Implementing custom renderers
RCP 3.x rendering

- Application model tightly coupled to rendering layer
- SWT only
- Customized by Presentation API
- Limited and difficult modification
- Not portable
RCP 4.x rendering

- Well defined, extandable Application Model decoupled from its presentation
- Allows to use any widget toolkit (e.g. Swing, JavaFX)
- Simplified customization of default SWT renderers
- Easy to add and render custom model elements
public interface IPresentationEngine {

    public Object createGui(MUIElement element, Object parentWidget, IEclipseContext parentContext);

    public Object createGui(MUIElement element);

    public void removeGui(MUIElement element);

    public void focusGui(MUIElement element);

    public Object run(MApplicationElement uiRoot, IEclipseContext appContext);

    public void stop();

}
Custom Presentation Engine

<extension
  id="product"
  point="org.eclipse.core.runtime.products">
  <product
    application="org.eclipsecon.E4Application"
    name="E4 Application">
    <property
      name="presentationURI"
      value="bundleclass://org.eclipsecon/org.eclipsecon.MyPresentationEngine">
    </property>
    ...
  </product>
</extension>
Custom Presentation Engine
Standard SWT Presentation Engine

- implemented by PartRenderingEngine
  (org.eclipse.e4.ui.internal.workbench.swt package)
- CSS support
- easy to customize
public interface IRendererFactory {

    public AbstractPartRenderer getRenderer(MUIElement uiElement, Object parent);

}
public interface IRendererFactory {

    public AbstractPartRenderer getRenderer(MUIElement uiElement, Object parent);

}
<extension
    id="product"
    point="org.eclipse.core.runtime.products">
  <product
    application="org.eclipsecon.E4Application"
    name="E4 Application">
    <property
      name="rendererFactoryUri"
      value="bundleclass://org.eclipsecon/org.eclipsecon.MyRendererFactory">
    ...
  </product>
</extension>
Overriding initial element's style

use `styleOverride` key in a model element's persisted state

used by WBWRenderer and StackRenderer
Custom renderer for a specific part

use Custom Renderer key in a model element's persisted state

equivalent value:

bundleclass://mybundle/renderers.CustomPartRenderer
See more...

- www.vogella.com
- toedter.com
- tomsondrev.bestsolution.at
Exercise 1
### Exercise 1

![Contact Management Interface]

**File**
- First Name: Chris
- Last Name: Aniszczyk
- First Name: Boris
- Last Name: Bokowski
- First Name: Peter
- Last Name: Friese
- First Name: Kevin
- Last Name: McGuire
- First Name: Tom
- Last Name: Schindl
- First Name: Hallvard
- Last Name: Traetteberg
- First Name: Kai
- Last Name: Tödter
- First Name: Lars
- Last Name: Vogel
- First Name: Yves
- Last Name: YANG
- First Name: Angelo
- Last Name: ZERR

**Details**
- **General**
  - Title:
  - Name:
  - Company:
  - Job Title:
  - Note:
- **Business Address**
  - Street:
  - City:
  - ZIP:
  - State/Prov:
  - Country:
- **Business Phones**
  - Phone:
  - Mobile:
- **Business Internet**
Exercise 2
Exercise 3

Image of a contact management application with a list of contacts and detailed information for each contact.
Exercise 4

This RCP Application was generated from the PDE Plug-in Project wizard. This sample shows how to:
- add a 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
- perspectives with placeholders for new views
- use the default about dialog