Eclipse Project 3.4 Release Review

Eclipse Project PMC

Highlights

- 3.4 new features:
  - SWT 64-bit,Linux BiDi, Mac Carbon Internalization & accessibility, product level configurability, provisioning (p2), serviceability, API tooling, security, concurrent compiler and Eclipse 4.0 planning
- API quality:
  - High. 7 changes in porting guide.
  - Binary compatible for compliant plug-ins
  - 164 new API: Platform (105), JDT (50), PDE (5), Equinox (4)
  - 18 deprecated API: Platform (13), JDT (5)
  - 7 breaking changes: Platform (4), JDT (3)
- End-of-life issues:
  - org.eclipse.apache : was never an API
- 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:
  - 205 committers, 79 active in past 9 months
  - Organizations: IBM (62), Individuals (9), Prosyst Soft.(2), Embarcadero Tech. (1), Adobe (1), Code 9 (1), QNX Soft. (1), compeople AG (1), Wind River (1)
  - Geographies: Canada (40), USA (20), France (6), Switzerland (5), Poland (2), Bulgaria (2), Germany 2), Austria (1), Japan (1).
  - Commits: IBM (96.85%), Individuals (1.95%), Embarcadero Tech. (0.80%), Prosyst Soft. (0.19%), QNX Soft. (0.11%), compeople AG (0.10%)
  - Consumed by all other Eclipse projects

Themes and Plan Items

- Platforms
  - Port SWT win32 to 64-bit
  - Support BIDI on Linux GTK
  - Provide full internationalization on Mac OS X
  - Implement accessibility for Mac OS X
  - Exploit the capabilites of modern JREs
- Consumability
  - Improve performance of large, Eclipse-based products
  - Serviceability
  - Provide additional product level configurability
  - Provisioning (p2)
- Reliability
  - Provide API for missing/internal features
  - Focus on architectural integrity
  - API Tooling
  - Provide commonly requested Java security features
- The Future
  - Create the Eclipse 4.0 plan
  - OSGi standards participation
  - Investigate new user-interface directions
  - Investigate the next generation of JDT capabilities

http://www.eclipse.org/eclipse/development/eclipse_project_plan_3_4.html

New and Noteworthy - Platform

- Improved regular expressions in Find/Replace dialog
- Various printing improvements for textual editors
- Line support in overview ruler
- Configure annotation preferences via overview ruler
- New annotation text styles
- Improved key binding support for content assist and quick fix
- Multiple hyperlink presenter
- Direct interaction with text hovers aka rich hovers
- Retain case of match when replacing in textual editors
- Select text via Shift+Click into line number ruler
- Resource actions in IDE now support LTK-style refactoring
- Line matches shown in text search results
- Line numbers for text search results
- Improved text search replace dialog
- Improved support for regular expressions in text search
- Retain case in text search replace
- The console encoding used when running or debugging a program automatically matches the encoding of the program being run or debugged.
- Watch expressions can be created by dragging text into the Expressions view. As well, expressions can be reordered with drag and drop.
- Import/export of launch configurations
- Enhanced contextual launch (i.e. click run or debug for active selection/editor) to support non-resource based selections/editors.
- Modules view has been pushed down to the platform from CDT
- An example debugger has been added to the Eclipse examples.
- Team preference page now allows to specify ignore path patterns as well as file name patterns
- Sharing multiple projects simultaneously
- File changes can be viewed in CVS Commit wizard
- Apply Patch wizard offers now better support during applying patches (more reliable fuzz factor)
- Line counter in Apply Patch wizard
- Import Team Project Set wizard allows users to fully configure the repository location, enabling them to specify a user name and change the connection method if required
- The new system proxy configuration option allows to reuse OS proxy settings in Eclipse
- SWT and Eclipse for 64-bit Windows
- SWT and Eclipse for HP-UX on IA64 (32-bit mode)
- SWT for Windows 64-bit Itanium Edition
- Accessibility for Mac OS X
- SWT RTL support for Linux GTK
- Allows the setting of an alpha value on a shell
- Allows the setting of a Region into a Control, which allows for the creation of non-rectangular controls
- Shell full-screen state
- SWT inline input method
- FileDialog overwrite prompt
- No scrollbars style for Table and Tree

New and Noteworthy - Platform (cont'd)

- Print control to GC
- ProgressBar now supports native Windows Vista styles (ERROR, PAUSED)
- Tristate check Buttons
- API that allows for shearing of images
- Create a region from a path
- Image transfer support
- URL transfer support
- RowLayout center alignment
- TextLayout and StyledText now allow different strikeout colors, underline colors, and different underline styles
- Proxy settings on Mozilla-based Browsers
- New customizable marker support
- New service infrastructure
- Expression-based Activities
- New customization preferences specify CSS to be included in each help page.
- Help content can be contributed by more than one remote infocenter
- The keyword index memory usage has been reduced by about 50%.
- Help table of contents is synchronized with the currently displayed topic.
- New intro preferences allow use of a user contributed start page which still using Universal Intro.
- A new feature org.eclipse.help simplifies creating RCP applications which use the help system.
- A number of security fixes have been made in the Web Application.
- Relevance of search results has been improved by giving more weight to matches in the title

New and Noteworthy - JDT

- Unused declared thrown exception optionally ignored if documented or if Exception or Throwable
- CLDC 1.1 class file generation
- Improved unnecessary code detection
- Java 5 annotations supported by Java model
- Fine-grained Java search
- Improved unnecessary @SuppressWarnings diagnosis
- Diagnosis of redundant superinterfaces
- Support for external class folders on build path
- Improved resilience with missing types
- Batch compiler using the Class-Path clause of JAR file manifests to complete the classpath
- Java compiler on multi-CPU machines
- Flexible ZIP archive extensions
- Comments processed by standalone code formatter
- Rearrange content of files per drag and drop
- Different colors for read and write occurrences in Java editor and Search view
- Breadcrumb in Java editor
- New Clean Ups:
  - correct indentation (also available as save action)
  - add unimplemented methods
- Improved cursor jumping on Quick Fix/Quick Assist
- More quick assists and fixes:
  - Create getter and setters for fields (encapsulate fields)
  - Extract method
  - Add solutions for unresolvable types
  - Use 'StringBuilder' for string concatenation
  - Use 'MessageFormat' for string concatenation
  - Extract local variable (without replacing all occurrences)
- Colored labels proposal popup
- Refactoring:
  - Clients can contribute participants to the change method signature refactoring
  - Refactoring flags references in binaries
  - New Extract Class refactoring
- SWT templates
- Extended template variables
- Specify location/context for templates
- Allow extenders to add pages to Javadoc export wizard
- Runnable JAR export wizard which can also generate an Ant script
- Improved recovery of missing types in APT mirror types
- JUnit view shows execution time
- Paste creates snippet for members and statements
- Formatting any Java element can be undone
- Content Assist :
  - code completion helps with casts
  - complete static members of not yet imported types
  - inserting parameter name also shows guessed arguments
  - uses colored label in proposal popup
- Allow to only format edited lines on save
- Javadoc hover and Javadoc view show constant values
- Rich Java problem hover
- Call Hierarchy view works with more members

New and Noteworthy - JDT (cont’d)

- Improved Javadoc view
  - current input is now shown with icon
  - allow to go back and forth in the history
  - enable/disable link with selection
  - can open the current input in an external browser
- Toggle Comment in Properties File editor
- Execution environment description files can be used to define JVM properties such as boot path, endorsed directories, language level, executables, source attachments, debug attributes, and so on, such that JDT can launch those JVMs. This allows for multiple "views" of one physical VM installation, and allows JDT to launch non-standard JREs.
- Added default suspend policy for newly created watchpoints (access and/or modification).
- Support for installed JRE search on Mac OS.
- A socket listening connector has been added to the Remote Java Application launch type. This allows one to start the Java debugger listening for connection on a specific socket.
- Launch configurations now appear when searching for references to a main type. Double-clicking on the search result opens the launch dialog on that configuration.
- Hovers for variables have been enhanced to display an object inspector. The inspector will display logical structures according to the toggle setting in the visible Variables or Expressions view.

New and Noteworthy - Equinox

- A replacement for update manager
  - New UI, simplified workflows
  - Manage Eclipse and more (exe, ini, bundles, registry keys, …)
  - Share bundles across Eclipse-based products
  - An installer
- A provisioning solution for OSGi systems
  - Manage non-running instance
  - Start level, framework extension
  - Fine-grained dependency management
- A provisioning framework
  - Pluggable and modular infrastructure
  - Provision more than Eclipse-based systems
  - No UI required
- Simple update workflow
  - Replace multi-steps wizards
- New metaphors
  - Drag n Drop for install, adding repos, …
- More flexible repositories
  - Connect to p2 repos, Update Sites, OBR, Maven, …
- Managed folders
  - Explicit “watched” locations
  - Drop content to have it installed
  - No need to unzip and –clean
- Installer
  - A subset of p2 bundles: ~5M
  - Advantages of using an installer:
    - Bundle pooling
    - Pack200
    - Simplicity

- Equinox Transforms
  - General Framework for bundle customization
  - XSLT to transform any XML file
- New OSGi Service Implementations
  - Declarative Services
  - Configuration Admin
  - Initial Provisioning
  - IO Connector Service
  - Wire Admin
- Secure Storage
  - Storage for sensitive data
  - OS integration for Windows and MAC
- Authentication with JAAS
  - Contribute services
  - Manage configuration
  - Observe lifecycle with events
- Trusted bundles
  - As bundles are loaded, perform authorization based on trust in signer

New and Noteworthy - PDE

- API tooling - integrated analysis of binary compatibility (against a previous release), missing/invalid @since tags, API leaks, and bundle version errors
- API tooling Ant tasks that can be leveraged by a release engineering build process to create API problem reports
- Support to compile runtime workspace plug-ins against the launching workspace
- Support to mark an extension point as internal. This allows extension point clients to be notified when they are using an internal extension point.
- A new p2 target provisioner that lets you add plug-ins to your target platform from a p2 repository or update site
- The Plug-in Registry view was enhanced to include information about OSGi services
- The product editor was enhanced to allow for per-platform configuration files (config.ini) and the ability to version product definitions.
- Support to identify IDs in extension point schemas such that other schemas can refer to them and code assist is provided for them
- An "Open Plug-in Artifact" dialog has been added to quick find extension references, extension point declarations, plug-ins and exported packages
- PDE, in conjunction with Equinox, now supports development against other OSGi framework implementations
- A Context Help editor to assist clients creating context-specific help for their applications
- A Table of Contents editor to help create, modify and visualize tables of contents for Eclipse products
- The Error Log view has been refactored into a separate plug-in, can now be used to quickly browse a log from a runtime workbench, and entries can be grouped by session
- Support to convert an existing JAR to a plug-in project
- To further align with OSGi R4.1, PDE supports the Bundle-ActivationPolicy manifest header, the replacement for the now-deprecated Eclipse-LazyStart header
- Plug-in Spy provides introspection into the IDE (shows active shell, part, ids, contributor, etc).
- Supports for individual source bundles (which are now shipped with the SDK)
- Execution environments can be specified for building plug-in projects and for runnin Eclipse Application launch configurations
- The Plug-in Registry view supports advanced bundle-related operations: start, stop, enable, disable
- Refactoring support to rename extension points
- Warning to ensure that source entries found in the build.properties are properly accounted for in the plug-in manifest
- Quick fixes to export/import packages
- Drag and drop in the manifest editor to support re-ordering of extensions, required plug-ins, classpath, etc.
- Highlighting of externalized strings within all XML editor source pages
- Enhanced cheat sheet editor
- Support to preview changes made by the Organize Manifests and the Externalize Strings wizards before they are applied to the plug-in manifest files
- Structural compare for plugin.xml files

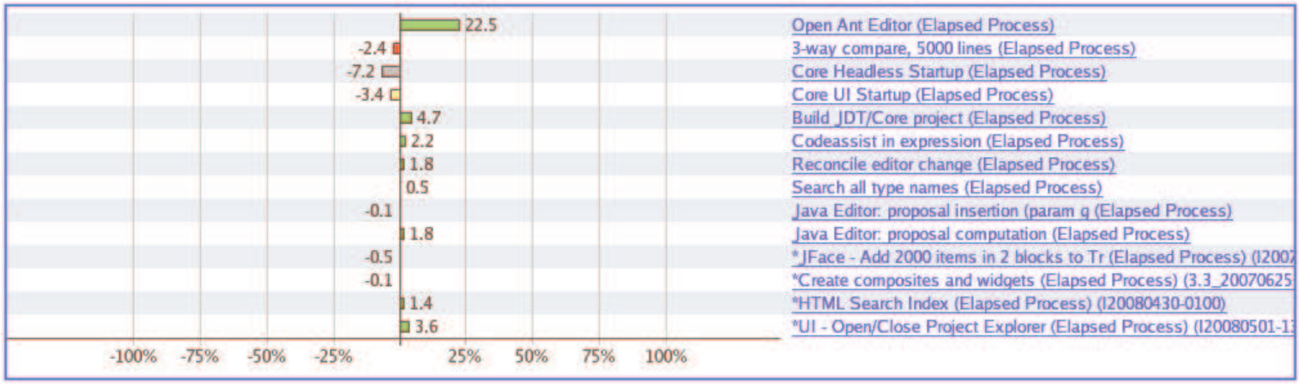
Deferred 3.4 Plan Items

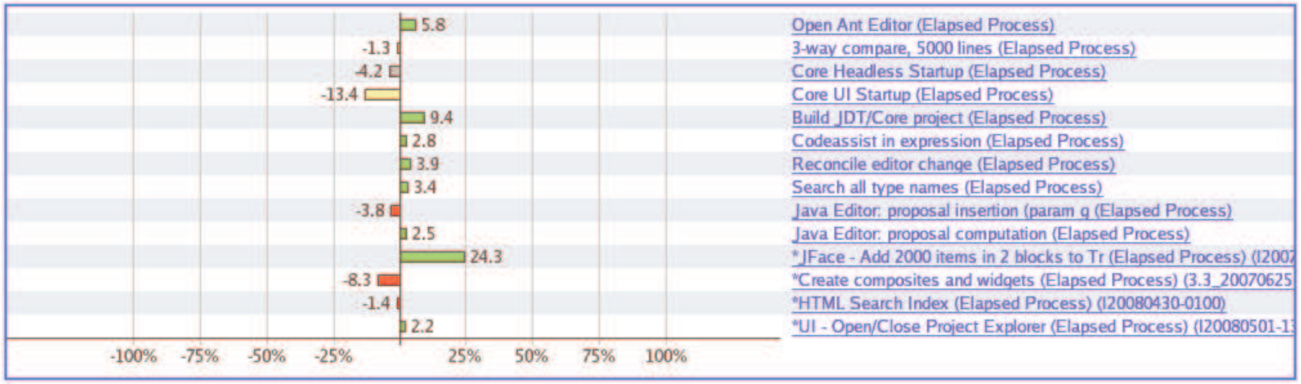
- Platforms
  - Complete SWT WPF port
- Reliability
  - Invest in PDE Build and Release Engineering
- The Future
  - Model the IDE

Performance of pre-3.4.0 vs. 3.3.0

Disclaimer 1: These results are not representing final performance stats for 3.4.0, rather work in progress during 3.4RC3.

Disclaimer 2: Test machines only have one CPU, thus do not fully reflect the benefits from multi-threaded Java compiler


RHEL 4.0 Sun 1.4.2_10 (3 GHz 2.5 GB)


Win XP Sun 1.4.2_10 (3 GHz 2 GB)

http://download.eclipse.org/eclipse/downloads/drops/I20080530-0100/performance/performance.php

3.4 Plug-in Changes from 3.3

Added Plug-ins (42)

- org.eclipse.core.net.win32.x86
- org.eclipse.ecf.filetransfer
- org.eclipse.ecf.identity
- org.eclipse.ecf.provider.filetransfer.ssl
- org.eclipse.ecf.provider.filetransfer
- org.eclipse.ecf.ssl
- org.eclipse.ecf
- org.eclipse.equinox.frameworkadmin.equinox
- org.eclipse.equinox.frameworkadmin
- org.eclipse.equinox.p2.artifact.repository
- org.eclipse.equinox.p2.console
- org.eclipse.equinox.p2.core
- org.eclipse.equinox.p2.director.app
- org.eclipse.equinox.p2.director
- org.eclipse.equinox.p2.directorywatcher
- org.eclipse.equinox.p2.engine
- org.eclipse.equinox.p2.exemplarysetup
- org.eclipse.equinox.p2.extensionlocation
- org.eclipse.equinox.p2.garbagecollector
- org.eclipse.equinox.p2.jarprocessor
- org.eclipse.equinox.p2.metadata.generator
- org.eclipse.equinox.p2.metadata.repository
- org.eclipse.equinox.p2.metadata
- org.eclipse.equinox.p2.reconciler.dropins
- org.eclipse.equinox.p2.touchpoint.eclipse
- org.eclipse.equinox.p2.touchpoint.natives

Added Plug-ins

- org.eclipse.equinox.p2.ui.sdk
- org.eclipse.equinox.p2.ui
- org.eclipse.equinox.p2.updatechecker
- org.eclipse.equinox.p2.updatesite
- org.eclipse.equinox.security.ui
- org.eclipse.equinox.security.win32.x86
- org.eclipse.equinox.security
- org.eclipse.equinox.simpleconfigurator.manipulator
- org.eclipse.equinox.simpleconfigurator
- org.eclipse.pde.api.tools.ui
- org.eclipse.pde.api.tools
- org.eclipse.pde.p2.ui
- org.eclipse.ui.views.log
- org.objectweb.asm
- org.sat4j.core
- org.sat4j.pb

Removed Plug-ins (1)

- org.eclipse.tomcat

Unchanged Plug-ins (2)

- org.eclipse.core.runtime.compatibility.auth
- org.junit4

Non-Code Aspects

- The 3.4 release will contain updated User and ISV documentation
- Community is very active
  - Mailing lists and newsgroups have steady activity
    - new E4 mailing list: eclipse-incubator-e4-dev@eclipse.org
  - Blogs dedicated to Eclipse are active e.g.
    - http://www.planeteclipse.org
  - Wiki content is growing
    - http://wiki.eclipse.org/index.php/Eclipse_Project
    - new E4 wiki: http://wiki.eclipse.org/E4

Non-Code Aspects

- Internationalization
  - Latin1 and Latin2 locales are supported in all operating environments
  - DBCS locales are supported on all platforms
  - BIDI locales (with mirroring) supported on Windows and Linux GTK, BIDI text supported on Mac.
  - GB18030-1 Chinese codepage standard is supported on Windows, Linux GTK and Mac.
- Localization
  - Tested for Localization (awaiting readiness of 'Eclipse Globalization Project')
- Accessibility
  - We have one accessibility issue: Welcome fonts/colors don't update when switching to high contrast (was already in 3.3)

Non-Code Aspects

- Articles, examples, and tutorials
  - New and updated articles and tutorials on eclipse.org (3)
    - http://www.eclipse.org/articles/article.php?file=Article-Forms33/index.html
    - http://www.eclipse.org/articles/article.php?file=Article-AddingHelpToRCP/index.html
    - http://www.eclipse.org/articles/article.php?file=Article-MemoryView/index.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 3.4, 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.
- API changes between 3.3 and 3.4 are checked automatically by new API tooling integrated into integration build process.
- The 3.4 migration guide identifies 7 changes:
  - http://dev.eclipse.org/viewcvs/index.cgi/org.eclipse.platform.doc.isv/porting/3.4/incompatibilities.html?view=co
  - http://dev.eclipse.org/viewcvs/index.cgi/org.eclipse.jdt.doc.isv/porting/3.4/incompatibilities.html?view=co
  - For each, a description of the change, what code is affected, and the action that needs to be taken is described.
  - We are not aware of any API compliant plug-ins breaking as a result of these changes.
  - The 3.4 migration guide also describes changes required to adopt mechanisms and APIs that are new in 3.4.
- The PMC is comfortable supporting the API that is in the Eclipse project 3.4

3.4 API – Platform

New

- New APIs have been to get all file buffers and to create a new text file buffer manager
- New APIs have been added document and annotation model types to improve performance and ease iterating over annotations
- Many new APIs have been added for the rich hover support
- New APIs have been added for the new colored labels support
- org.eclipse.jface.text.IRepairableDocumentExtension has been added to check whether document line information needs to be repaired
- Added several new APIs for the new Templates view support
- Added new APIs to better support emacs key bindings in content assist and quick fix proposal popup
- Added API to handle hidden resources
- Added symbolic links to ResourceAttributes API
- Added API in IFilePatch that return the before and after dates of a patch
- Added IFileStore##getFileStore(...)
- TokenComparator is API now
- DelegatingStorageMerger is API now
- New API added to ExpressionInfo to access all properties that have been accessed
- New constant added to IEvaluationContext for representing undefined variables
- Added 'Rename resource refactoring', 'Move resources refactoring' and 'Delete resources refactoring': Changes, refactoring descriptors and refactorings (pushed down from JDT).
- Wizards for 'Rename resource refactoring', 'Move resources refactoring' and 'Delete resources refactoring'

New

- ProcessorBasedRefactoring (org.eclipse.ltk.core.refactoring.participants) can now be instantiated (was abstract before)
- Added API ParticipantExtensionPoint (org.eclipse.ltk.core.refactoring.participants) to be used to manage contributions of participants
- New API added to AbstractTextSearchViewPage to control which elements are later sent to elementsChanged
- New attribute for extension point org.eclipse.search.searchResultViewPages to add help context ID for a page
- IHandler2/AbstractHandler now include a call to setEnabled(IEvaluationContext) from the framework
- new extension point org.eclipse.ui.services
- IContributionService and accessor on WorkbenchAdvisor
- IComparableContribution
- expression bindings in activities extension point
- New methods on ITriggerPointAdvisor
- Added to IMemento/XMLMemento
- Added java.util.EventListener to ISelectionListener
- MODE_FORCE_TEXT now has an declarative equivalent in actionSets and commands
- Link With Editor command
- Collapse All command+handler
- Switch page commands+handler
- views can be declaratively marked as non-restorable
- IEvaluationContext.UNDEFINED_VARIABLE now valid from an ISourceProvider
- MPEP now has API to deal with services for non-editor pages

3.4 API – Platform (cont’d)

New

- IEvaluationService provided as plugin API (all other services except ContextService work in terms of this)
- ShowIn menu API published (allow legacy contributions as well as new menu contributions)
- WidgetMethodHandler will use JDK 1.4 Swing API reflectively if available
- Added a 'createSash' to the AbstractPresentationFactory
- Added the ability to have a ViewPart implement ISizeProvider
- API to retrieve the encoding for a launch configuration
- API to allow launch shortcuts to provide configurations to launch for a given selection/active editor
- API to test if a launch configuration has specific attributes and to remove attributes
- API to allow "Create Watch Expression" action to work on arbitrary objects rather than just variables
- API to retrieve adapters from objects that consults the Debug Platform's adapter factories for objects that don't subclass PlaformObject
- API to allow a debug model presentation to control whether it is called in the UI thread.
- API for debug modules view ID and menu group constants
- API for debug error status code
- New customization preferences have been added in org.eclipse.help.hase the preferences topic.css, nav_css, narrow_css, disabled_css can be used to provide custom style sheets for help pages.
- org.eclipse.help.base.page_not_found can be used to specify the page shown when a link cannot be followed.
- org.eclipse.help.base.showBreadcrumbs allows breadcrumbs to be hidden

New

- The five remote help preferences can now accept a comma separated list of hosts, ports etc instead of just one.
- In org.eclipse.ui.intro the preferences INTRO_HOME_PAGE, INTRO_START_PAGE and INTRO_STANDBY_PAGE can be used to make a different page show when Welcome is first shown
- New extension point org.eclipse.ui.services
- IContributionService and accessor on WorkbenchAdvisor
- IComparableContribution
- Expression bindings in activities extension point
- Methods on ITriggerPointAdvisor
- Added to IMemento/XMLMemento
- Added EventListener to ISelectionListener
- MODE_FORCE_TEXT now has an declarative equivalent in actionSets and commands
- Link With Editor command
- Collapse All command+handler
- Switch page commands+handler
- Views can be declaratively marked as non-restorable
- IEvaluationContext.UNDEFINED_VARIABLE now valid from an ISourceProvider
- MPEP now has API to deal with services for non-editor pages
- IEvaluationService provided as plugin API (all other services except ContextService work in terms of this)
- ShowIn menu API published (allow legacy contributions as well as new menu contributions)
- WidgetMethodHandler will use JDK 1.4 Swing API reflectively if available
- Added a 'createSash' to the AbstractPresentationFactory
- Added the ability to have a ViewPart implement ISizeProvider

3.4 API – Platform (cont'd)

New

- New API was added to support the IME (International Input Method) for in-line editing
- New API was added to support images for the clipboard and drag and drop
- Support for tri-state buttons was added
- Browser can now be queried for the HTML text for the page
- Combo box can be dropped down and hidden programmatically
- New API was added to allow Controls to have an arbitrary shape based on a region
- FileDialog API was added to get the extension that the user selected
- Support for Vista progress bar states was added
- Alpha is now supported in top level shells
- Tables and trees no longer must have scroll bars
- Paths can be flattened into line segments
- TextLayout supports new strike through, border and underline styles
- Transform supports shearing
- RowLayout supports centering of controls
- Added IHandler2 interface which allows handlers to recieve contextual information from the framework to use in determining enabled states
- AAdded API (DataBindingContext.bindSet() and SetBinding) to allow ease of using databinding with Sets
- Added abstract ValidationStatusProvider class to easily allow monitoring of observable status changes
- Added ComputedList class for lazy list generation from observable inputs

New

- Added ListDiffVisitor for list delta processing
- Added CompositeMap which allows easy chaining of multiple maps
- Added MasterDetailObservable.detailMap() that allows for enumeration of observable properties
- Added MultiValidator for cross-observable constraint validation
- Added PojoObservables for observing java objects
- Added SWTObservables.observeDelayedValue() which allows for delayed observation of SWT controls
- Added IViewerObservable for observing jface viewers
- Added ObservableListTreeContentProvider, ObservableSetTreeContentProvider, and TreeStructureProvider as reusable jface providers that work with observables
- Added ObservableValueEditingSupport for integration of CellEditors with data binding
- Added StatusHandler, allowing for pluggable error handling within jface
- Added AbstractComboBoxCellEditor and ComboBoxViewerCellEditor as reusable combo-based cell editors
- Added StyledString, StyledCellLabelProvider, DecoratingStyledCellLabelProvider, and DelegatingStyledCellLabelProvider to allow for rich style control in jface viewers
- additional methods on MenuManager for image descriptions and better command integration
- Added StatusLineContributionItem for use in StatusLineManagers

3.4 API – Platform (cont'd)

Deprecated

- Deprecated ITextHover.getHoverInfo(ITextViewer, IRegion) replaced by ITextHoverExtension2. getHoverInfo2(ITextViewer, IRegion)
- Deprecated ProjectionDocument. getProjectionMapping() replaced by ProjectionDocument. getDocumentInformationMapping()
- Deprecated AbstractDocument. computeIndexInPositionList(List, int) replaced by AbstractDocument.computeIndexInPositionList(List, int, boolean)
- Deprecated SpellingProblem.removeAllInActiveEditor(ITextEditor, String) replaced by SpellingProblem.removeAll(ISourceViewer, String)
- Deprecated several constructors in DefaultInformationControl in favor of new ones
- Deprecated AnnotationPainter.SquigglesStrategy replaced by AnnotationPainter.UnderlineStrategy
- Deprecated AnnotationBarHoverManager.Closer replaced by closer from super class
- Deprecated AnnotationBarHoverManager.Closer.stop(boolean) replaced by AnnotationBarHoverManager.Closer.stop()
- Deprecated PerformRefactoringHistoryOperation. createRefactoring(RefactoringDescriptor, RefactoringStatus)

  replaced by RefactoringHistoryWizard . createRefactoring(RefactoringDescriptor, RefactoringStatus, IProgressMonitor))
- IDebugUIConstants.EXTENSION_POINT_MEMORY_RENDERIGNS has been deprecated to fix a spelling error. It has been replaced with EXTENSION_POINT_MEMORY_RENDERINGS. The value of the constant is the same
- EditionSelectionDialog is deprecated
- IStreamMerger is deprecated
- IFileStore#getChild is deprecated

Breaking changes

- Changed scheduling rules for IProject#setDescription
- IPartListener2 partVisible/partActive are now delayed on startup until the SWT controls are visible
- New problem view does not allow for object contributions on IMarker. Clients must use menus/commands/handlers to achieve this.
- WorkbenchActionBuilder now uses CommandContributionItems instead of RetargetActions

3.4 API – JDT

New

- API for invoking the batch compiler, reporting progress, and providing cancelation
- API to parse an int represented by an array
- API to format comments when a compilation unit is formatted
- API to format a set of regions
- Options to format annotations according to the annotated element
- API to get the kind of location of the completed token
- API to get the enclosing Java element at the completion location
- API to get the visible Java elements at the completion location
- API to return whether the completion is in a field reference (or a method reference) with a casted receiver
- API to return the receiver's signature and positions in the cases above
- APIs for Java 5 annotations: Java elements, annotations deltas, member value pairs, default value, search
- API to apply a text edit to a compilation unit's buffer
- Options to warn about a missing description in a Javadoc tag
- Option to warn about redundant super interface
- Option to exclude Exception and Throwable, or documented exception when warning about unused thrown exceptions
- Option to warn about usage of type arguments when invoking a non-generic method
- API to get the option corresponding to a problem id
- API to signal a build participant that the build is finished
- API to find a Java element given a binding key
- APIs for fine-grained search

New

- API to return the local element of a method reference match or a field reference match
- CamelCase APIs to restrict to match only the same count of parts Added APIs to CompletionProposalLabelProvider in order to support colored labels
- API for new Java refactorings: IntroduceParameterObjectDescriptor, ExtractClassDescriptor
- Added refactoring participation for 'Change Method Signature': new extension point and APIs
- Added API JavaUI.getEditorInputTypeRoot(IEditorInput) to get the Java editor inputs
- New API SharedASTProvider to gives access to the DOM AST used by the active Java editor
- New extension point javadocExportWizardPage allows to add pages to the Javadoc export wizard
- Both new Java project wizard pages are now API. Clients can reuse or modify the pages for their own New Java project wizards
- New extension point classpathFixProcessors allows to add functionality to fix a project class path when a type can not be resolved. For example PDE can add a plug-in dependency
- Added API for the JAR packager to support runnable JARs
- All occurrence actions available in for find occurrence. New actions and constants
- Support for external class folder selection
- New API to offer styled text labels of Java elements
- All action groups now also accept a special selection provider on construction which is used instead of the page selection provider
- Added ITestElement.getElapsedTimeInSeconds()

3.4 API – JDT (cont’d)

New

- API added on PreferenceConstants for specifying categories for excluded completion proposals
- Provided preference page IDs for build path preference pages on JavaUI
- Constants added for 'Show Breadcrumbs' action on IJavaEditorActionDefinitionIds
- API classes for Java launch shortcuts that can be extended by clients
- API for IDs on Java launch configuration tabs
- API to create VM installs from execution environment description files
- API to set/get arbitrary string attributes on VM installs
- API constant for setting the a socket listening connector on a Java launch configuration
- API for setting a source attachment on a JRE library
- A new extension point to contribute a wizard page for creating specific kinds of JREs
- Abstract API class to subclass when contributing a JRE creation wizard page
- API to set a range of values in an array object on a VM being debugged
- API for VM name and version of a VM being debugged, as well as unique IDs from objects in a VM
- API to determine if a VM supports selective garbage collection and to enable/disable garbage collection on specific objects
- API to set preference values for maximum number of instances/references to display in variables view

Deprecated

- The formatter option that inserts a new line after an annotation has been split into 3 options (for annotation on local variables, members and parameters). See DefaultCodeFormaterConstants.
- The factory method that creates an 'and' pattern (SearchPattern#createAndPattern(...)) has been deprecated since it has never fulfilled its specification.
- Deprecated some PreferenceConstants that got pushed down to JFace
- Deprecated org.eclipse.jdt.core.manipulation.ResourceRenameDescriptor and IJavaRefactorings.RENAME_RESOURCE: Pushed down to org.eclipse.ltk.core.refactoring
- Deprecated ResourceRenameDescriptor and IJavaRefactorings.RENAME_RESOURCE: Pushed down to org.eclipse.ltk.core.refactoring

Breaking changes

- Library entries on the build path can now target external folders
- Removed the assumption that a build path entry denoting a ZIP archive always had a .zip or .jar extension.
- Asking the qualified name of a binary type with a dot ('.') separator will now honor this separator

3.4 API – Equinox

New

- Added Secure Storage API
  - Extension point available for password providers
- Added JAAS Login integration API
  - Extension points for LoginModule, CallbackHandler, Configuration provider
- Added API to query information about signed content (bundles, plug-ins, jars etc.)
- Added API to establish authenticity of certificate chains used for signing

3.4 API – PDE

New

- New API launch configuration attribute determining whether to display only selected plugins
- New protected (API to subclasses) methods added to display the correct default value in combos in PDE plugin template wizard
- Override API methods in launch shortcuts and launch configurations to support an easier more-managed way of launching RCP apps
- New API methods to allow clients to add arbitrary headers to a manifest.mf generated by a template wizard
- New API class PluginReference to provide public implementation of IPluginReference

Tool Usability

- Eclipse is a superior IDE for Java tooling and plug-in development
- Many usability enhancements made in 3.4 to continue this tradition
  - Improve serviceability (http://wiki.eclipse.org/index.php/Status_Handling_Best_Practices)
  - Better managing and sharing settings and launch configurations
  - Sharing multiple projects simultaneously
  - Improved UI usability
  - Improved text editors productivity features
  - Simplified workflows in update manager replacement (p2)
  - Better resilience to code with errors in Java tools
  - More fine grain search criteria
  - More quick assists and fixes
  - 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

- Sources shipped as individual bundles (instead of being aggregated in existing plug-ins)
- Primary runtime is still a 1.4 JRE. Complementary functionalities on 5.0 JRE (junit4, APT 5) and 6.0 JRE (APT 6, compiler API)
- Eclipse now uses ECF and reships it.
- New provisioning (p2) support is replacing Update Manager
- 42 new plug-ins, 1 removed plug-in (org.eclipse.tomcat)

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 3.4 migration guide.
- org.eclipse.tomcat plug-in removed, but was never API.

Bugzilla

- Between June 25, 2007 and May 30, 2008 (RC3)
  - More than 14,000 reports were created
  - Over 12,100 were resolved
  - Over 4,900 were resolved without changing code
    - invalid, duplicate, worksforme, etc...
  - Over 450 were backported to 3.3.x maintenance
- Current state (RC3) is
  - 18 blockers, 57 critical
  - 0 P1, 34 P2 (2 planned for 3.4.1)
- 3.3 final state was
  - 23 blockers, 90 critical
  - 0 P1, 136 P2

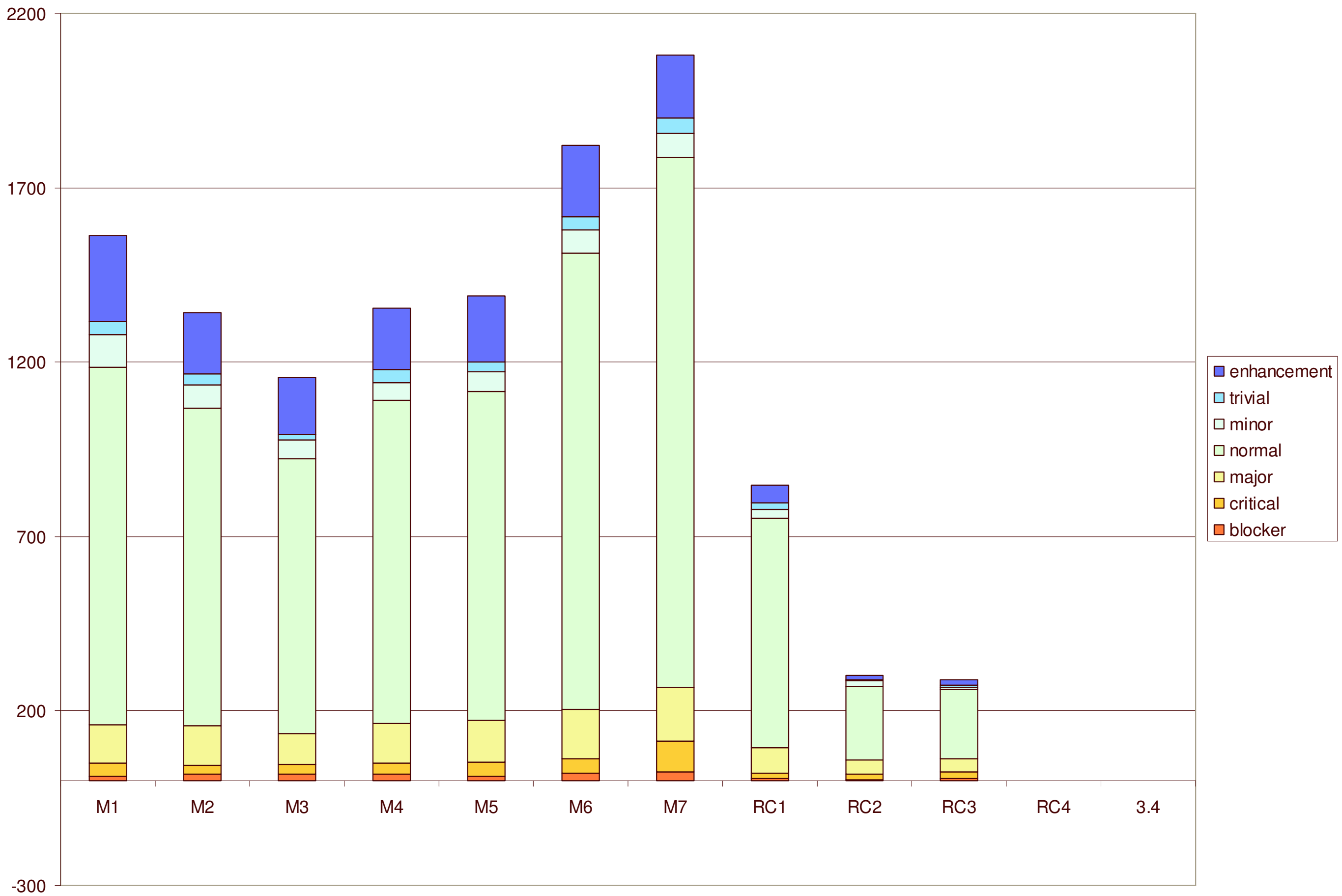
Bug resolution during 3.4

RESOLVED	M1	M2	M3	M4	M5	M6	M7	RC1	RC2	RC3	RC4	3.4	Total	*Diff 3.3*
blocker	12	17	19	17	12	20	25	7	4	7	?	?	**140**	*-35*
critical	39	28	28	32	41	43	87	16	13	17	?	?	**344**	*-43*
major	110	111	87	114	121	140	154	71	41	39	?	?	**988**	*-139*
normal	1024	912	788	927	940	1311	1520	658	213	198	?	?	**8491**	*-414*
minor	93	66	54	50	59	66	70	27	14	7	?	?	**506**	*-128*
trivial	40	32	18	39	28	38	44	19	4	5	?	?	**267**	*21*
enhancement	244	176	161	177	190	203	181	51	12	16	?	?	**1411**	*-128*
Total	**1562**	**1342**	**1155**	**1356**	**1391**	**1821**	**2081**	**849**	**301**	**289**	**?**	**?**	**12147**	*-866*
Diff 3.3	*27*	*30*	*-67*	*25*	*-213*	*263*	*-381*	*16*	*-57*	*37*	*-242*	*-304*	*-866*	

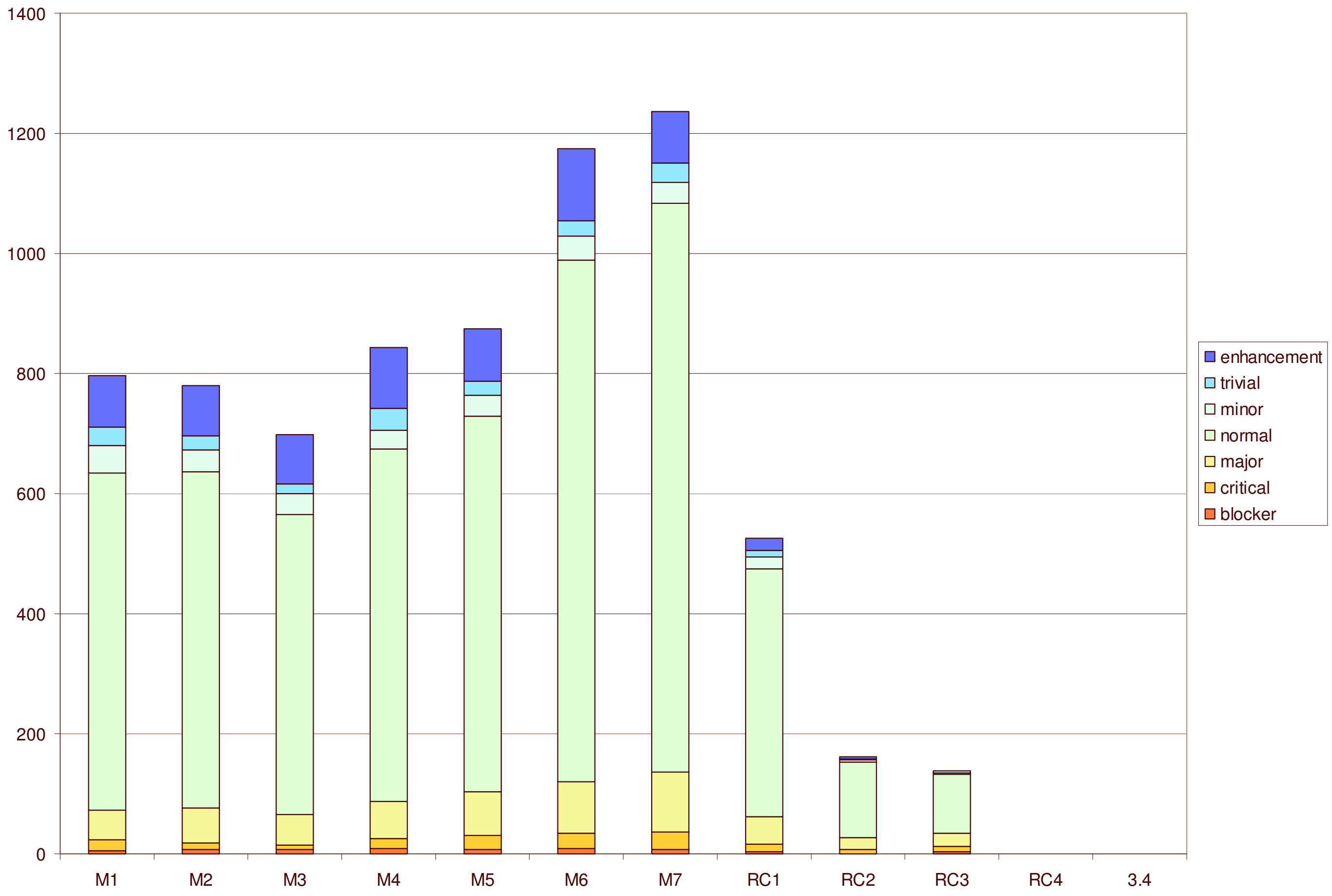
FIXED	M1	M2	M3	M4	M5	M6	M7	RC1	RC2	RC3	RC4	3.4	Total	*Diff 3.3*
blocker	5	7	7	9	8	10	8	4	0	4	?	?	**62**	*0*
critical	18	11	7	16	23	24	29	12	8	8	?	?	**156**	*-45*
major	49	59	52	62	73	86	99	46	20	22	?	?	**568**	*-65*
normal	562	559	499	587	625	870	948	412	124	99	?	?	**5285**	*-13*
minor	46	36	35	31	35	40	34	20	4	1	?	?	**282**	*-95*
trivial	31	25	16	36	24	25	33	12	3	4	?	?	**209**	*13*
enhancement	85	83	83	103	87	120	86	20	3	1	?	?	**671**	*-24*
Total	**796**	**780**	**699**	**844**	**875**	**1175**	**1237**	**526**	**162**	**139**	**?**	**?**	**7233**	*-229*
Diff 3.3	*-67*	*142*	*73*	*75*	*-112*	*200*	*-125*	*47*	*-50*	*15*	*-123*	*-304*	*-229*	

Resolved bugs

including fixed, invalid, …



Fixed bugs (only)



Standards

- OSGi
  - Service Platform Core Specification, Release 4.1
  - Elements of the OSGi Service Platform Service Compendium, Release 4.1
- Annotation Processing APIs
  - com.sun.mirror 1.5
  - javax.annotation.processing 1.6
- Java compiler API
  - javax.tools 1.6
- User Assistance consumes (parses) a small subset of RSS 1.0 to get news from eclipse.org
- JUnit 3.8.2 and JUnit 4.3.1
- J2SE
  - Tools are build against J2SE 1.4
  - Compiler can generate 1.3, 1.4, 1.5, and 1.6 code
  - Clients can run 1.4, 1.5 or 1.6.
- SWT
  - Win32, GDI, GDI+, OLE, IE, Carbon, Cocoa, Core Graphics, Quick Draw, Safari, ATSUI, X Windows, X/t, Motif, GTK, GDK, Pango, cairo, ATK, Mozilla, Uniscribe, WPF, OpenGL

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 and Linux GTK, bidirectional text on Mac OS X
- Accessibility support for Windows, Linux GTK and Mac OS X
- We are aware of one non-compliance with accessibility standards in the user interface:
  - The Welcome page fails accessibility because it doesn't pick up the font and color changes in high-contrast mode (already broken in 3.3, but not noticed. Experimental patch targeted for 3.4.1, real fix in 3.5)

Schedule

- Milestones every 6 weeks, 6 cycle duration
  - Feature and API frozen on March 28, end of M6 cycle
  - Adjusted M5/M6 duration (resp. 7 weeks and 5 weeks) for EclipseCon
  - http://www.eclipse.org/eclipse/development/eclipse_project_plan_3_4.html#Milestones
- Tracked schedule
  - All milestones except M6 delivered as promised
    - M6a was produced to address some invalid plug-in dependencies which could reduce the adoption of M6.
- 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 13, 2008
  - Increasingly stringent approval, checking, and change notification requirements in this stage
  - http://www.eclipse.org/eclipse/development/freeze_plan_3.4.php

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
  - Use blogging infrastructure at Eclipse.org
  - http://www.planeteclipse.org
- Some teams are using the eclipse-dev IRC channel
  - irc.freenode.net#eclipse-dev
  - irc://irc.freenode.net/#eclipse-e4
  - irc://irc.freenode.net/#equinox-dev
  - also see: http://wiki.eclipse.org/index.php/IRC
- The Eclipse team participates in code camps, conference presentations, and tutorials, including
  - EclipseCon, JavaOne, JavaWorld, JAOO, Eclipse Summit Europe, Eclipse Forum Europe, JAX, JAX Asia
- 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.
  - http://www.eclipse.org/eclipse/development/eclipse_project_log.html

Project Plan for Eclipse 3.5

- Pending - still in planning stage
- Preliminary priorities:
  - Platforms
  - Reliability
  - Consumability
  - The Future