	24	16	17	36	30	34	22	11	4	4	2	241
enhancement	186	190	152	184	201	180	239	60	26	19	17	56	1510
Total	1536	1300	1213	1326	1594	1542	2435	823	359	247	239	253	12867

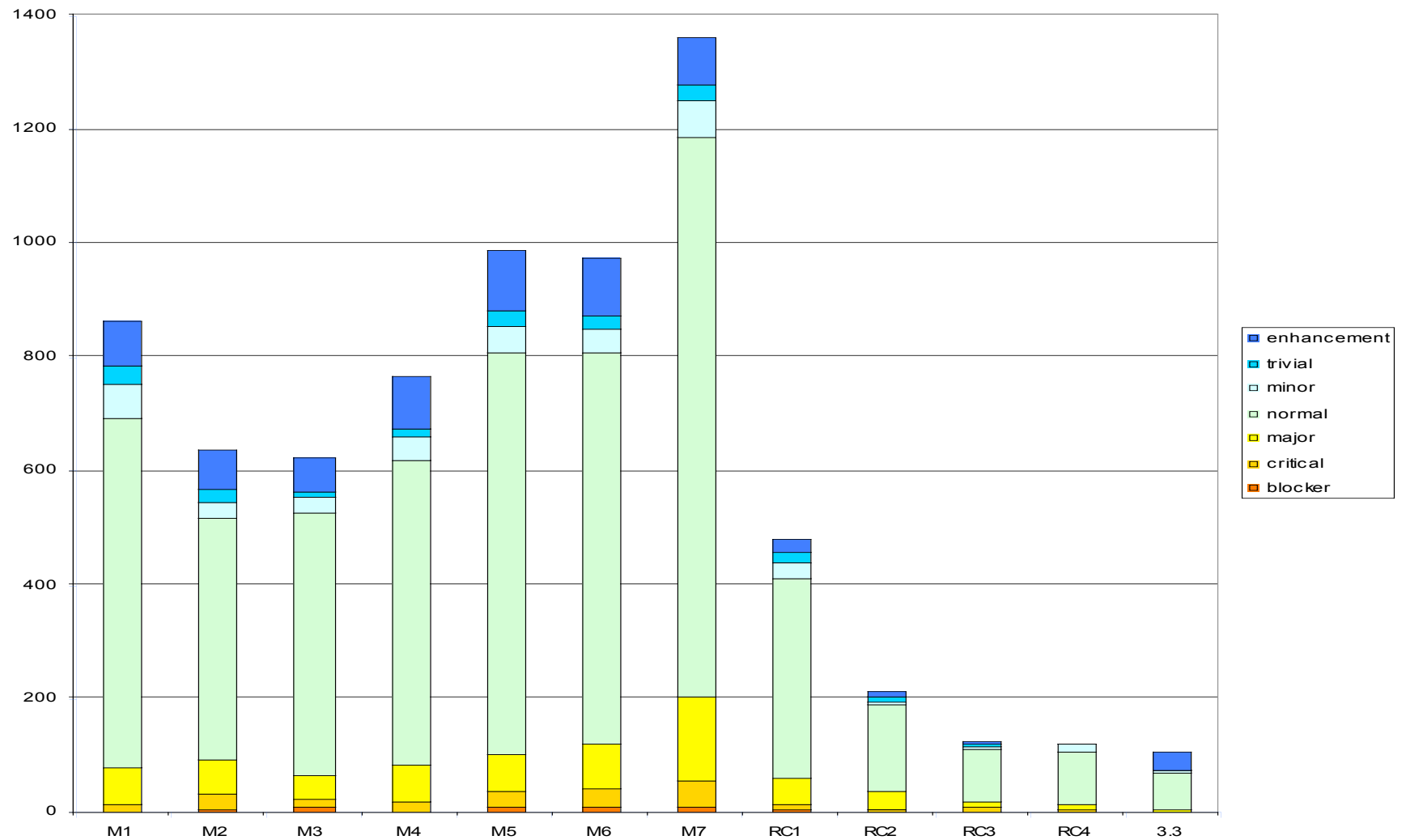
FIXED	M1	M2	M3	M4	M5	M6	M7	RC1	RC2	RC3	RC4	3.3	Total
blocker	3	4	9	3	9	11	10	6	2	2	1	0	60
critical	10	29	15	15	27	30	47	7	5	7	7	0	199
major	65	61	41	64	67	78	145	48	30	12	8	5	624
normal	613	425	460	539	704	691	984	350	151	90	93	66	5166
minor	62	28	29	41	46	41	65	29	8	6	10	3	368
trivial	31	20	11	12	30	24	27	18	10	4	4	1	192
enhancement	79	71	61	95	104	100	84	21	6	3	0	33	657
Total	863	638	626	769	987	975	1362	479	212	124	123	108	7266

Resolved bugs

including fixed, invalid, ...



Fixed bugs (only)



Standards



- OSGi
 - Service Platform Core Specification, Release 4
 - Elements of the OSGi Service Platform Service Compendium, Release 4
- Annotation Processing APIs
 - com.sun.mirror 1.5
 - javax.annotation.processing 1.6
- Java compiler API
 - javax.tools 1.6
- User Assistance consumes (pareses) a small subset of RSS 1.0 to get news from eclipse.org
- JUnit3 and JUnit4
- J2SE
 - Tools are build against J2SE 1.4
 - Compiler can generate 1.3, 1.4, 5.0, and 6.0 code
 - Clients can run 1.4, 1.5 or 1.6.

UI Usability



- Strings are externalized to support translation into other languages.
- Extensive use of mnemonics and shortcut keys in the user interface enhances usability.
- Full Bidirectional support (mirroring) on Windows.
- We are unaware of any non-compliance with accessibility standards in the user interface.

Schedule



- Milestones every 6 weeks, 6 cycle duration
 - Feature and API frozen on March 23, end of M6 cycle
 - http://www.eclipse.org/eclipse/development/eclipse_project_plan_3_3.html#Milestones
- Tracked schedule
 - All milestones except M5 delivered as promised
 - M5eh was produced to ease transition of clients who were broken by a contract change in newly deprecated FormColors.
- End game (release candidate) milestones for 4 cycles
 - Duration reduced from 2-week to 1-week cycles at RC2 milestone
 - No new features or API allowed without proper approvals
 - Development to end on June 8, 2006
 - Increasingly stringent approval, checking, and change notification requirements in this stage
 - http://www.eclipse.org/eclipse/development/freeze_plan_3.3.html

Process



- The Eclipse project is developed using an open, transparent, and inclusive process
- Teams rely on Bugzilla, mailing lists and newsgroups for input
- Weekly planning calls conducted with the PMC and component leads
 - Meeting minutes posted to the eclipse-dev mailing list
- Component teams have publicly available milestone plans
 - Use project's web space on eclipse.org to broadcast component milestone plan items and provide status on each item, per milestone

Community



- Eclipse team members are active in Bugzilla, newsgroups, and mailing lists
- Blogs started by Eclipse committers are evolving
 - New blogging infrastructure at Eclipse.org
 - <http://www.planeteclipse.org>
- Some teams are using the eclipse-dev IRC channel
 - `irc.freenode.net#eclipse-dev`
- The Eclipse team participates in code camps, conference presentations, and tutorials, including
 - EclipseCon, JavaOne, JavaWorld, JAOO
- The Eclipse team interacts with other open source projects, standards bodies, and other projects on eclipse.org, including
 - OSGi, Apache Ant, JLS, WTP, Apache Harmony, GCJ, GTK

IP Issues



- All significant and third party contributions have been reviewed and approved by Eclipse legal.
- About files and license files are complete and correct.
- Project log complete, yet to be reviewed by Eclipse legal.
 - http://www.eclipse.org/eclipse/development/eclipse_project_log.html

Project Plan for Eclipse 3.4



- Pending - still in planning stage
- Preliminary priorities:
 - Completeness and Consistency
 - Startup/footprint
 - Performance
 - Security
 - Globalization
 - Accessibility
 - Reliability
 - Consumability
 - Management