Eclipse Packaging Project

Release Review Version 1.1.0

Document classification:	**©2009 by Markus Knauer.** **Made available under the Eclipse Public License v1.0.**
Date:	**June 1, 2009**

Abstract: This document contains the Release Review Documentation for the Eclipse Packaging Project (EPP). The 1.1.0 EPP release is scheduled for 2009-06-24 together with the release of Galileo.

Contents

1 Overview

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 Eclipse IDE for Java Developers, Eclipse IDE for Java EE Developers, Eclipse IDE for C/C++ Developers, and Eclipse for RCP/Plug-in Developers. These downloads are available from the main Eclipse download page. In addition to that, other packages maintained by the community and coordinated by EPP are being made available, such as Eclipse IDE for PHP Developers, Eclipse Modeling Tools, Eclipse IDE for Java and Report Developers, and Pulsar for Mobile Java Developers.
- **Provide and integrate the EPP Usage Data Collector.** The Usage Data Collector collects 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.
- **Provide a dynamic installer that improves the install experience of new users of 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 sturcture that is easier to consume.
- **Provide a central build infrastructure for the eclipse.org package builds.** The EPP package builds are running on Hudson and allow early feedback on the content of the release streams (Europa, Ganymede, Galileo).

Since June 2007, the project delivered packages for all release trains, including Europa, Ganymede and all of their service releases and had millions of downloads.

2 Features

EPP for Galileo in version 1.1.0 includes

- Buckminster scripts that generate the p2 repository with the package definitions
- 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.

Note that the version number of the packages (currently 1.2.0 for Galileo) is independent from the version number of the software that is used by and delivered by the EPP project.

`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
- Environment, JVM, platform (new)

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.

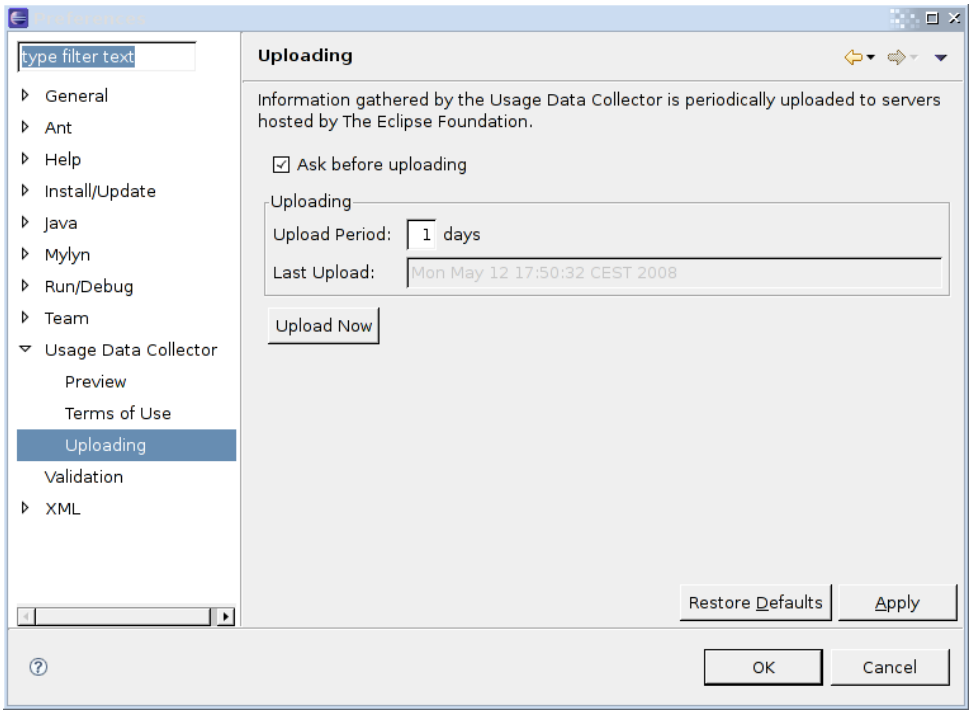

Figure 2.1: EPP Usage Data Collector Upload

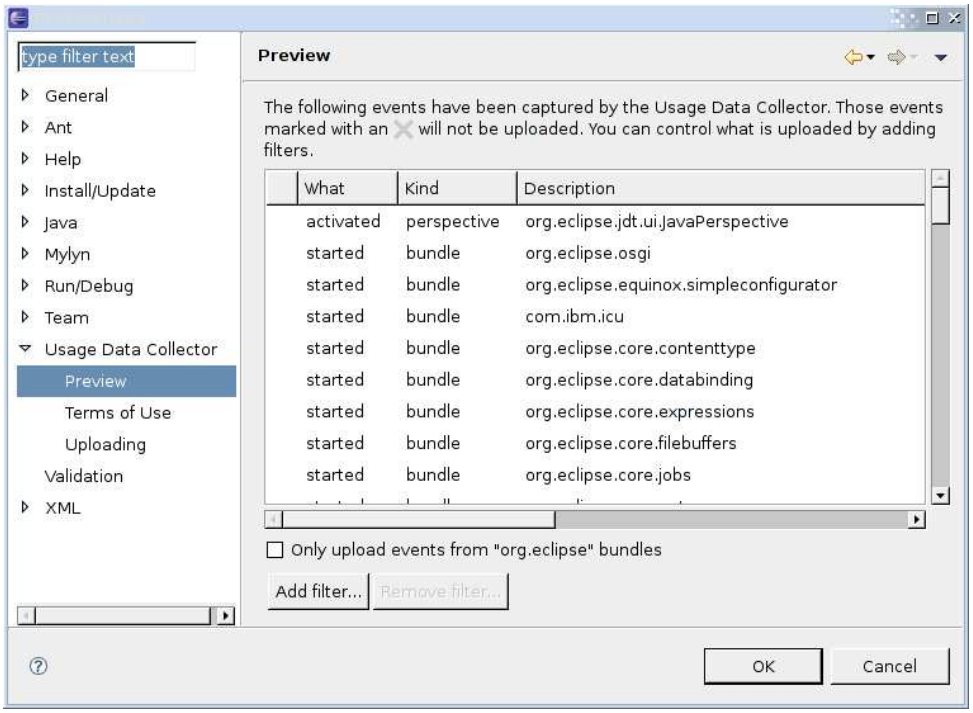

Figure 2.2: EPP Usage Data Collector Preview

3 Non-Code Aspects

3.1 User Documentation

User documentation has been created and updated for this release in the form of web pages or wiki pages (http://wiki.eclipse.org/index.php/Category:EPP):

- How-to specify an EPP configuration file
- How-to create the package definition files, package defining feature, branding plug-in
- How-to build your own package
- Package Testing
- Build Infrastructure

3.2 Localization or Externalization

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

Components in Babel are provided but the team does not translate the strings.

4 APIs

4.1 EPP Packaging

The mechanism how the packages are generated has been changed since the last release. The former technology was build on top of the Eclipse update manager technology. Since this technology is now deprecated and Galileo relies on the usage of the new p2 technology, EPP had to change its own underlying technology.

EPP packages are now build in two steps and allow every modification that is possible with the standard Eclipse branding.

1. Create a p2 repository with the package metadata; automated with technology from the Buckminster project.
2. Use the p2 metadata from step 1 in order to create the packages with the p2 director application.

The old XML configuration file with a format specified by EPP[1] is used to generate the web pages only. The package definition is using standard Eclipse technology, such as features and plugins for branding.

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; implementators 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.

[1] http://wiki.eclipse.org/EPP/Configuration_File_Format

5 Architectural Issues

5.1 EPP Packaging

The EPP configuration file is not used any more for the package build. Using Eclipse standards, such as feature.xml, etc.

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.

6 Tool Usability

6.1 EPP Packaging

With millions of downloads in the last 2 years, packages generated by EPP have been proven stable.

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

More than 120.000 users of the Ganymede Packages already send their UDC data to the Eclipse Foundation every month. The results are available from the Eclipse Foundation web pages (http://www.eclipse.org/org/usagedata/).

7 End-of-Life

The EPP packaging application cannot be supported any more. The project uses now existing technology from Equinox p2.

8 Bugzilla

As of 2009-06-01 there are 347 bugs in technology/epp, 126 listed with status new, assigned, open. There are no blockers, 5 critical bugs. In the end, there will be no blockers left and all critical bugs for 1.1.0 will be fixed until the release.

9 Standards

- EPP uses Java 1.5, compatible with Eclipse 3.4 and 3.5.

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 Galileo release train (http://www.eclipse.org/projects/project-plan.php?projectid=technology.packaging).

The scheduled dates for the Ganymede release have been met and the packages were released together with the Ganymede Release in June 2008. Updates have been created and made available for all Ganymede Service Releases.

For the Galileo release, EPP started to deliver initial packages for Galileo M6, and provided regular package builds based on the new build system with Galileo M7. Since then, EPP delivers all packages for each of the Galileo milestones and release candidates.

12 Communities

- Active committers (6) and contributors from 4 partners (EclipseSource Inc., Eclipse Foundation, Cloudsmith Inc., Instantiations, Xored)
- Participation (Talks) at Eclipse events (EclipseSummit 2008, EclipseCon 2009)
- Public conference calls
- Developer mailing list with about 630 e-mails, newsgroup with more than 200 messages
- The Eclipse Packaging Project has been mentioned in many blog postings, other mailing lists (e.g. cross-project-issues-dev)
- Participation in the Eclipse Planning Council and in the Eclipse Architecture Council

13 IP Issues

See IP Log at http://www.eclipse.org/projects/ip_log.php?projectid=technology.packaging

- Initial code contribution got IP clearance from Eclipse Legal (bug 244666)
- External contributions are listed in the IP Log and were submitted via Bugzilla

List of committers:

- Wayne Beaton - committer since 12/2007
- Henrik Lindberg - committer since 06/2008
- Jordi Bhme Lpez - committer since 06/2008
- Alexander Kazantsev, initial committer
- Markus Knauer, initial committer
- Jeff McAffer - committer since 06/2008
- Dan Rubel, initial committer
- Mark Russell, initial committer

Committer emeritus (committers who have been removed from the list of EPP committers during the last 12 months):

- Elias Volanakis, initial committer
- Leif Frenzel, initial committer

14 Project Plan

Version 1.1.1 is scheduled for October 2009 (Galileo SR1) and will contain mainly bugfixes.
A detailed plan is not yet available.