Remote Application Platform (RAP) 2.0

Release Review

January 21, 2013, Ralf Sternberg, EclipseSource

Project page: http://eclipse.org/rap

Communication Channel: RAP Mailing List rap-dev@eclipse.org

IP Log: http://eclipse.org/rap/iplog/ - has been approved

Project Plan: http://eclipse.org/projects/project-plan.php?planurl=http://www.eclipse.org/rap/plan/plan-2.0.xml

New and Noteworthy: http://eclipse.org/rap/noteworthy/

Eclipse Remote Application Platform (RAP)

RAP is a platform for modular business applications that run on a server and can be accessed from various client implementations. It enables developers to build rich user interfaces using the Eclipse tools and common APIs.

RAP provides a powerful, multi-platform widget toolkit with SWT API that enables developers to write applications entirely in Java and re-use the same code on different platforms. Regardless of the client platform, RAP applications run on a server that communicates with the clients over HTTP. Applications can be deployed on any servlet container. The default web client uses JavaScript to render the UI in the browser.

The project provides a complete target platform based on Equinox, including subsets of SWT, JFace, and Workbench APIs. The combination of RCP and RAP allows to develop desktop and web applications from a single code base (single sourcing). With the RAP OSGi integration, they can be composed of modules and communicate using the OSGi service model. The core library can also be used in traditional web applications without OSGi.

Features

The RAP project provides these four features:

Runtime

The RAP runtime includes all org.eclipse.rap.* runtime bundles, i.e.:

- RWT, the RAP implementation of SWT
  - provides broad coverage of SWT 3.7 APIs
  - additional API for the web, such as theming, browser history, session store, etc.
  - includes the default web client that uses JavaScript to render the UI in the browser
  - complies with the Servlet specification 2.3 … 3.0
  - can also be used as a library in traditional web applications without OSGi
- RWT OSGi integration
  - provides API to launch application in an OSGi environment
  - auto-starts RAP applications that are registered as an OSGi service

- RAP versions of JFace and JFace databinding
  - broad coverage of JFace 3.7 APIs
- RAP versions of Workbench 3.7 bundles
  - org.eclipse.ui.workbench
  - org.eclipse.ui.views
  - org.eclipse.ui.cheatsheets
  - org.eclipse.ui.forms

Basic Target Requirements

This feature complements the runtime with bundles from the Eclipse Platform that are needed for a complete RAP target platform:

- Equinox OSGi, including
  - the Eclipse extension registry
  - the Equinox console
  - declarative services
  - the servlet bridge to deploy RAP applications as web archive (.war)
- Eclipse core bundles, basically
  - core.commands
  - core.databinding
  - core.runtime
  - core.jobs
- Servlet API
- ICU replacement (icu.base)
- Jetty

Examples

- A workbench-based application
- RWT Controls demo application
- Examples demo

Tools

The RAP Tools include

- the RAP Developer's Guide integrated into the Eclipse help
- a simple target installer to download and install a RAP target platform
- launcher for RAP applications to start application from the IDE
- project templates

Project Health

Committer Activity

- active committers: 5
- continuous committer activity
- organizations: EclipseSource, Individuals
- see http://dash.eclipse.org/dash/commits/web-app/project-diversity.cgi

Software Quality

- continuous integration including more than 5000 unit tests (https://hudson.eclipse.org/hudson/job/rap-2.0-runtime/)
- more than 1,600 JavaScript unit tests for the client
- strict coding standards back our code quality (http://wiki.eclipse.org/RAP/CodingStandards)
- active community of users and testers

RAP Incubator Project

The independent RAP Incubator project is the home for add-ons and experimental new features for RAP. The project provides a common build infrastructure for its components to ensure compatibility with the latest version of RAP. An incomplete list of components:

- ClientScripting
- FileUpload
- Nebula Grid
- Common Navigator framework (CNF)
- Graphical Editing Framework (GEF)

Contributers

Many of the components in the RAP Incubator project are maintained by external contributors. Moreover, the IP log lists more than 100 patches from external contributors to the RAP project itself.

Community

RAP has an active community that discusses with the RAP team in the newsgroup and reports issues in Bugzilla. The RAP team participates in many Eclipse Demo Camps and conferences including EclipseCon, Eclipse Summit Europe and Eclipse Day Florence.

Bug Statistics

including enhancement request, without RAP Incubator

Bugs newly reported since June 27, 2012:	291
Bugs RESOLVED FIXED since June 27, 2012:	208
Bugs RESOLVED with a 2.0 target milestone:	208

Documentation

- Complete API documentation available online
- RAP developer guide has been improved and extended
- A huge set of resources available for SWT/RCP that can be reused for RAP
- See http://eclipse.org/rap/documentation/ for more information

Integration with other Eclipse projects

- EMF provides RAP support
- Scout uses RAP for the web client
- The Gyrex Admin UI is based on RAP

RAP 2.0

The 2.0 release marks the completion of the project's migration to a new, open communication protocol based on JSON. This protocol opens RAP for alternative client implementations. While RAP has always been a platform for applications that run on a remote server, this change moves the remote aspect into the center of the project. To reflect this, the project name has been changed to "Remote Application

Platform".

Main Achievements

- Completed new JSON-based client-server protocol, used for all C/S communication now
- Revised and consolidated RWT's APIs (breaking changes)
- New Client API to provide client-specific services
- New Remote API for development of custom widgets and add-ons, including JavaScript API
- Complete rewrite of RWT's event system

See http://eclipse.org/rap/noteworthy/2.0/ for details

Schedule

RAP 2.0 delivered milestone builds every 6 weeks according to the Kepler schedule (M1–M4), see [1]. All milestones have been delivered on time.

For the end game, our ramp down plan [2] applies.

There are no service releases planned for this release. Instead we plan a follow-up 2.1 release together with Eclipse Kepler.

[1] http://www.eclipse.org/projects/project-plan.php?projectid=rt.rap#release_milestones
[2] http://wiki.eclipse.org/RAP/Ramp_down_plan

Project Plan for RAP 2.1

We plan to release RAP 2.1 together with Eclipse Kepler in June 2013. The plan is still in an early stage, but will likely include:

- Add support for multi-tab browsing
- Support custom exception handlers
- Improve Resources API for custom widgets (proposed)
- Allow reconnect to a running session (proposed)

Further Information

- http://eclipse.org/rap - RAP project page
- http://wiki.eclipse.org/RAP - RAP wiki
- http://eclipse.org/rap/demos/ - Demo applications