

Eclipse RCP Part IV

Automotive Financial Services Insurance Life Science & Healthcare Public Sector
Telecommunications & Media Travel & Logistics Utilities Automotive Financial
Services Insurance Life Science & Healthcare Public Sector Telecommunications
& Media Travel & Logistics Utilities Automotive Financial Services Insurance
Life Science & Healthcare Public Sector Telecommunications & Media Travel
& Logistics Utilities Automotive Financial Services Insurance Life Science
Healthcare Public Sector Telecommunications & Media Travel & Logistics
Utilities Automotive Financial Services Life Science & Healthcare Public
Sector Telecommunications & Media Travel & Logistics Utilities Automotive
Financial Services Insurance Life Science & Healthcare Public Sector
Telecommunications & Media Travel & Logistics Utilities Automotive
Financial Services Insurance Life Science & Healthcare Telecommunications
& Media Travel & Logistics Utilities Automotive Financial Services Insurance
Life Science & Healthcare Public Sector Telecommunications & Media Travel &
Logistics Utilities Automotive Financial Services Insurance Life Science &
Healthcare Public Sector Telecommunications & Media Travel & Logistics Utilities
Automotive Financial Services Insurance Life Science & Healthcare Public Sector



.consulting .solutions .partnership



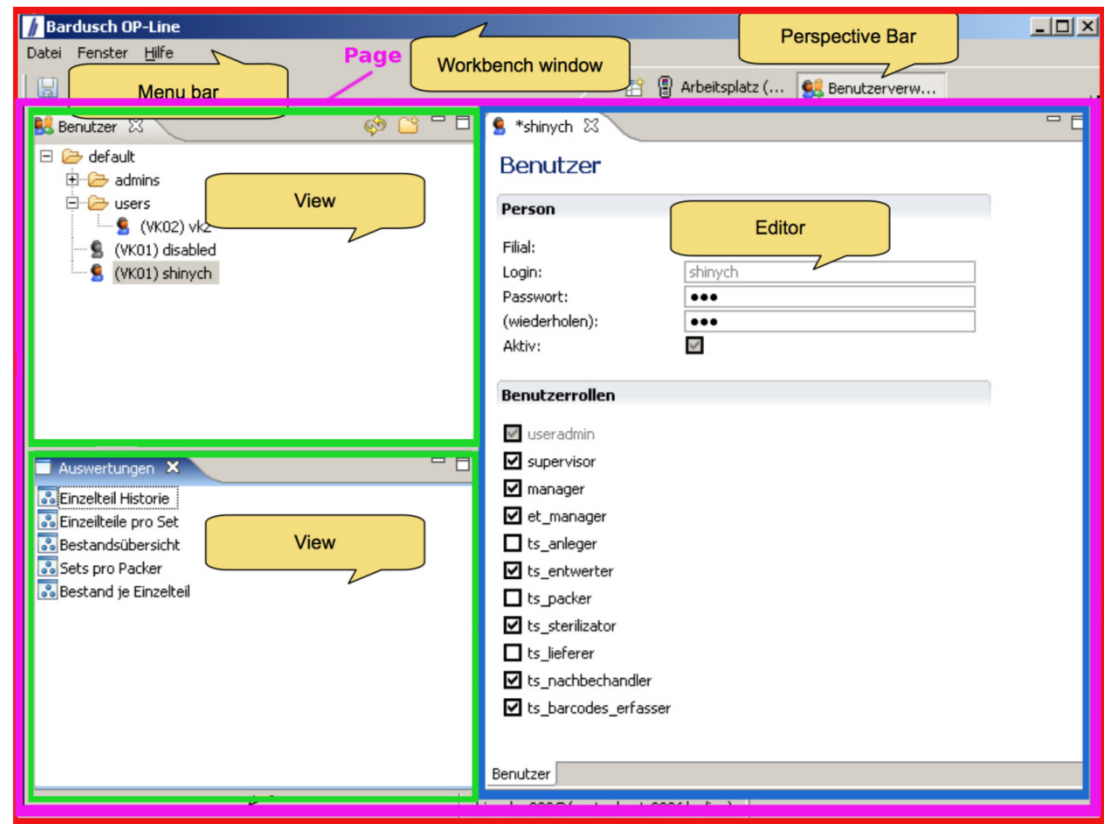
Objective



- Building your first RCP Application
- Understanding the Advisors
- Add a menu to your application

Workbench structure

- Workbench (1)
- Windows (0-N)
- Page (0-1)
- Perspectives (0-N)
- Views (0-N)
- Editors (0-N)

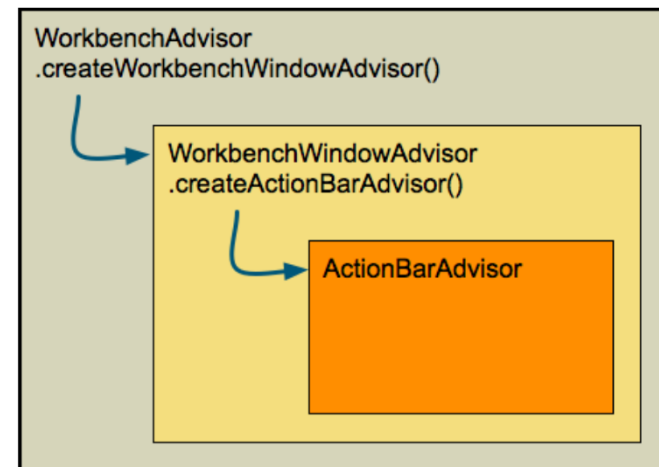


- An instance of IWorkbench is the entry point for most APIs
- There is only one instance of this interface within each application (Singleton):

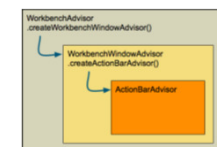
```
IWorkbench workbench =  
    PlatformUI.getWorkbench();
```

- Provides access to platform UI components:
 - Workbench windows
 - Progress service
 - Shared images
 - ...
- Provides several controlling methods:
 - close()
 - restart()
 - saveAllEditors()
 - showPerspective(id, window)
 - ...

-
- The screenshot shows the Eclipse IDE's Project Explorer. The 'src' folder is expanded, revealing the 'rcp' folder. Inside 'rcp', the following files are listed:
- Activator.java
 - Application.java
 - ActionBarAdvisor.java
 - ApplicationWorkbenchAdvisor.java
 - ApplicationWorkbenchWindowAdvisor.java
 - Perspective.java



- Strategy object for configuring the Workbench:
 - abstract String getInitialWindowPerspectiveId()
- Workbench lifecycle methods:
 - preStartup(), postStartup(), preShutdown(), postShutdown(), ...
- Memento methods:
 - saveState(IMemento), restoreState(IMemento)



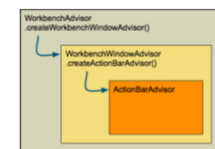
```
org.eclipse.ui/Memento
public interface IMemento {
    void putString(String key, String value);
    String getString(String key);

    void putString(String key, Integer value);
    Integer getString(String key);

    IMemento createChild(String type);
    IMemento[] getChildren(String type);
    //...
}
```

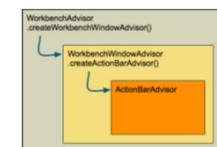
To enable the persistence of the workbench state override the following method of your WorkbenchAdvisor:



```
public void initialize(IWorkbenchConfigurer config) {
    config.setSaveAndRestore(true);
}
```

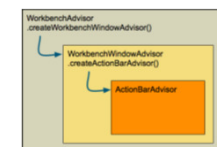



```
public void preWindowOpen() {  
    IWorkbenchWindowConfigurer configurer = getWindowConfigurer();  
    configurer.setInitialSize(new Point(400, 300));  
    configurer.setShowCoolBar(false);  
    configurer.setShowStatusLine(false);  
    configurer.setTitle("Hello RCP");  
    configurer.setShowPerspectiveBar(true);  
}
```

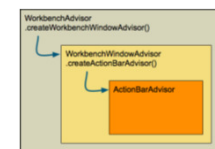
- Configures the Workbench
 - IWorkbenchWindowConfigurer
- Window lifecycle methods:
 - **preWindowOpen()**, postWindowCreate(), postWindowClose(), postWindowRestore(), ...
- Memento methods:
 - saveState(IMemento), restoreState(IMemento)



- Creates Workbench actions with the method:
 - `makeActions(IWorkbenchWindow)`
- Standard action factories may be useful: 
 - `org.eclipse.ui.actions.ActionFactory`
 - `org.eclipse.ui.actions.ContributionItemFactory`
- Fills menu bar, toolbar and status line with actions: 
 - `fillMenuBar(...)`, `fillCoolBar(...)`, `fillStatusLine()`
- Memento methods:
 - `saveState(IMemento)`, `restoreState(IMemento)`


























```
public class ApplicationActionBarAdvisor extends ActionBarAdvisor {  
  
    private IWorkbenchAction exitAction;  
  
    public ApplicationActionBarAdvisor(IActionBarConfigurer configurer) {  
        super(configurer);  
    }  
  
    protected void makeActions(IWorkbenchWindow window) {  
        exitAction = ActionFactory.QUIT.create(window);  
        register(exitAction);  
    }  
  
    protected void fillMenuBar(IMenuManager menuBar) {  
  
        MenuManager fileMenu = new MenuManager("&File",  
        IWorkbenchActionConstants.M_FILE);  
        fileMenu.add(exitAction);  
        menuBar.add(fileMenu);  
  
    }  
}
```



- Create a new RCP Application using the Wizard
- Play with the Advisors to achieve the desired look of your application
- Look which workbench actions can be created using ActionFactory and ContributionItemFactory.
- Create some of the following top-menu items:
 - File/New
 - File/Close
 - Window/Open new window
 - Window/Open Perspective
 - Window/Show View
 - Help/Help Contents
 - Help/About

- Für das Hilfe-System müssen folgende Plugins eingebunden werden:

 javax.servlet (0.0.0)	 org.eclipse.equinox.http.jetty (0.0.0)
 javax.servlet.jsp (0.0.0)	 org.eclipse.equinox.http.registry (0.0.0)
 org.apache.ant (0.0.0)	 org.eclipse.equinox.http.servlet (0.0.0)
 org.apache.commons.el (0.0.0)	 org.eclipse.equinox.jsp.jasper (0.0.0)
 org.apache.commons.logging (0.0.0)	 org.eclipse.equinox.jsp.jasper.registry (0.0.0)
 org.apache.jasper (0.0.0)	 org.eclipse.help.base (0.0.0)
 org.apache.lucene (0.0.0)	 org.eclipse.help.ui (0.0.0)
 org.apache.lucene.analysis (0.0.0)	 org.eclipse.help.webapp (0.0.0)
 org.eclipse.ant.core (0.0.0)	 org.eclipse.osgi.services (0.0.0)
 org.eclipse.core.runtime.compatibility (0.0.0)	 org.eclipse.ui.forms (0.0.0)
 org.eclipse.core.runtime.compatibility.registry (0.0.0)	 org.mortbay.jetty (0.0.0)
 org.eclipse.core.variables (0.0.0)	

Vielen Dank für Ihre Aufmerksamkeit



.consulting .solutions .partnership

