

Eclipse Packaging Project

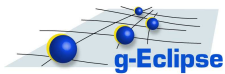
Graduation Review Version 1.0.0

Document classification: **©2008 by Markus Knauer.**
 Made available under the Eclipse Public License v1.0.

Date: **May 12, 2008**

Abstract: This document contains the Graduation Review Documentation for the Eclipse Packaging Project

Please add your comments and votes to https://bugs.eclipse.org/bugs/show_bug.cgi?id=xxxxxx



Contents

1 Overview	3
1.1 Scope and goals of the project	3
2 Features	4
3 Non-Code Aspects	6
3.1 User Documentation	6
3.2 Localization or Externalization	6
4 APIs	7
4.1 EPP Packaging	7
4.2 EPP UDC	7
5 Architectural Issues	8
5.1 EPP Packaging	8
5.2 EPP Usage Data Collector	8
6 Tool Usability	9
6.1 EPP Packaging	9
6.2 EPP Usage Data Collector	9
7 End-of-Life	11
8 Bugzilla	12
9 Standards	13
10 UI Usability	14
11 Schedule	15
12 Communities	16
13 IP Issues	17
14 Project Plan	18

1 Overview

Eclipse is seeing tremendous adoption of its tool and platform offerings. With thousands of downloads every day the Eclipse Platform SDK is the most popular download offering. The SDK includes everything needed for Eclipse plug-in development (Platform, PDE, JDT and Sources). When Eclipse was young, this was almost everything that Eclipse.org had to offer. This situation has changed over the last five years, and Eclipse is offering tools from A (AspectJ) to W (Web Tools Platform) in more than 80 projects.

At the same time, Eclipse is not only serving plug-in developers anymore, but also developers who want to explore Eclipse as a tool for a specific language or domain. Those developers are interested in downloading tools that may differ quite a bit from the Platform SDK download. For instance, developers often don't require the source code or Plug-in Development Environment (PDE), if they are just looking to use Eclipse as a Java IDE. It is possible to extend the Eclipse Platform SDK by using the update manager, but developers generally prefer a single download to get started. This is especially important for developers that are new to Eclipse.

The Eclipse Packaging project aims to provide a set of entry-level downloads for different user profiles. The goal is to improve the usability and out-of-box experience of developers that want to use multiple Eclipse projects.

1.1 Scope and goals of the project

- **Create entry level downloads based on defined user profiles.** The project defined and created the EPP downloads of Java Developer, Java EE Developer, C/C++ Developer and RCP Developer. These downloads are available from the main Eclipse download page and help users to start with Eclipse.
- **Provide feedback about the content.** With the integration of the EPP Usage Data Collector it will be possible to collect information about how individuals are using the Eclipse platform. The intent is to use this data to help committers and organizations better understand how developers are using Eclipse.
- **Help projects to integrate with each other.** With the package centric approach it is possible to build products which contain features of many different Eclipse projects. This leads to an early detection of dependency problems, better integration testing, and a project structure that is easier to consume.
- **Provide a platform that allows the creation of packages (zip/tar downloads) from an update site.** The core technology of the project enables the creation of download packages that are created by bundling Eclipse features from one or multiple Eclipse update sites.
- **Provide a central build infrastructure for the eclipse.org package builds.** The EPP package builds are running every night and allow early feedback on the content of the release streams (Europa, Ganymede).
- **Provide an installer** that improves the install experience of new users of Eclipse. (*postponed*)

Since June 2007, the project delivered packages for all Europa Releases with more than 8,000,000 downloads.

2 Features

EPP in version 1.0.0 includes

- a packaging component that uses the Eclipse Update Manager and the PDE packager
- build scripts that are used in the nightly package builds
- the UDC (Usage Data Collector) that collects data on an Eclipse client, e.g. an EPP package and sends the data back to the Eclipse Foundation servers.

`org.eclipse.packaging.core` core EPP packaging application. The application can be run from the command line and creates the packages defined in the EPP package configuration.

1. Use Eclipse Update Manager to pull features from update sites. Once downloaded into a local repository, the content can be reused in future builds.
2. Use the PDE packager to create packages for different platforms and configurations.
3. Everything driven by a single configuration file that contains additional information to create the website content of the packaging website.

`org.eclipse.usagedata.*` client components of the Eclipse Usage Data Collector. The usage data monitors what is being used and when (timestamp), including

- Loaded bundles
- Commands accessed via keyboard shortcuts
- Actions invoked via menus or toolbars
- Perspective changes
- View usage
- Editor usage

Captured data is associated with a user through a combination of workstation and workspace ids that are automatically generated by the collector. This identification is not tied to any personal information about the user. Where possible, the usage data collector also capture the symbolic name and version of the bundle contributing the command/action/perspective/view/editor.

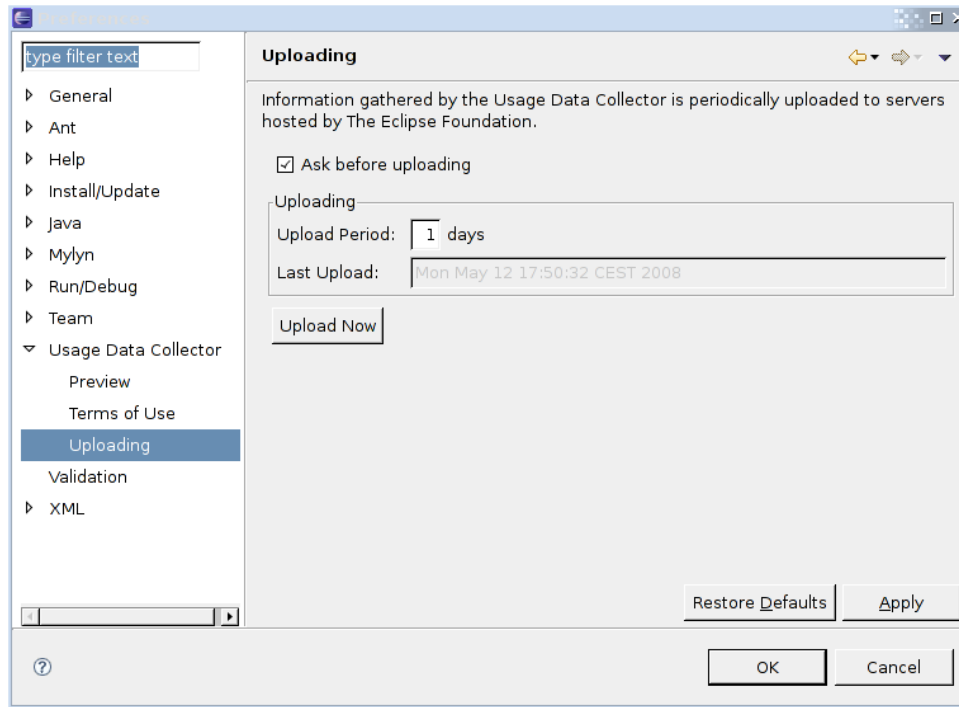


Figure 2.1: EPP Usage Data Collector Upload

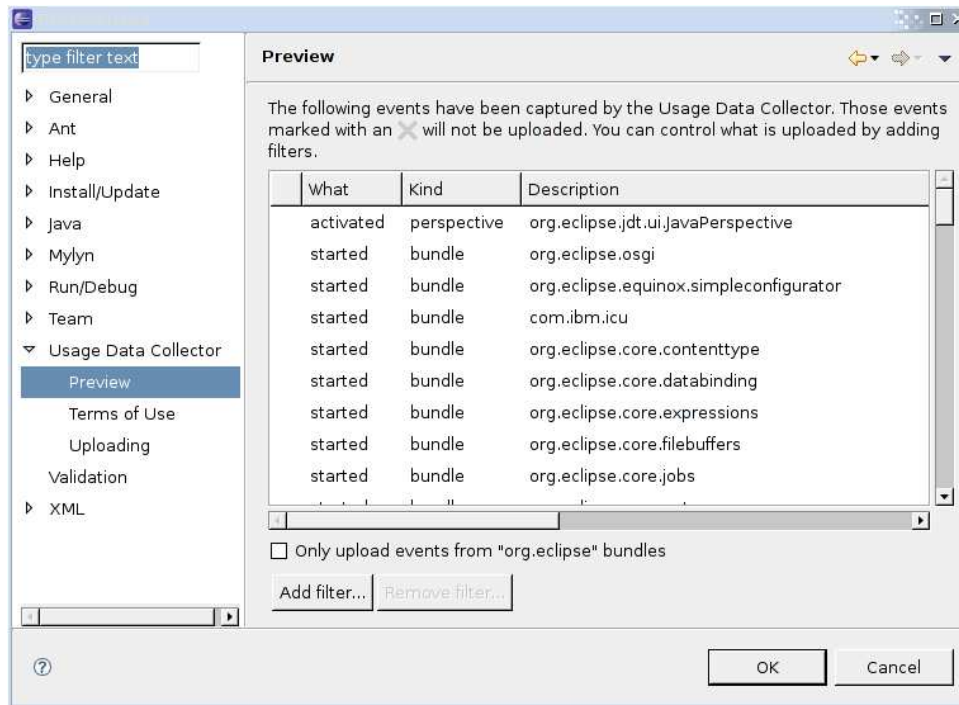


Figure 2.2: EPP Usage Data Collector Preview

3 Non-Code Aspects

3.1 User Documentation

User documentation has been created for this initial release mainly in form of web pages or wiki pages (<http://wiki.eclipse.org/index.php/Category:EPP>):

- How-to build your own package
- How-to specify an EPP configuration file
- How-to start as a Package Mainainer
- Package Testing
- Build Infrastructure

3.2 Localization or Externalization

EPP is available for the English language; strings are externalized.

4 APIs

4.1 EPP Packaging

The EPP packaging application does not define any extension points. It uses an XML configuration file with a format specified by EPP (http://wiki.eclipse.org/EPP/Configuration_File.Format). It contains

- name, perspective, product ID
- a set of update sites
- a set of required features
- the base platform archive
- a platform specific eclipse.ini file

In the future we will add metadata with a package description that can be used on a webpage etc (see <http://wiki.eclipse.org/EPP/Packaging-Site>). This metadata is optional.

4.2 EPP UDC

The EPP UDC functionality is split into

`org.eclipse.epp.usagedata.gathering` which defines the `org.eclipse.epp.usagedata.gathering.monitors` extension point; this extension point is used to plug new monitors to Eclipse. Three monitor implementations are included: `PartUsageMonitor`, `BundleUsageMonitor`, `CommandUsageMonitor`. And it defines the `org.eclipse.epp.usagedata.listeners.event` extension point; implementors act as receiver of the events generated by the monitors.

`org.eclipse.epp.usagedata.recording` which defines the `org.eclipse.epp.usagedata.recording.uploader` extension point; this extension point allows the creation of different systems to process the data collection.

`org.eclipse.epp.usagedata.ui` defines the UI elements (i.e. preferences pages) and provides an implementation of the uploader extension point that uploads the UDC data to an Eclipse Foundation server.

5 Architectural Issues

5.1 EPP Packaging

The EPP configuration file will be modified to reflect changes in its downstream 'consumers' (package build, website, content management system, installers, ...). These changes will contain additional elements and therefore are compatible with the old versions.

5.2 EPP Usage Data Collector

The current implementation of the UDC works in an RCP environment. Future planned enhancements include a UDC that will run unmodified in a RAP environment. Currently, there are no known API changes necessary.

6 Tool Usability

6.1 EPP Packaging

With more than 8,000,000 downloads in the last 10 months, packages generated by EPP have been proven stable.

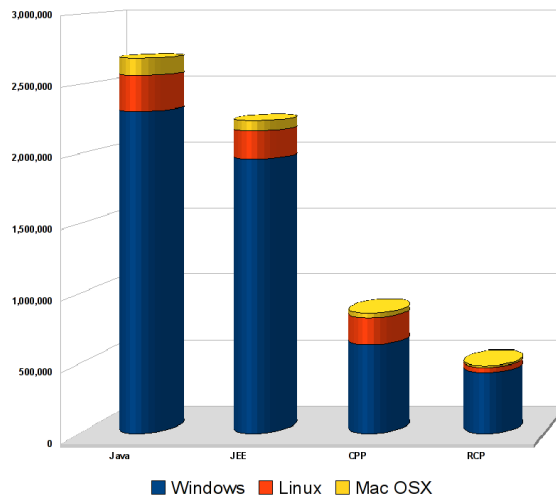


Figure 6.1: EPP Download Statistics (first 9 months)

The EPP packages are available from the main eclipse.org download page and all community packages from a Drupal driven site.

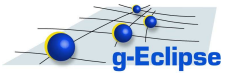
6.2 EPP Usage Data Collector

Early adopters of the Ganymede Milestone Packages already sent their UDC data to the Eclipse Foundation. The following data was collected on a 14 days period from 2008-02-29 to 2008-03-14: 2,359,688 usage data events were been generated by 453 users (an average of 5,209 events per user):

Views	
org.eclipse.jdt.ui.PackageExplorer	26257
org.eclipse.ui.console.ConsoleView	12766
org.eclipse.ui.navigator.ProjectExplorer	7522
org.eclipse.search.ui.views.SearchView	4941
org.eclipse.debug.ui.DebugView	4882
org.eclipse.ui.views.ProblemView	3700
org.eclipse.ui.views.ContentOutline	3526

Editors	
org.eclipse.jdt.ui.CompilationUnitEditor	17129
org.eclipse.wst.xml.ui.internal.tabletree.XMLMultiPageEditorPart	2595
org.eclipse.jdt.ui.ClassFileEditor	2175
org.eclipse.ui.DefaultTextEditor	1387
org.eclipse.cdt.ui.editor.CEditor	1279
org.eclipse.compare.CompareEditor	1176
org.eclipse.jst.jsp.core.jspsource.source	1164

Figure 6.2: EPP Usage Data Collector Results



7 End-of-Life

This is an initial release, so there are currently no deprecated or removed APIs or features.

8 Bugzilla

As of 2008-05-12 there are 136 bugs in technology/epp. In the end, there will be no blockers left and all open bugs for 1.0.0 will be fixed until the release.

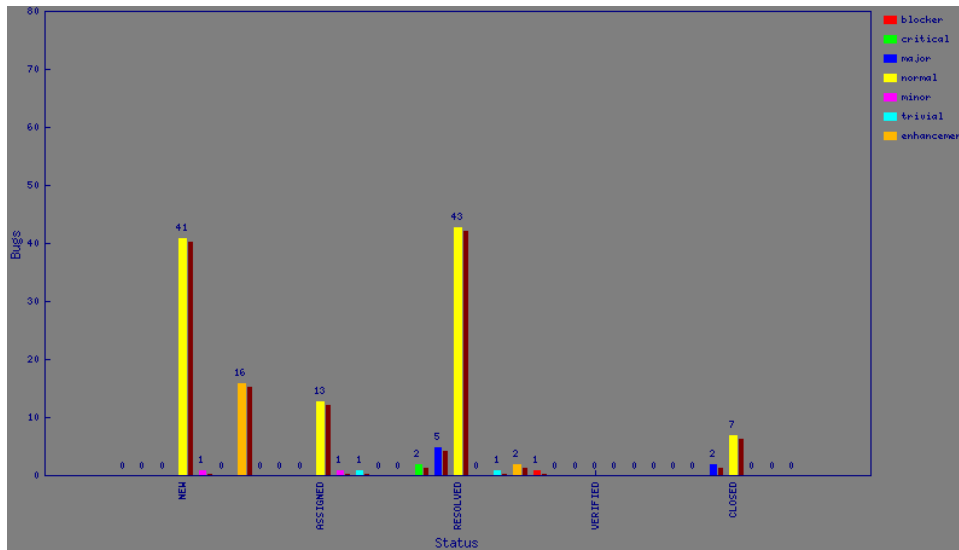


Figure 8.1: EPP Bugzilla Overview

9 Standards

- EPP uses Java 1.5, compatible with Eclipse 3.3 and 3.4

10 UI Usability

Only the EPP UDC contains UI elements in form of preferences pages.

- Following Eclipse UI usability guidelines
- Usability changes based on users' feedback

11 Schedule

The plan of the Eclipse Packaging Project is always in parallel with the release train plans, i.e. the Europa and the Ganymede release trains (<http://www.eclipse.org/epp/plan.php>). Within the Europa timeframe, EPP

- defined initial packages for the Eclipse download page
- built these packages for Windows, Linux-GTK, MacOSX-Carbon

Due to a lack of resources, the installer could not be provided in time and is currently postponed.

Within the Ganymede timeframe, EPP

- created a package eco system where package maintainer can add new packages.
- developed and integrated the EPP Usaged Data Collector in all packages.
- delivers milestone builds and nightly build based on the Ganymede Update Site in time.
- set up an automated build process and integrates it with the Ganymatic build.
- re-defined initial packages for the Eclipse download page
- builds these packages for Windows, Linux-GTK, Linux-GTK64, and MacOSX-Carbon

12 Communities

- Active committers (5) and contributors from 4 partners (INNOOPRACT, Inc., Eclipse Foundation, Instantiations, Xored)
- Participation (Talks, BoF) at Eclipse events (EclipseCon 2007, Provisioning Workshop 2008, EclipseCon 2008)
- Public conference calls
- Developer mailing list with about 200 e-mails, newsgroup with about 150 messages
- The Eclipse Packaging Project

13 IP Issues

See IP Log at <http://www.eclipse.org/epp/project-info/eclipse-project-ip-log.csv>

- Initial code contribution got IP clearance from Eclipse Legal (CQ1395, CQ1536, CQ1898)
- All CQs of this release got IP clearance from Eclipse Legal (CQ1913, CQ1914, CQ1915)
- External contributions are listed in the IP Log and were submitted via Bugzilla

List of committers:

- Wayne Beaton - committer since 12/2007
- Alexander Kazantsev, initial committer
- Markus Knauer, initial committer
- Dan Rubel, initial committer
- Mark Russell, initial committer
- Elias Volanakis, initial committer

14 Project Plan

Version 1.1.0 is scheduled for October 2008 (Ganymede Fall). Among other improvements it will include

- an update of the Usage Data Collector including a RAP-enabled version
- a p2-ification of remaining parts.