Single Sourcing
The Rich Ajax Platform

Benjamin Muskalla
RAP Development Team
Who We Are

RAP Development Core Team
Karlsruhe, Germany
Knowledge?
Web 2.0?
A solution?
**single sourcing** [ˈsɪŋɡl sɔːrsciŋ]

Single source publishing, also known as single sourcing, allows the same content to be used in different documents or in various formats.
single sourcing

Single source publishing, also known as single sourcing, allows the same content to be used in different documents or in various formats.
Impossible?
Exchange the runtime!
Layers of RCP

- Application
- Workbench
- JFace
- SWT
- Operating System
RAP Layers

Application

Workbench

JFace

RWT (Server)

Servlet Container

HTTP

RWT (Client)

Web Browser
On the Surface
WHY?
It's possible!
It saves time...
...and money!
How much can we reuse?
Code Reuse

80%–98% is possible

platform specific code

shared code
What about the 2-20%?
RAP Runs in a Browser

- File system
- GraphicsContext

File upload

- RCP

- RAP
RAP is Multi-User!
API Differences

RCP

Desktop-only features

RAP

Web-specific features
Techniques
Hand Tools

Eclipse SDK

RAP SDK

includes
- RAP Runtime
- Tooling
- Help
Workplace
Shared Projects

- **RAP Workspace**
  - `example.core`
  - `example.rap`

- **RCP Workspace**
  - `example.core`
  - `example.rcp`

- **Projects**
  - `projects`
  - `example.core`
  - `example.rap`
  - `example.rcp`

- **Workspaces**
  - `workspaces`
    - `RAP` (Empty)
    - `RCP` (Empty)
Example Application

RCP Mail Demo

New Plug-in Project

Templates
Select one of the available templates to generate a fully-functioning plug-in.

Create a plug-in using one of the templates

Available Templates:

- Hello RCP
- RCP application with a view
- RCP application with an intro
- RCP Mail Template

This wizard creates a standalone RCP application that shows how to:
- add top-level menu and toolbar with actions
- add keybindings to actions
- create views that can’t be closed and multiple instances of the same view
- create perspectives with placeholders for new views
- use the default About dialog
- create a product definition

Extensions Used:
- org.eclipse.core.runtime.applications
- org.eclipse.core.runtime.products
- org.eclipse.ui.commands
- org.eclipse.ui.perspectives
- org.eclipse.ui.views
Import into RAP Workspace
Dependencies

Package Imports
Optional Dependencies

org.eclipse.ui

org.eclipse.rap.ui

Shared Bundle
Optional Dependencies
Only Two Errors Left
Fragments

Host Bundle

Shared Code

Fragments

RCP

RAP
Create Fragments
Move extensions
Only One Error Left
Delegation

Bundle

Fragment

SingleSourcingHelper

- IMPL
  + foo()
  # fooInternal()
+ bar()
# barInternal()

ImplementationLoader

+ newInstance(Class)

delegates to IMPL#fooInternal()

SingleSourcingHelperImpl

+ fooInternal()
+ barInternal()
API Differences

RCP

```java
exitAction = ActionFactory.QUIT.create(window);
register(exitAction);

aboutAction = ActionFactory.ABOUT.create(window);
register(aboutAction);
```

RAP

```java
aboutAction = new AboutAction(window);
register(aboutAction);
```

RCP + RAP

```java
aboutAction = AboutActionHelper.create(window);
register(aboutAction);
```
public abstract class AboutActionHelper {

    private static final AboutActionHelper IMPL;
    static {
        IMPL = (AboutActionHelper) ImplementationLoader.newInstance(AboutActionHelper.class);
    }

    public static IWorkbenchAction create( IWorkbenchWindow window ) {
        return IMPL.createInternal( window );
    }

    abstract IWorkbenchAction createInternal( IWorkbenchWindow window );
}
public class ImplementationLoader {

    public static Object newInstance(Class type) {
        String name = type.getName();
        Object result = null;
        try {
            result = type.getClassLoader().loadClass(name + "Impl").newInstance();
        } catch (Throwable throwable) {
            // TODO handle exception
        }

        return result;
    }
}
Implementations

**RCP**

```java
public class AboutActionHelperImpl extends AboutActionHelper {

    IWorkbenchAction createInternal(IWorkbenchWindow window) {
        return ActionFactory.ABOUT.create(window);
    }
}
```

**RAP**

```java
public class AboutActionHelperImpl extends AboutActionHelper {

    private final class AboutAction extends Action implements IWorkbenchAction {

        IWorkbenchAction createInternal(IWorkbenchWindow window) {
            return new AboutAction(window);
        }
    }
```
Zero Errors
public class EntryPoint implements IEntryPoint {

    public int createUI() {
        Display display = PlatformUI.createDisplay();
        return PlatformUI.createAndRunWorkbench(display, new ApplicationWorkbenchAdvisor());
    }
}
Deployment?
OSGi vs. J2EE ?
As server...

- Application Bundles
- Servlet Container Bundle
- RAP Bundles
- OSGi (Equinox)
..or embedded

- Application Bundles
- RAP Bundles
- OSGi (Equinox)
- OSGi Servlet Bridge
- J2EE Servlet Container
Conclusion?
“Cool, one runtime to rule them all.”

Chris Aniszczycy, EclipseSource
The **actors**

RAP
http://www.eclipse.org/rap/

RCP
http://www.eclipse.org/rcp
Thanks for listening!

www.eclipsesource.com/rap
www.eclipse.org/rap