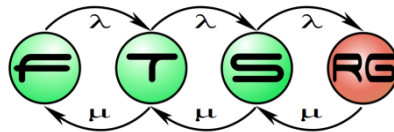


Eclipse Rich AJAX Platform



Tartalom

- RAP bevezetés
- RAP és RCP
- RAP-alapú UI design
- A RAP lehetőségei és korlátai
- A RAP illesztése
 - RAP és OSGi
 - RAP és J2EE

Napjaink tendenciái

Asztali alkalmazások	Webalkalmazások
Operációs rendszer-integráció	Web 2.0 nyílt szabványai
Gazdag widgetkészlet	Nem szükséges telepítés
Biztonság és gyorsaság	Könnyű kollaboráció & kommunikáció

Mindkettőből a legjobbat...

Mi is a RAP?

- Komponens-orientált webes fejlesztői környezet
- SWT, JFace, Eclipse workbench alapokon
- Eclipse plug-in menedzsment
 - Függések automatikus kezelése
 - Kiterjeszthetőség (extension points)
 - Életciklus menedzsment
 - Moduláris UI (~mashup Web 2.0 terminológiával)

A RAP célja

- Rich Internet Application fejlesztés...
 - Java nyelven
 - A teljes Java API használatával (vö.: GWT)
 - Egy Java-alapú widget toolkit használatával
 - Komponens-orientált alkalmazási modell alapján (OSGi)
 - Modularizáltan (Eclipse extension points)
 - AJAX és egyéb web-specifikus technológiák kézi kódolása nélkül
- Miért jó?
 - „skill preservation” – meglévő Eclipse RCP fejlesztők „újrahasznosítása”
 - „single sourcing” – két legyet egy csapásra, azaz egyszerre asztali és webes kliens fejlesztése

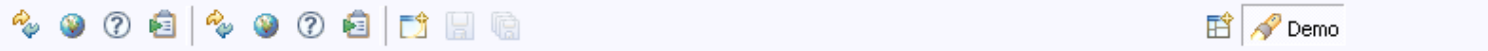
Hogy működik ez a weben?

- Az Eclipse/RCP kód...
 - 70-90%-a újrahasználható
 - A RAP az RCP egy (bővülő) részhalmazát valósítja meg
 - De: figyelni kell az inkompatibilis kódrészek jó elkülönítésére!

 - Fontos: fel kell készíteni az alkalmazást a több felhasználóra.

Workbench Demo

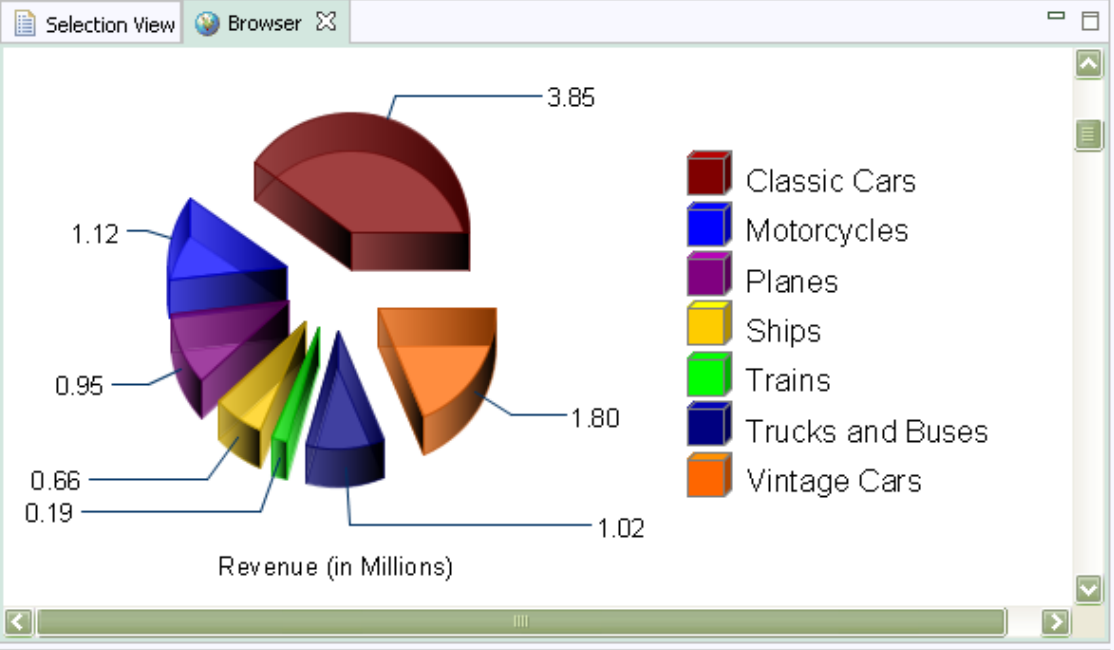
File Window Help



View I View II

Selection View Browser

- Root
 - Locate in browser view
 - EclipseCon location
 - Eclipse Foundation
 - Innoopract Inc
 - Parent 2
 - Child X - filter me!



View III

View IV View V

Column0	Column1	Column2	Column3	Column4	Column5	Column6
Item0-0	Item0-1	Item0-2	Item0-3	Item0-4	Item0-5	Item0-6
Item1-0	Item1-1	Item1-2	Item1-3	Item1-4	Item1-5	Item1-6
Item2-0	Item2-1	Item2-2	Item2-3	Item2-4	Item2-5	Item2-6
Item3-0	Item3-1	Item3-2	Item3-3	Item3-4	Item3-5	Item3-6

Nem feltétlenül kell az RCP kinézet...

The screenshot shows a web browser window with the address bar containing `http://127.0.0.1:2869/rap?startup=mail`. The page layout includes a blue header with the word "Banner" and three buttons labeled "click me 0", "click me 1", and "click me 2". Below the header is a green navigation bar with tabs for "In", "Out", "Over", "Under", and "Through", and two buttons for "Perspective 1" and "Perspective 2". The main content area is titled "Content" and "The content goes here". On the left, there is a list of four items: "This is Text for chapter 0.", "This is Text for chapter 1.", "This is Text for chapter 2.", and "This is Text for chapter 3.". On the right, there are three video thumbnails, each with a "rap it" button. The first video is titled "recep ivedik fragman" with ID "XD_7uUQKERO" and a description "Recep ?vedik'in beklenen filminin fragman?". The second video is titled "amazing guitar player" with ID "m3qMqK7h-BA" and a description "...one of those that will make you say...holy %\$#^". The third video is titled "Peanut and Jeff # 2" with ID "EpRW8jh8AqY" and a description "Ventriliquist". Each video thumbnail also includes a "2y" icon and a "youtube.com" link.

Itt is workbench-et látunk!

Nézetek, szerkesztők...

http://127.0.0.1:2869/rap?startup=mail

Banner

click me 0 | click me 1 | click me 2

In Out Over Under Through

me@this.com
Inbox
Drafts
Sent

other@aol.com

views

editors

+ Message

Subject: This is a message about the cool Eclipse RCP!

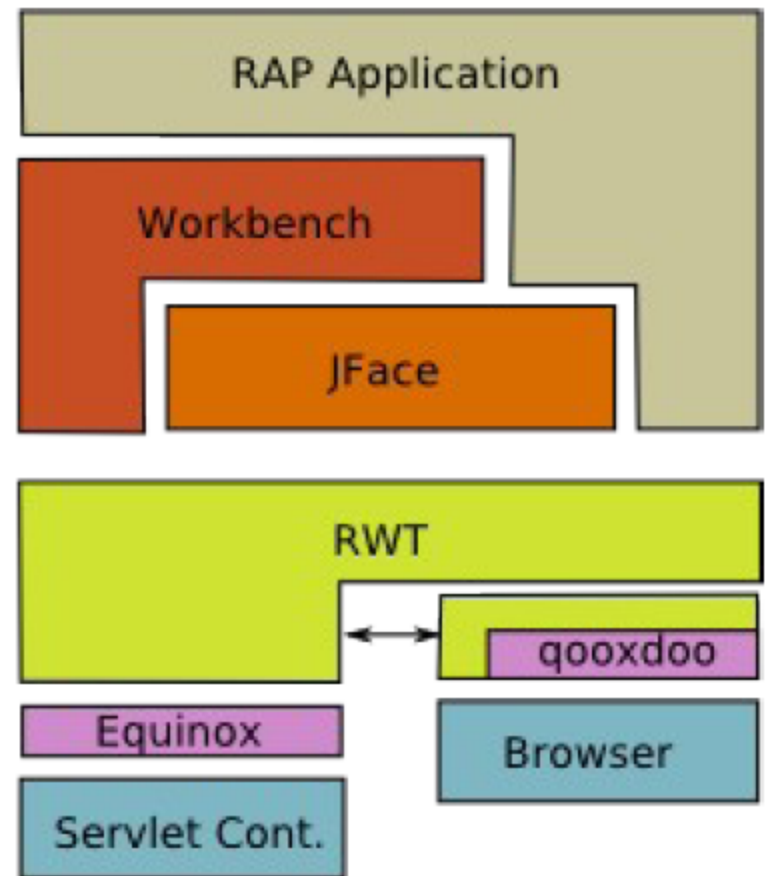
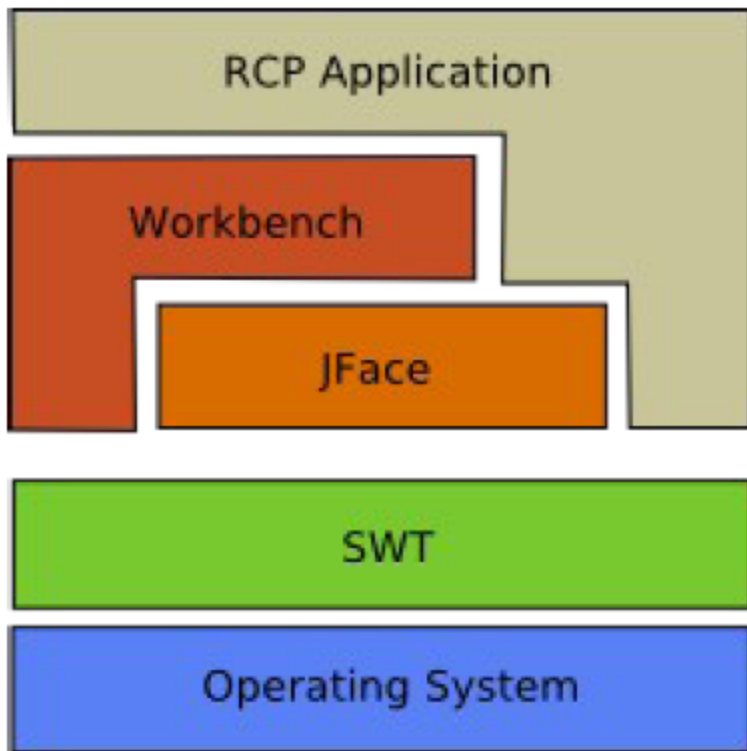
From: nicole@mail.org

Date: 10:34 am

This RCP Application was generated from the PDE Plug-in Project wizard.

- add a top-level menu and toolbar with actions
- create views that can't be closed and multiple instances of the same view
- perspectives with placeholders for new views
- use the default about dialog

Hogyan működik?



Működés

- RWT (RAP Widget Toolkit)
 - SWT web alapú implementációja
 - A qooxdoo nevű JavaScript toolkit fut a böngészőben
- Az összes, SWT-re épülő könyvtár (többé-kevésbé) működik
- Egy fontos különbség: multiuser működés
 - Application és egyéb OSGi kötetek megosztottak a különböző munkamenetek (session) között
 - → Session kezelésre külön oda kell figyelni!

Belső működés

- Szerver-centrikus
 - Minden logika a szerveren fut
- Renderelés
 - Futásidőben Javascript kód snippetek küldése a böngésző felé
- Minden felhasználói esemény egy állapotváltást eredményez
 - → amit a szerver értékel ki!
- A böngésző felé csak a „változás” (delta) megy vissza.

A fejlesztő szemszögéből...

```
org.eclipse.rap.demo  *DemoTreeViewPart.java X
+ * Copyright (c) 2002-2006 Innoopract Informationssysteme GmbH.
package org.eclipse.rap.demo;
+ import java.util.ArrayList;
public class DemoTreeViewPart extends ViewPart implements IDoubleClickListener {
    private TreeViewer viewer;
- public void createPartControl( final Composite parent ) {
    viewer = new TreeViewer( parent );
    viewer.setLabelProvider(new DecoratingLabelProvider(new LabelProvider(),
                                                         new LeafStarLabelDecorator()));
viewer.setC
viewer.
viewer.
getSite
)
- private f
- public
    if (
< >
Press 'Ctrl+Space' to show Template Proposals
```

The AbstractTreeViewer implementation of this method checks to ensure that the content provider is an ITreeContentProvider.

A fejlesztő szemszögéből...

```
org.eclipse.rap.demo  *DemoTreeViewPart.java x
+ * Copyright (c) 2002-2006 Innoopract Informationssysteme GmbH.

package org.eclipse.rap.demo;

+import java.util.ArrayList;

public class DemoTreeViewPart extends ViewPart implements IDoubleClickListener {

    private TreeViewer viewer;

- public void createPartControl( final Composite parent
    viewer = new TreeViewer( parent );
    viewer.setLabelProvider(new DecoratingLabelProvide
        new Le

viewer.setC
viewer.
viewer.
getSite
)

- private f
- public
    if (

Press 'Ctrl+Space' to show Template Proposals
```

Mintha csak RCP-t programoznánk:

- JDT alapszolgáltatások
- PDE stb.
- Alapesetben nem kell HTTP-vel, Javascripttel, HTML-lel és hasonlókkal törődni.

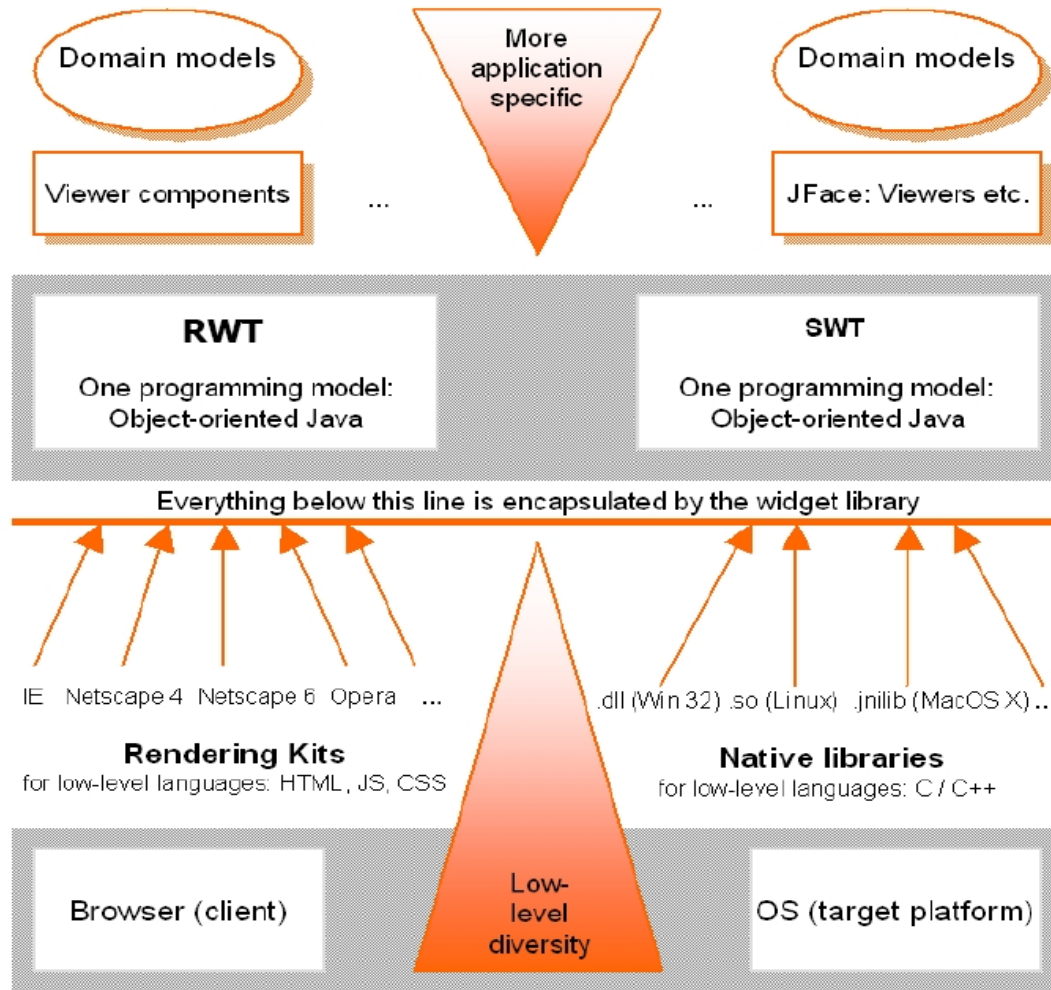
RAP alkalmazás futtatása, debuggolása

The screenshot displays the Eclipse IDE interface with the 'Run' dialog box open. The dialog is titled 'Create, manage, and run configurations' and is used to launch a RAP application. The 'Name' field is set to 'RMS'. The 'Framework' is set to 'Equinox', the 'Default Start level' is '4', and 'Default Auto-Start' is 'true'. The 'Bundles' section shows a list of bundles with their start levels. The 'Workspace' section is expanded, showing the following bundles:

Bundle	Start Level
org.eclipse.equinox.http.servletbridge (1.0.0)	default
org.eclipse.equinox.servletbridge (1.0.0.qualif	default
org.eclipse.rap.custom.viewers (1.0.0)	default
org.eclipse.rap.demo (1.0.0.20071008-1834)	
org.eclipse.rap.demo.gmaps (1.0.0)	
org.eclipse.rap.maildemo (1.0.0)	default
org.eclipse.rap.pde.runtime (3.3.0.v20070606	default
org.eclipse.rap.rms.data (1.0.0)	default
org.eclipse.rap.rms.ui (1.0.0)	default
org.eclipse.rap.rwt.custom (1.0.0)	default
org.eclipse.update.configurator (3.2.100.qual	default

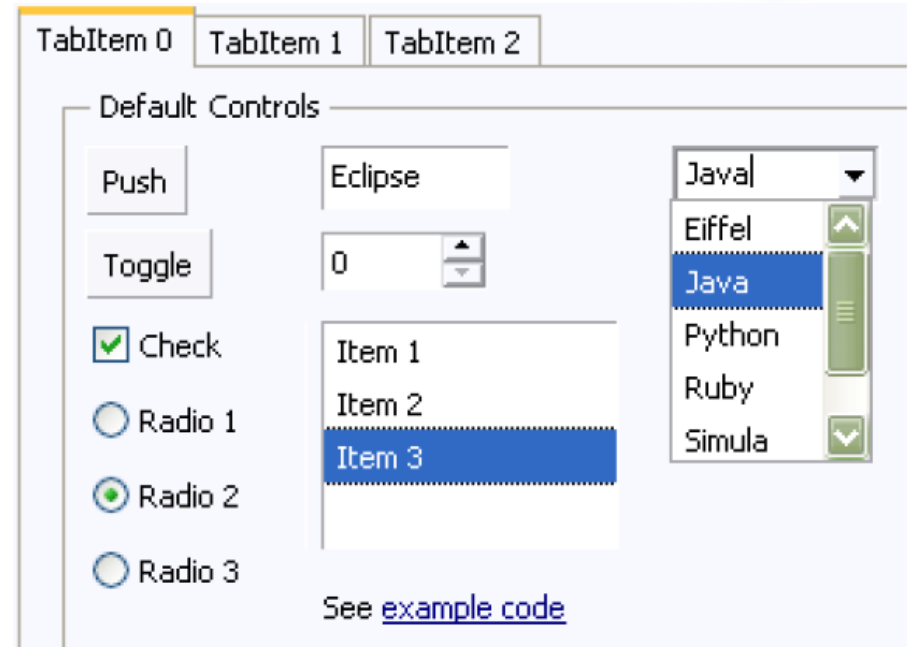
The 'Bundles' section also includes a 'Workspace' section with a tree view showing the project structure. The 'RMS' bundle is selected. The 'Include optional dependencies when computing required bundles' and 'Add new workspace bundles to this launch configuration automatically' checkboxes are checked. The '35 out of 44 selected' status is shown at the bottom right of the bundles list.

RWT architektúra



RWT

- A legtöbb alap SWT widget megvan
- Fejlettebb workbench elemek is (CTabFolder, Cbanner)
- Browser widget ☺
- Tree, Table

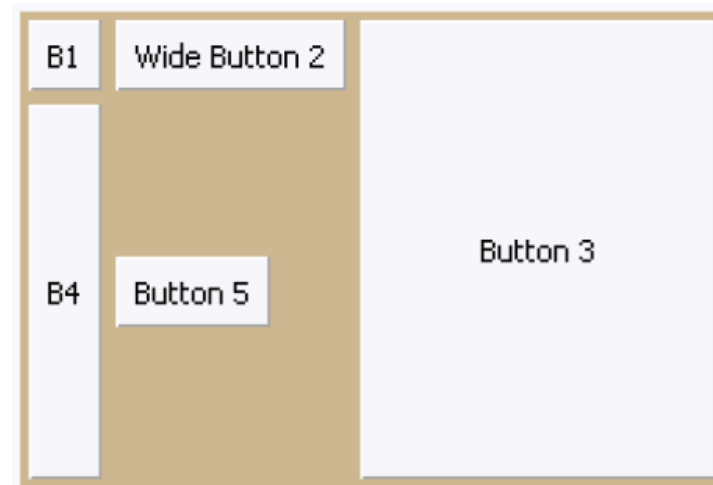


Column 1	Column 2
<input type="checkbox"/> Node_1	abc
<input checked="" type="checkbox"/> Node_2	jklmnop
<input type="checkbox"/> Subnode	jklmnop
<input type="checkbox"/> Node_3	jklmnop
<input type="checkbox"/> Node_4	jklmnop

Col 0	Col 1	Col 2	Col 3	Col 4
<input checked="" type="checkbox"/> Item0-0	Item0-1	Item0-2	Item0-3	Item0-4
<input type="checkbox"/> Item1-0	Item1-1	Item1-2	Item1-3	Item1-4
<input type="checkbox"/> Item2-0	Item2-1	Item2-2	Item2-3	Item2-4
<input type="checkbox"/> Item3-0	Item3-1	Item3-2	Item3-3	Item3-4
<input type="checkbox"/> Item4-0	Item4-1	Item4-2	Item4-3	Item4-4

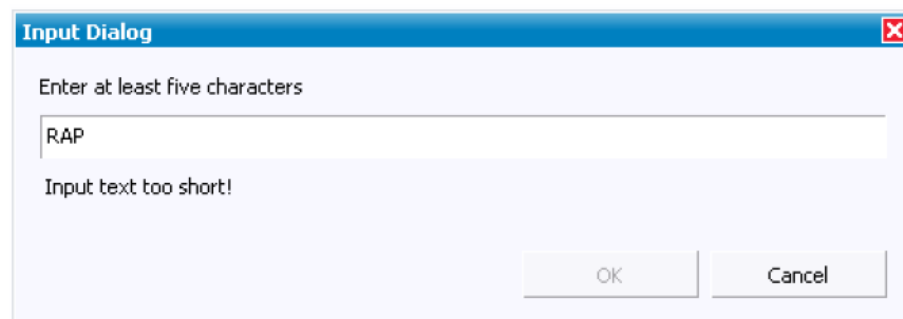
RWT Layoutok

- Minden layout megvan:
 - Grid, Row, Fill, Form, Stack
- Megvalósítás:
 - Általában az SWT kód másolatai = pontosan úgy működnek
- Működés
 - Minden layoutot a szerver oldal számol ki, a szerver pixelre pontosan tudja, hogy mit lát a felhasználó

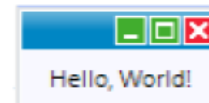


Eseménykezelés

- Típusos és általános listenereket támogat
- A legtöbb listener megvan, pl.:
 - SelectionListener
 - ControlListener
 - ShellListener
 - MenuListener
 - ...
- JFace data binding, validation működik



RAP Hello World



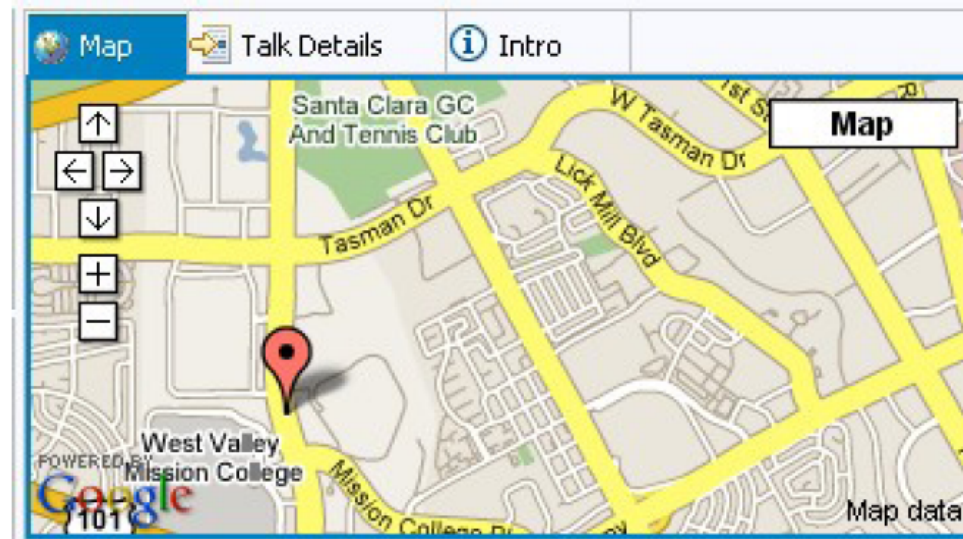
```
public class RWTHello implements IEntryPoint {

    public Display createUI() {
        Display result = new Display();
        final Shell shell = new Shell( result );
        RowLayout layout = new RowLayout();
        layout.justify = true;
        layout.pack = true;
        shell.setLayout( layout );
        Label label = new Label( shell, SWT.CENTER );
        label.setText( "Hello, World!" );
        shell.pack();
        shell.open();
        return result;
    }
}

<extension
    id="org.eclipse.rap.demo.demointerpoint"
    point="org.eclipse.rap.ui.workbench.entrypoint">
    <entrypoint
        class="org.eclipse.rap.tutorial.HelloWorld"
        parameter="default"/>
</extension>
```

Bővíthetőség (custom widgets)

- Lehetőség adott, mélyebb tudást igényel...
 - Komponens fejlesztő: Javascript, qooxdoo, RAP
 - Alkalmazás fejlesztő: csak Java API
- Példák
 - Google Maps integration: RAP Help
 - Lesz róla szó 😊

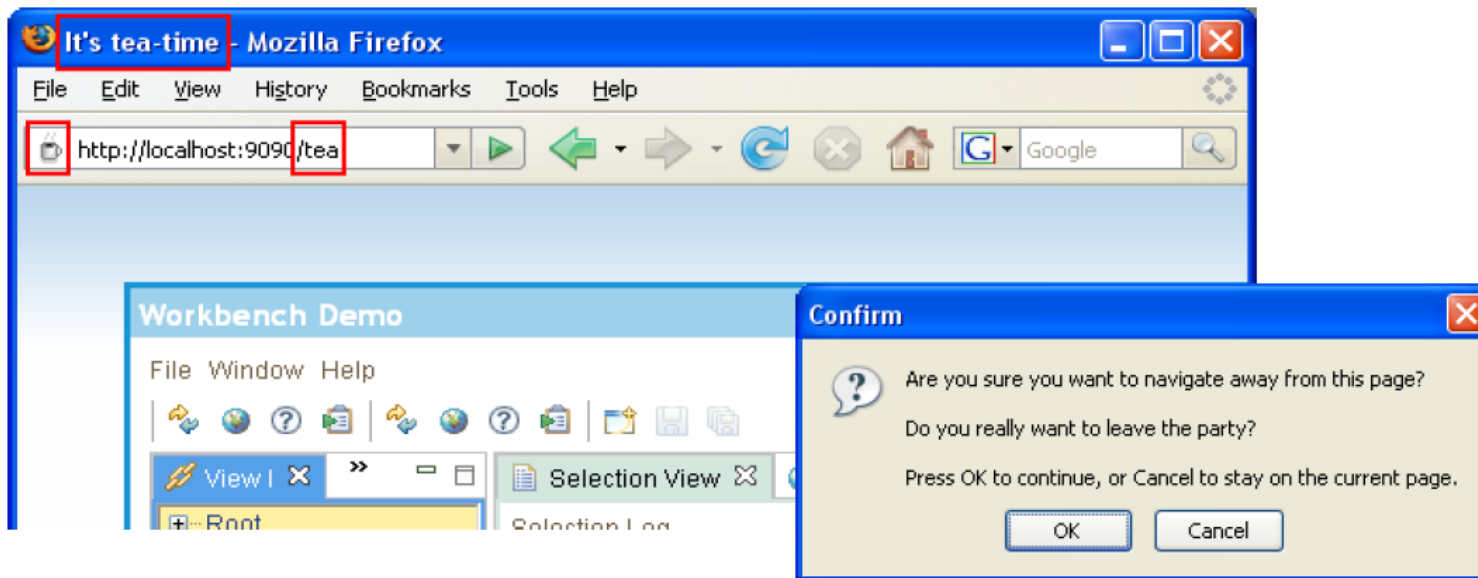


RWT testreszabása (témák)

- Cél: az alkalmazásunk megjelenése egyéni legyen
- Lehetőségek
 - A widgetek bizonyos tulajdonságait állíthatjuk (RAP 1.2 újdonság: gyakorlatilag mindent, CSS-hez nagyon hasonló kóddal)
 - Méret, színek, keretek, betűtípusok, díszítő elemek (képek)
 - A témákat a custom widgetek is használhatják (de erre fel kell őket készíteni)



Branding



```
<extension
  point="org.eclipse.rap.ui.branding">
  <branding
    id="org.eclipse.rap.demo.branding1"
    servletName="tea"
    defaultEntrypointId="org.eclipse.rap.demo.entrypoint1"
    themeId="org.eclipse.rap.demo.alttheme"
    title="It's tea-time"
    favicon="icons/favicon2.ico"
    body="body.html"
    exitConfirmation="Do you really want to leave the party?">
  </branding>
</extension>
```

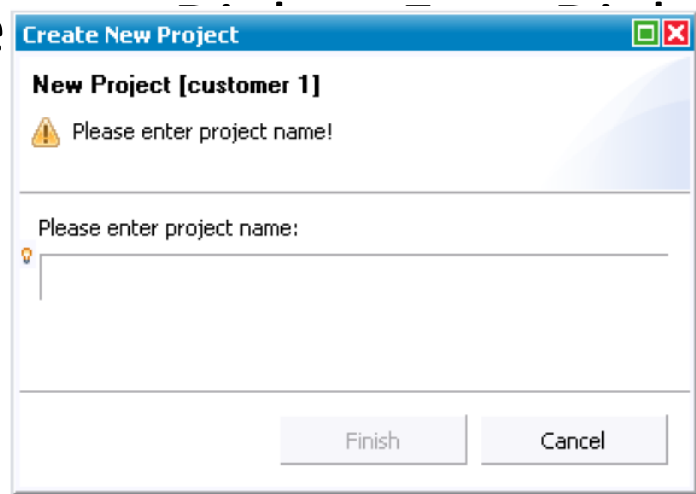

RAP és JFace

- JFace viewer framework működik
 - TableView, TreeViewer
 - Provider, Sorter, Filter, Decorator
 - RAP 1.2 újdonság: Cell Editor is működik

- Dialógusablakok

- Standard dialógusablakok (Me g)
- Modális ablakok
- Saját dialógusok

- Varázslók



RAP és a workbench

The screenshot displays the RAP Showcase application window, which is a web-based IDE. The interface is annotated with several colored boxes identifying key components:

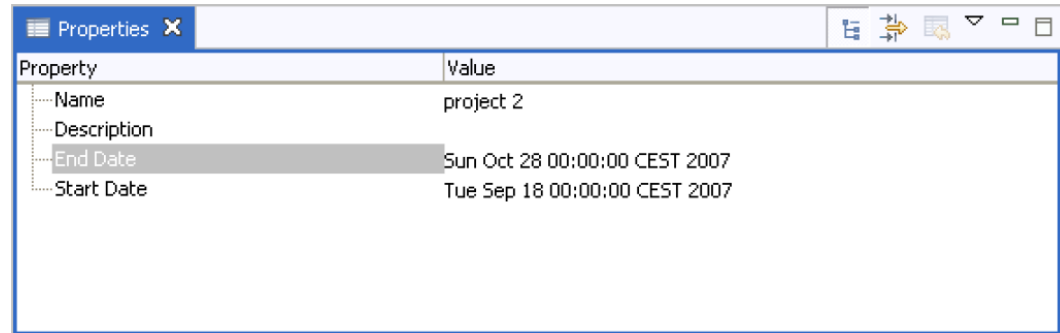
- Menu Bar:** Located at the top left, containing 'User' and 'Help' menus.
- Workbench Window:** The main application area, outlined in blue, containing a map and a contact list.
- Workbench Parts:** A red box highlighting the top navigation area with 'Map', 'Talk Details', and 'Intro' tabs.
- Workbench Page:** A purple box highlighting the main content area, which includes a map of Karlsruhe, Germany, and a contact list on the left.
- Cool Bar:** A yellow box highlighting the left sidebar containing a contact list for 'Me: Appel Frank'.
- Dialog:** A green box highlighting a 'Change User Data' dialog box. It contains a message: 'Please correct the input data which is marked red.' and fields for 'Firstname: Frank', 'Lastname: Appel', 'Street:', 'City: Karlsruhe', 'Country: Deutschland', 'Username: fa', and 'Password: ..'. A red error message at the bottom states: 'The password and its confirmation are not the same.'.

The map shows Karlsruhe, Deutschland, with various streets and landmarks labeled. The contact list on the left includes entries like 'B Jordi', 'Appel Renate', and 'Sternberg Ralf'.

RAP és a workbench

■ View

- Teljes támogatás
- Outline, Properties



Property	Value
Name	project 2
Description	
End Date	Sun Oct 28 00:00:00 CEST 2007
Start Date	Tue Sep 18 00:00:00 CEST 2007

■ Editor

- Multi-page editorok működnek

■ Perspective

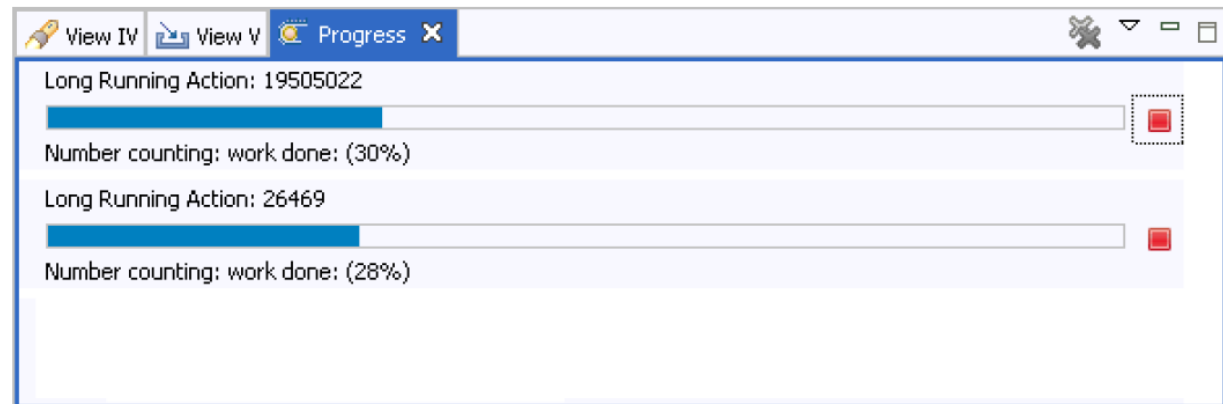
- Beépített perspective switcher

■ Action Framework (hamarosan deprecated)

■ Command Framework

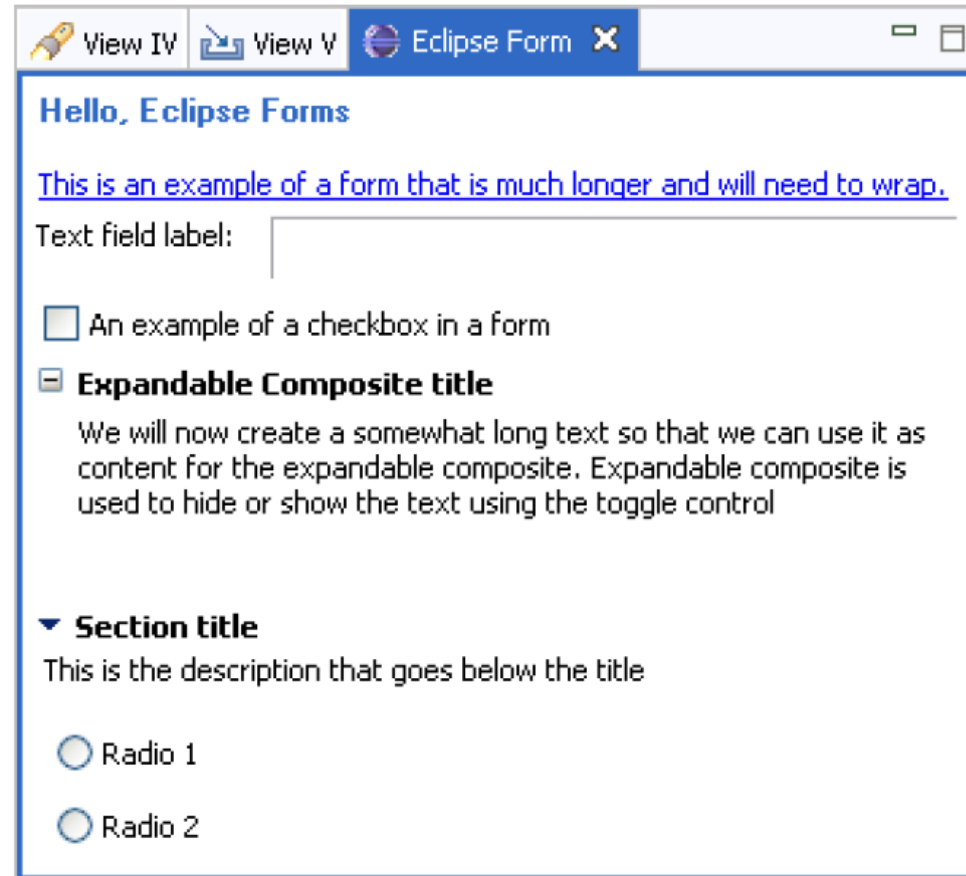
Workbench jobok, taskok

- RWT támogatja a Job API-t
- „Run in background” működik
- Progress Monitorok mennek



Forms, Data binding

- Eclipse 3.3 Data binding működik
- Eclipse UI Forms
 - Kezdetleges támogatás (RAP 1.1)
 - RAP 1.2 újdonság: Forms API nagy része elérhető



Teljesítmény, skálázhatóság

- Jellemző: per-session memória terhelési profil
- Fejlesztők jelentős erőfeszítéseket tettek az optimalizáció érdekében
- Néhány publikált benchmark létezik
 - Nem teljesen világos az összkép...
- Esettanulmány: Yoxos on Demand: egy „nagy” rendszer
- <http://ondemand.yoxos.com/geteclipse/start>

RAP Deployment

- A RAP alkalmazás telepíthető...
 - JEE Server konténerbe (WAR)
 - Servlet Bridge
 - OSGi standalone konténerbe (default)
 - Web kiszolgáló: OSGi plug-in (Jetty)

Ki fejleszti?

- Innopract
 - Fő fejlesztő
 - RAP/Eclipse konzultáns
 - Yoxos mögött ők állnak
- CAS
 - Qooxdoo fejlesztők
 - CRM, SME fejlesztésekkel foglalkoznak
- 1and1
 - Nagy nemzetközi ISP

RAP – Honnan?

- <http://www.eclipse.org/rap/>
- <http://rapblog.innoopract.com/>
- <http://www.ibm.com/developerworks/library/os-eclipse-richajax1/>
- <http://www.infoq.com/news/2008/06/eclipse-ganymede-rap;jsessionid=A0CC55BBF7B9C7CB44D51CD26D6FB6EE>

RAP és RCP együttes használata: single sourcing a gyakorlatban

Tartalom

- RAP és RCP közelebbről
- Konceptcionális különbségek
 - Session, application context
 - Kritikus erőforrások
- RCP alkalmazások migrálása
- Együttes fejlesztés: single sourcing
- RAP branding

Fő különbségek

RCP	RAP
Fájlrendszer	Fájlfeltöltés
Grafikus kontextus	Nincs rajzolási lehetőség
Egy felhasználó	Több felhasználó



Sessionkezelés

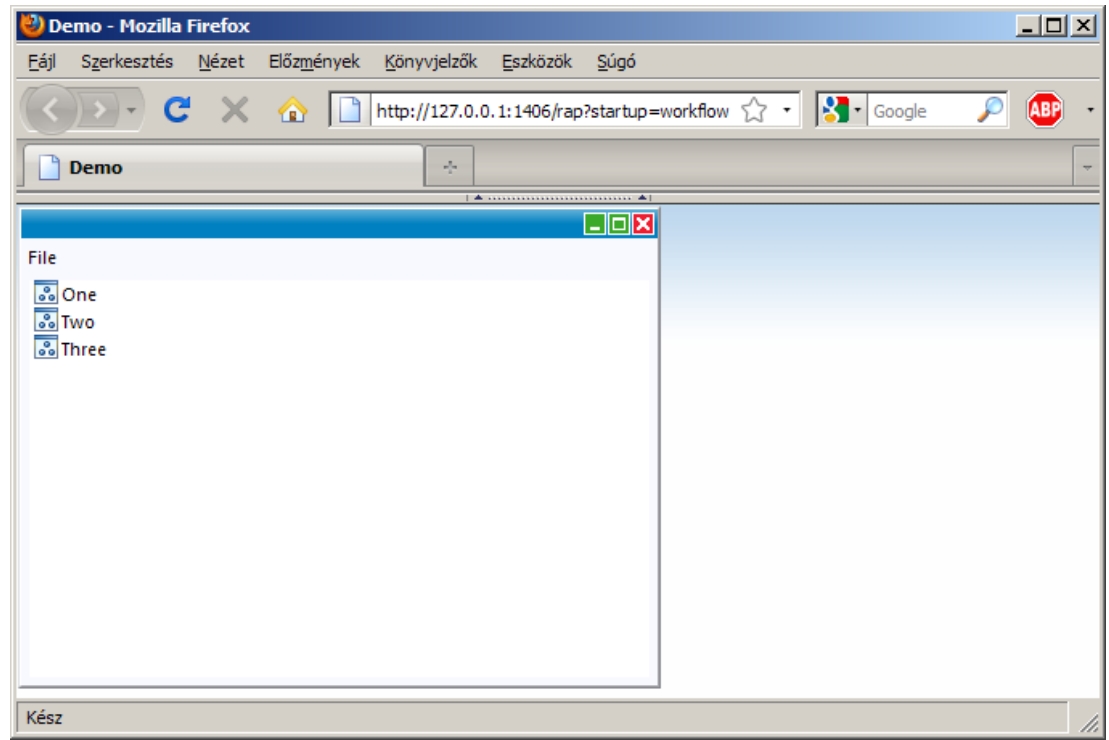
- Több felhasználós jelleg => gondoskodni kell a sessionkezelésről
- Nem jó a static/singleton állapot!
- Megoldás: RWT sessionkezelés
- `RWT.getSessionStore()`

RWT sessionkezelés

- Session id lekérése
 - `RWT.getSessionStore().getId();`
- Felhasználó-specifikus adat tárolása a sessionben
 - `RWT.getSessionStore().getHttpSession().put(key, value);`
- Értesítés kérése, amikor a session megszűnik
 - `RWT.getSessionStore().addSessionStoreListener(myListener);`
- A `HttpServletRequest` objektum elérése
 - `RWT.getRequest();`

Megjelenés

- Eltérő UI paradigmák
- „Ablak az ablakban”
- Zavaró
- Be lehet zárni



Megjelenés

■ Megoldás:

```
@Override
public void preWindowOpen() {
    IWorkbenchWindowConfigurer configurer =
        getWindowConfigurer();
    configurer.setShellStyle(SWT.NO_TRIM);
}
```

Keret
eltüntetése

```
@Override
public void postWindowOpen() {
    final IWorkbenchWindow window =
        getWindowConfigurer().getWindow();
    Shell shell = window.getShell();
    shell.setMaximized(true);
}
```

Maximalizálás

Single sourcing

- Cél: azonos funkcionalitású RCP és RAP alkalmazás fejlesztése
 - minimális költséggel
 - lehetőleg duplikáció nélkül
 - közös kódbázisból
- Eredmény: mindkettőből a legjobb
- Specifikus részek elkülönítése
- Két target platform váltogatása
- Megoldás: plugin fragment

Mi az a fragment?

- Nem önálló – egy host pluginen belül létezik
- Nincs Activatora
- Automatikusan látja a host plugin belsejét
- (fragment.xml a plugin.xml helyett)
- Tipikus felhasználási módok:
 - többnyelvűség
 - több platform támogatása



Single sourcing RAP fragmenttel

- Host plugin: RCP
 - org.eclipse.ui függőség opcionális
 - org.eclipse.rap.ui függőség opcionális
- Fragment: RAP
 - Minden RAP-specifikus dolog ide jön
 - Entry point
 - Branding

API-különbségek elfedése

- A közös ősinterface, ezt kell használni

```
public abstract class AboutActionHelper {  
  
    private static final AboutActionHelper IMPL;  
  
    static {  
        Class clazz = AboutActionHelper.class;  
        Object newInstance = ImplementationLoader.newInstance( clazz );  
        IMPL = ( AboutActionHelper )newInstance;  
    }  
  
    public static IWorkbenchAction create( IWorkbenchWindow window ) {  
        return IMPL.createInternal( window );  
    }  
  
    protected abstract IWorkbenchAction createInternal( IWorkbenchWindow  
        window );  
  
}
```

API-különbségek elfedése

■ Implementáció betöltése

```
public final class ImplementationLoader {  
  
    public static Object newInstance( Class type ) {  
        String name = type.getName();  
        Object result = null;  
        ClassLoader loader = type.getClassLoader();  
        try {  
            Class clazz = loader.loadClass( name + "Impl" );  
            result = clazz.newInstance();  
        } catch( Throwable t ) {  
            // ...  
        }  
        return result;  
    }  
  
    private ImplementationLoader() {}  
  
}
```

API-különbségek elfedése

■ RAP-specifikus implementáció

```
public class AboutActionHelperImpl extends AboutActionHelper {
    private static final class AboutAction extends Action
        implements IWorkbenchAction {
        private AboutAction() {
            setText( "About" );
            setId( "aboutAction" );
        }
        public void run() {
            MessageDialog.openInformation( null, "About", "About" );
        }
        public void dispose() {}
    }
    protected IWorkbenchAction createInternal( IWorkbenchWindow
        win ) {
        return new AboutAction();
    }
}
```

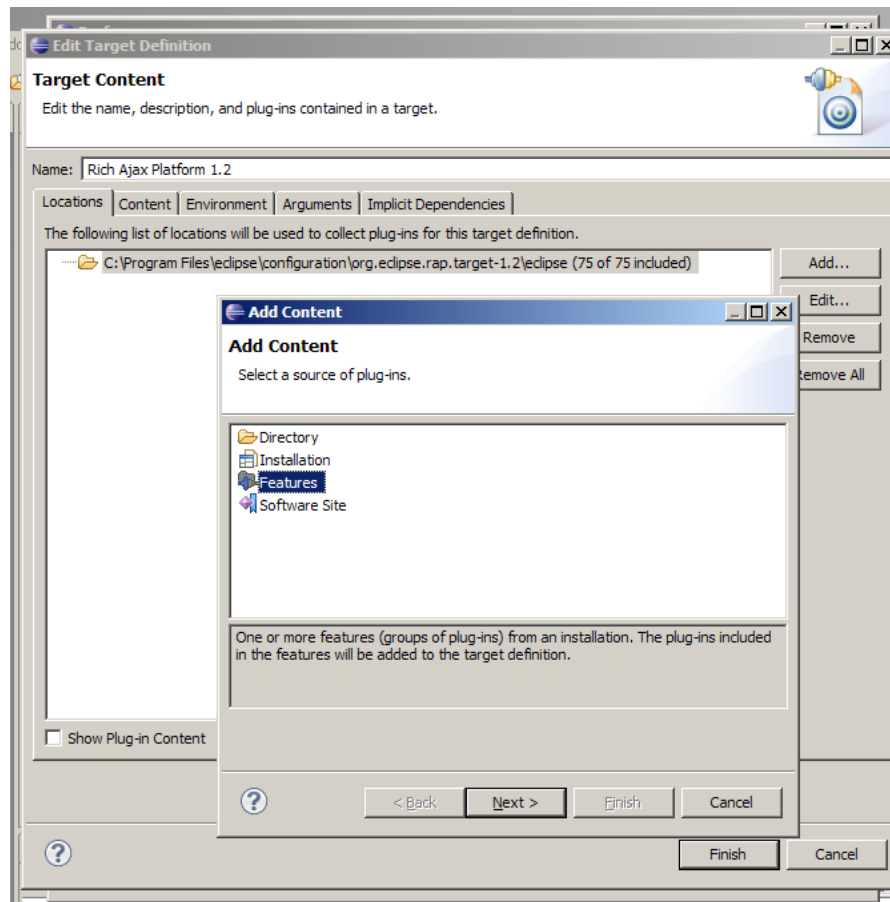
API-különbségek elfedése

■ RCP-specifikus implementáció

```
public class AboutActionHelperImpl extends AboutActionHelper {  
    protected IWorkbenchAction createInternal( IWorkbenchWindow  
        win ) {  
        return ActionFactory.ABOUT.create(win);  
    }  
}
```

Target platform-kezelés

- Nem szükséges külön workspace, de lehet
- Használt pluginek betehetőek a target platformba



RAP branding

- Az alkalmazás testreszabása
 - RCP-től eltérő módon
- `org.eclipse.rap.ui.branding` extension point

RAP branding részletesen

- `serviceName`
 - Az URL-ben jelenik meg (default: rap)
- `defaultEntrypointId`
 - Alkalmazásunk belépési pontja
- `themeld`
 - A téma azonosítója (ld. később)
- `title`
 - A böngésző fejlécében jelenik meg

RAP branding részletesen – folyt.

- **favicon**
 - Az oldal ikonja
- **body**
 - Extra HTML kód a weboldal <body> részébe
- **exitConfirmationClass**
 - Az alkalmazás/böngészőfül bezárásakor kérdést megjelenítő Java osztály
- **additionalHeaders gyermekelemek**
 - Extra HTML kódrészletek a weboldal <head> részébe

RWT theming

- Widgetek kinézetének megadása
- CSS-hez hasonló szintaxisú fájlban

```
* {  
  font: 12px Tahoma;  
}
```



Minden elem
betűtípusának
megváltoztatása

```
TableItem:even {  
  background-color: #eef7ff;  
}
```



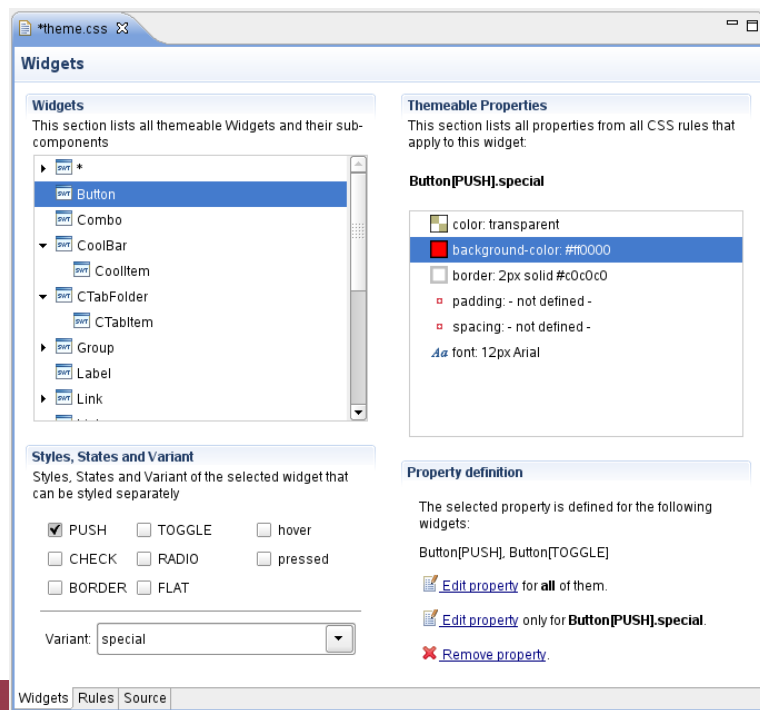
Páros
táblázatsorok
hátterszínének
megváltoztatása

RWT theming

- Selectorok:
 - Widget (Button, List...)
 - Style (PUSH, TOGGLE, CHECK, RADIO, BORDER, FLAT)
 - Állapot (hover, pressed, disabled, selected)
- Tulajdonságok:
 - color
 - padding
 - border
 - font
- Referencia:
 - <http://help.eclipse.org/galileo/topic/org.eclipse.rap.help/help/html/reference/theming/index.html>

RWT Theme Editor

- Készül a RAP Theme Editor
- Form alapú szerkesztő
- Előnézet
- http://wiki.eclipse.org/RAP_Theme_Editor



Honosítás

- A felhasználó nyelve requestek között változhat
 - Eclipse által generált statikus Messages osztály nem jó
- Megoldás: RWT.NLS (RWT-specifikus)
 - nem lehet használni single sourcinggal
- Ugyanúgy properties-fájl alapú
 - Lehet UTF-8!

Honosítás – megvalósítás

```
public class Messages {  
    private static final String BUNDLE_NAME =  
        "org.eclipse.rap.helloworld.messages"; //$NON-  
        NLS-1$  
  
    private Messages() {} // példányosítás tiltása  
  
    public static Messages get() { // példány visszaadása  
        Class clazz = Messages.class;  
        return (Messages)RWT.NLS.getUTF8Encoded  
        (  
            BUNDLE_NAME, clazz);  
        }  
  
    // Üzenetkonstansok  
    public String HelloWorldView_Message;
```


Honosítás – megvalósítás

■ Nyelvi fájlok:

- org.eclipse.rap.helloworld.messages.properties
 - HelloWorldView_Message = Hello RAP World!
- org.eclipse.rap.helloworld.messages_hu.properties
 - HelloWorldView_Message = Helló RAP világ!

■ Használat:

- `label.setText(Messages.get()
.HelloWorldView_Message);`

Saját RWT widget létrehozása – áttekintés

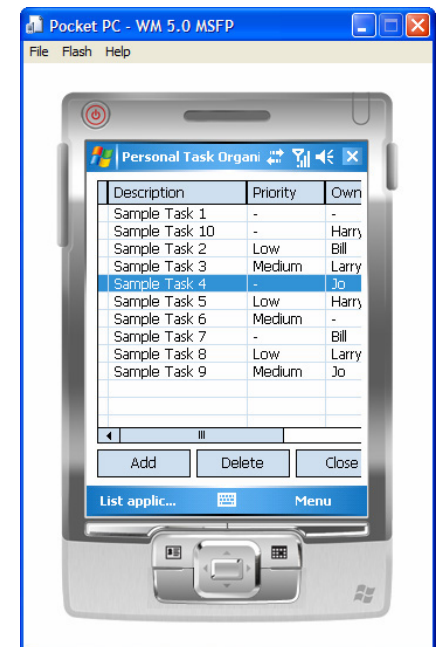
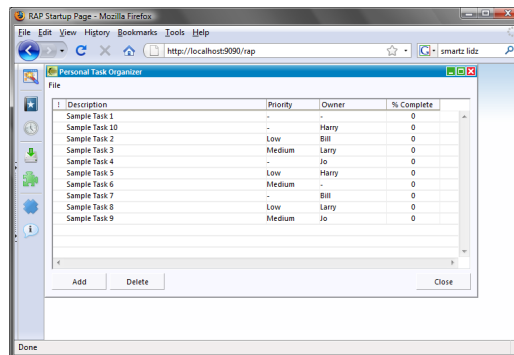
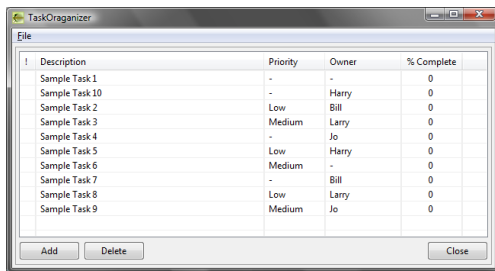
- Lépések
 - Java: Widget-interfész létrehozása
 - JavaScript: Widget létrehozása qooxdoo framework segítségével
 - A kliens- és szerveroldal összekötése
 - JavaScript beregisztrálása plugin.xml-ben
- Részletesen: <http://help.eclipse.org/galileo/topic/org.eclipse.rap.help/help/html/advanced/custom-widget.html>

Saját widget felkészítése themingre

- `org.eclipse.rap.ui.themeableWidgets` extension point
- Szükséges erőforrások:
 - Theme adapter:
`org.eclipse.rwt.theme.IControlThemeAdapter`
 - Theme definition: XML fájl
 - Appearance fragment: CSS template
- Részletesen: http://dev.eclipse.org/viewcvs/index.cgi/org.eclipse.rap/org.eclipse.rap.help/help/html/advanced/theming-custom.html?root=Technology_Project&view=co

Kitekintés: RCP-RAP-eRCP

- Három nagy platform



Kitekintés: RAP 1.3 újdonságok

- Theme contribution:
 - Nem kell teljesen új témát készíteni
 - Elég a különbségeket kezelni
- Kiterjesztési pont: `org.eclipse.rap.ui.themes`

Kitekintés: RAP 1.3 újdonságok

- Graphics Context:
 - SWT Canvas kezelés
 - Egyelőre csak részhalmaza a SWT opcióknak



Kitekintés: RAP 1.3 újdonságok

- EMF on RAP
 - EMF tree editor RAP felett is működik



Kitekintés: RAP 1.3 újdonások

- Egyéb apróságok
 - Böngésző előzmények kihasználása
 - DateTime választó widget
 - Widgetek egyedi témázása
 - Drag and drop
 - Eclipse Help támogatás
 - Cheat Sheet támogatás