Style It!
The Eclipse 4 Styling Tutorial

Kai Tödter, Siemens AG
Boris Bokowski, IBM
Bodgan Gheorghe, IBM

Who is Kai?

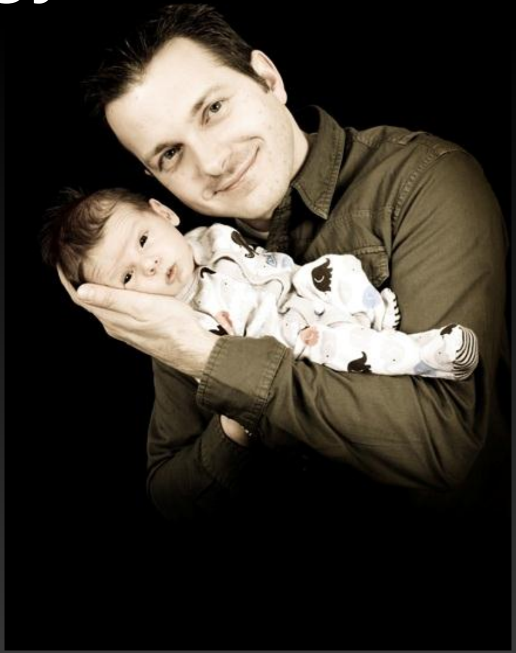
- Software Architect/Engineer at Siemens Corporate Technology
- Eclipse RCP expert and OSGi enthusiast
- Open Source advocate
- Committer at e4 and Platform UI
- E-mail: kai.toedter@siemens.com
- Twitter: twitter.com/kaitoedter
- Blog: toedter.com/blog

Who is Boris?

- Eclipse Platform UI Project Lead
- Committer on e4, Orion and platform
- From IBM Ottawa
- Supporter of cool new technology
- Email: boris_bokowski@ca.ibm.com

Who is Bogdan?

- Committer on e4, SWT and platform
- Working on CSS engine and styling
- From IBM Ottawa
- Supporter of cool new technology
- Email: gheorghe@ca.ibm.com



UI Styling

- In Eclipse 3.x, UI styling can be done using
  - The Presentation API
  - Custom Widgets
- These mechanisms are very limited
- It is not possible to implement a specific UI design, created by a designer
- e4 provides a CSS based UI styling that addresses many of the above issues

A Brief History of CSS in e4

- The goal – provide an easier way to customize the look and feel of an e4 app
- Make use of an established format to describe the styling
- Made a debut in the 0.9 e4 release
- Eclipse 4.0 SDK used CSS to style workbench

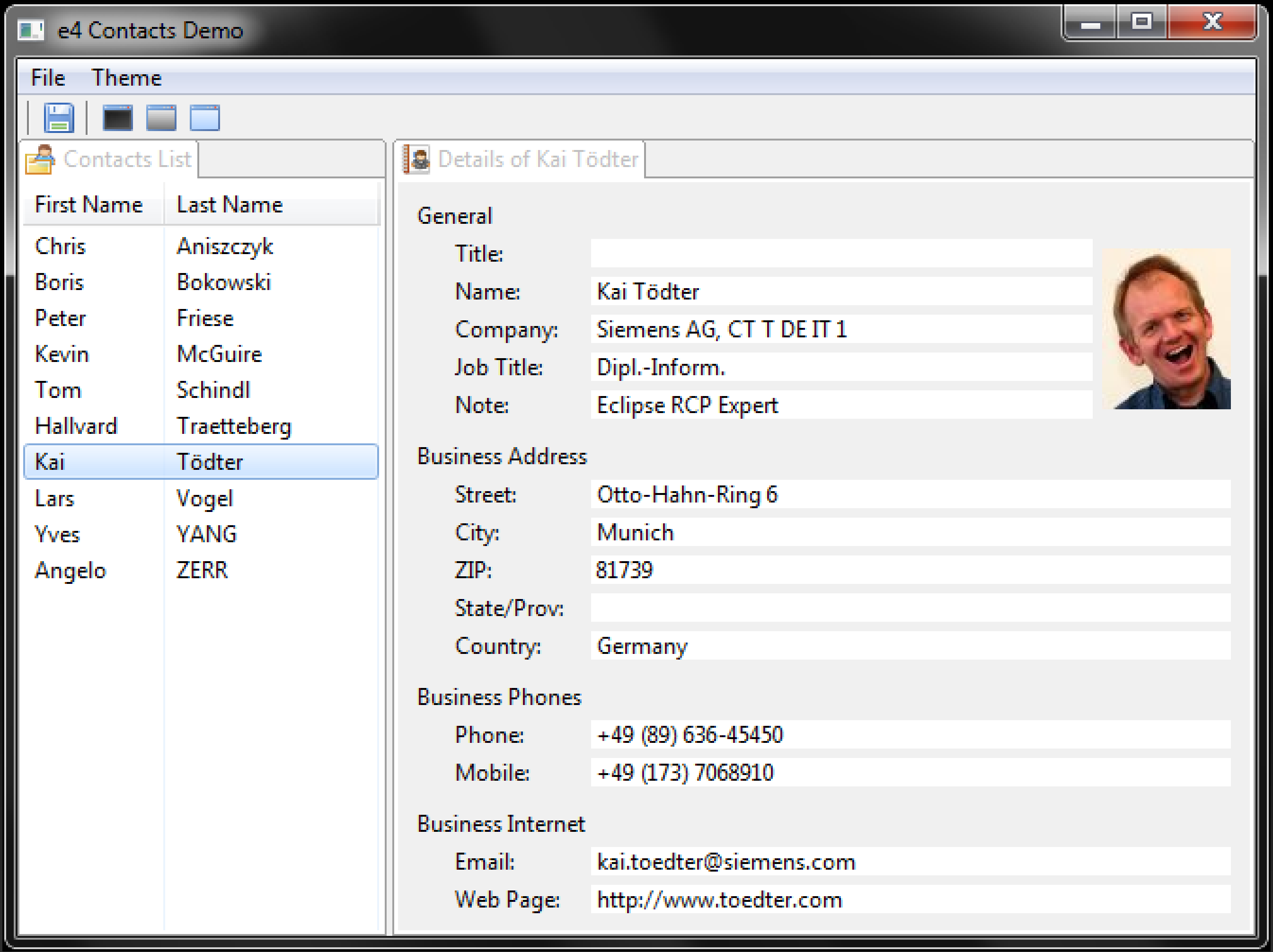
The Plan

- Styling an e4 RCP app
- CSS Overview
- CSS Engine internals
- Styling the e4 workbench
- Styling a 3.x RCP app
- Wrap Up

Styling an e4 RCP app

Picture from
http://www.sxc.hu/photo/1089931

Contacts Demo without CSS Styling



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

General

Title:	
Name:	Kai Tödter
Company:	Siemens AG, CT T DE IT 1
Job Title:	Dipl.-Inform.
Note:	Eclipse RCP Expert

Business Address

Street:	Otto-Hahn-Ring 6
City:	Munich
ZIP:	81739
State/Prov:	
Country:	Germany

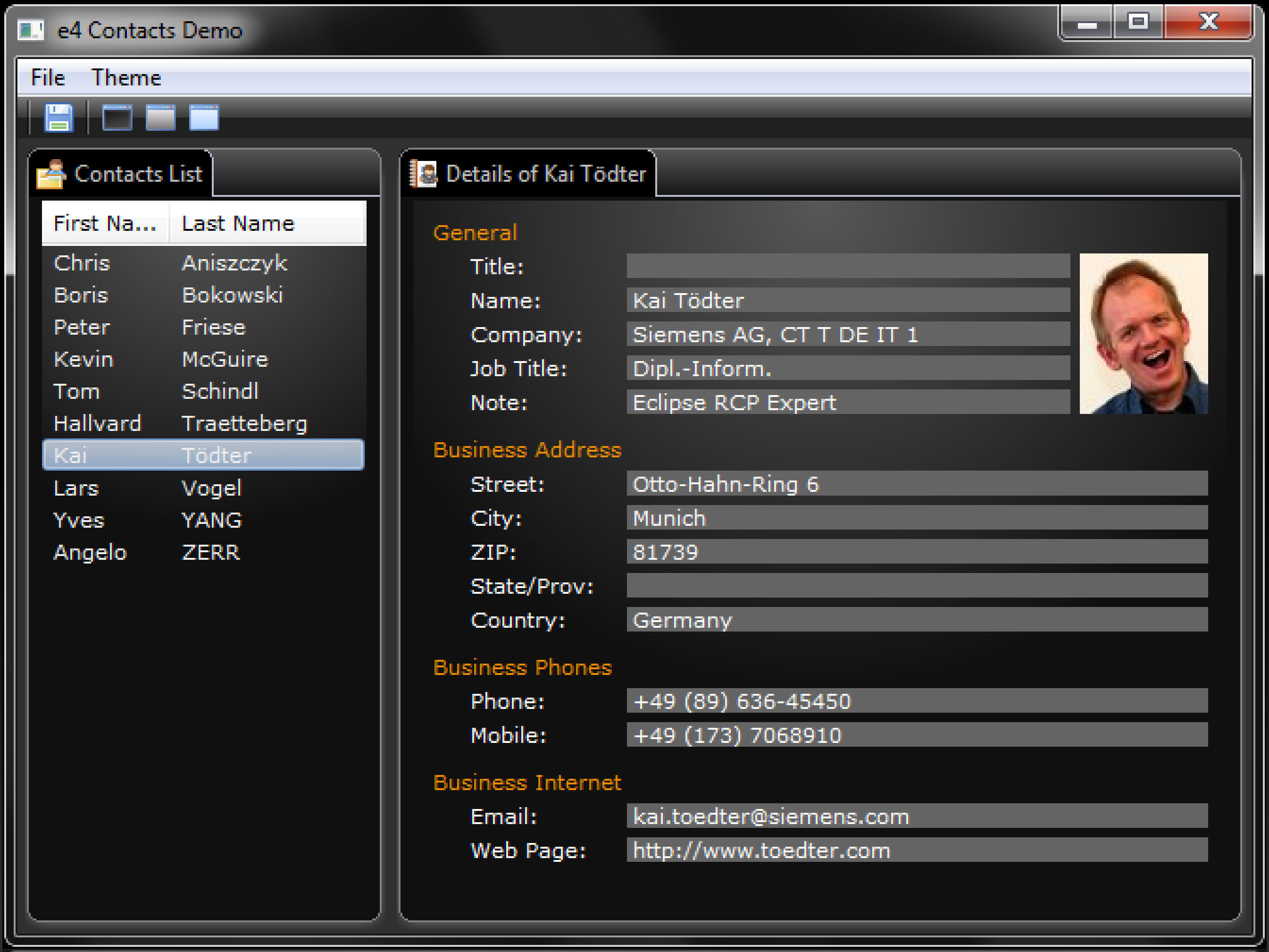
Business Phones

Phone:	+49 (89) 636-45450
Mobile:	+49 (173) 7068910

Business Internet

Email:	kai.toedter@siemens.com
Web Page:	http://www.toedter.com

Dark CSS Styling with radial Gradients



First Na...	Last Name
Chris	Aniszczyk
Boris	Bokowski
Peter	Friese
Kevin	McGuire
Tom	Schindl
Hallvard	Traetteberg
Kai	Tödter
Lars	Vogel
Yves	YANG
Angelo	ZERR

General

Title:	
Name:	Kai Tödter
Company:	Siemens AG, CT T DE IT 1
Job Title:	Dipl.-Inform.
Note:	Eclipse RCP Expert

Business Address

Street:	Otto-Hahn-Ring 6
City:	Munich
ZIP:	81739
State/Prov:	
Country:	Germany

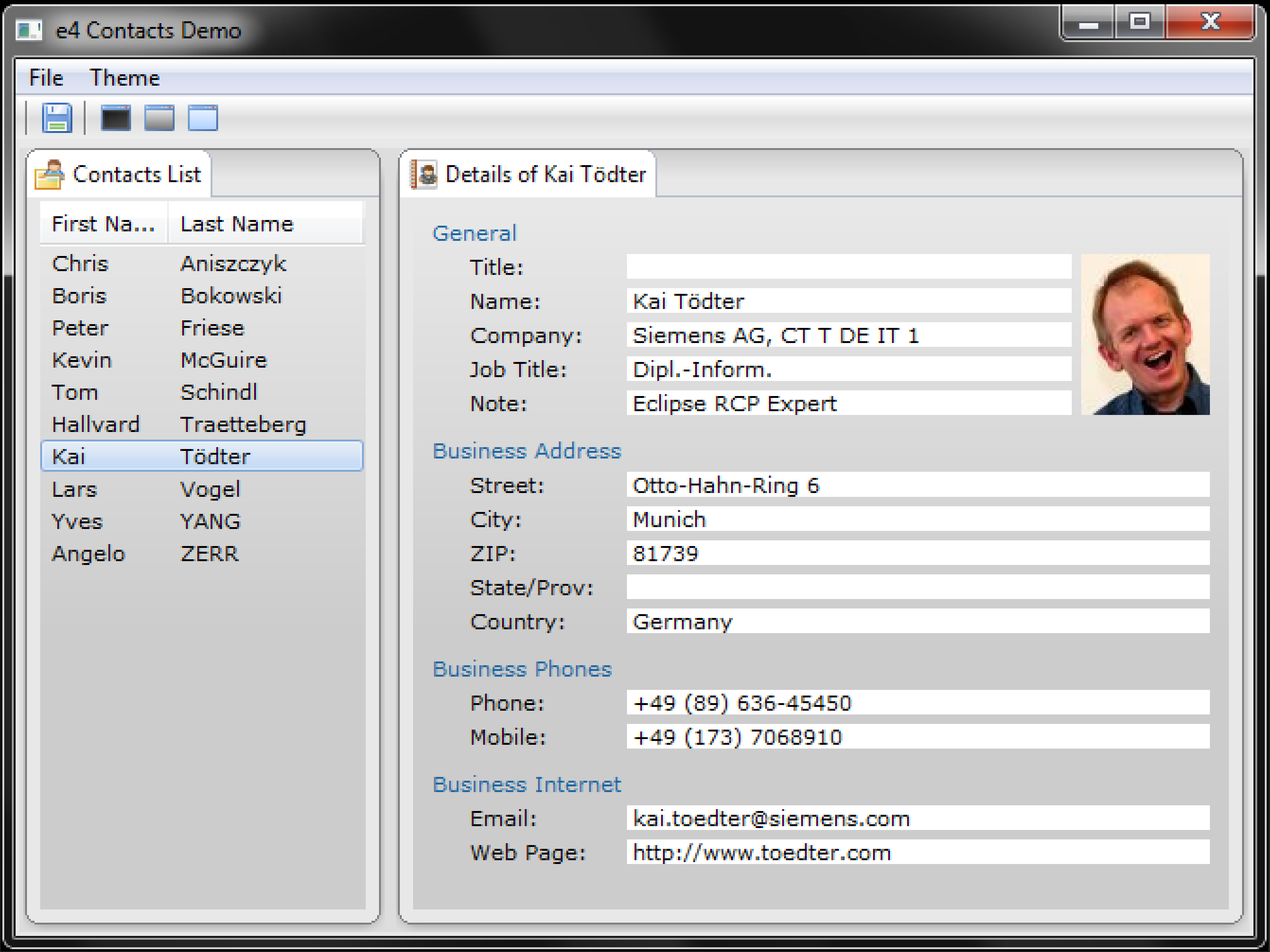
Business Phones

Phone:	+49 (89) 636-45450
Mobile:	+49 (173) 7068910

Business Internet

Email:	kai.toedter@siemens.com
Web Page:	http://www.toedter.com

Gray CSS Styling with linear Gradients



First Na...	Last Name
Chris	Aniszczyk
Boris	Bokowski
Peter	Friese
Kevin	McGuire
Tom	Schindl
Hallvard	Traetteberg
Kai	Tödter
Lars	Vogel
Yves	YANG
Angelo	ZERR

General

Title:	
Name:	Kai Tödter
Company:	Siemens AG, CT T DE IT 1
Job Title:	Dipl.-Inform.
Note:	Eclipse RCP Expert

Business Address

Street:	Otto-Hahn-Ring 6
City:	Munich
ZIP:	81739
State/Prov:	
Country:	Germany

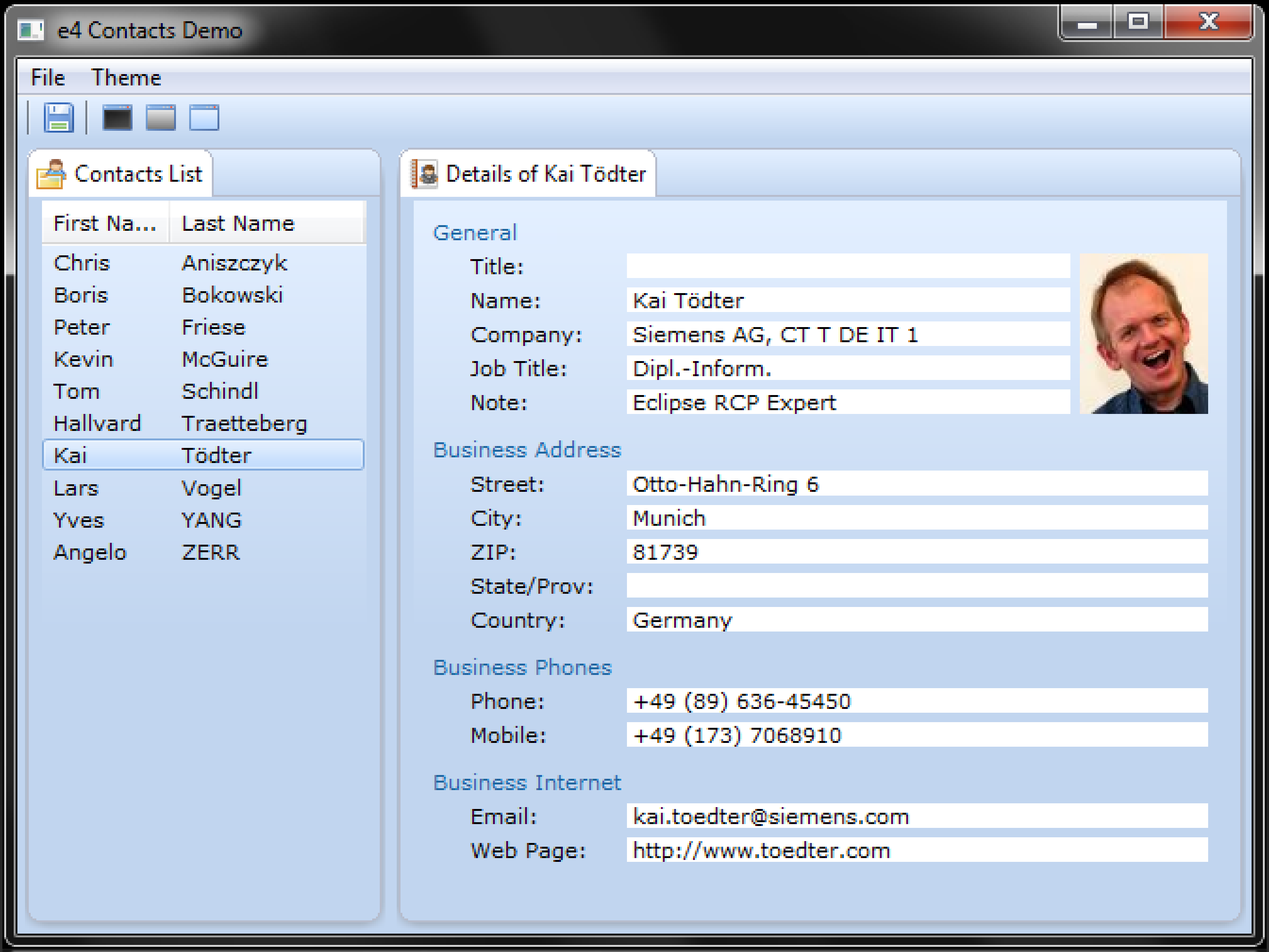
Business Phones

Phone:	+49 (89) 636-45450
Mobile:	+49 (173) 7068910

Business Internet

Email:	kai.toedter@siemens.com
Web Page:	http://www.toedter.com

Blue CSS Styling with linear Gradients



First Na...	Last Name
Chris	Aniszczyk
Boris	Bokowski
Peter	Friese
Kevin	McGuire
Tom	Schindl
Hallvard	Traetteberg
Kai	Tödter
Lars	Vogel
Yves	YANG
Angelo	ZERR

General

Title:	
Name:	Kai Tödter
Company:	Siemens AG, CT T DE IT 1
Job Title:	Dipl.-Inform.
Note:	Eclipse RCP Expert

Business Address

Street:	Otto-Hahn-Ring 6
City:	Munich
ZIP:	81739
State/Prov:	
Country:	Germany

Business Phones

Phone:	+49 (89) 636-45450
Mobile:	+49 (173) 7068910

Business Internet

Email:	kai.toedter@siemens.com
Web Page:	http://www.toedter.com

How does the CSS look like?

```
Label {
  font: Verdana 8px;
  color: rgb(240, 240, 240);
}
Table {
  background-color: gradient radial #575757 #101010 100%;
  color: rgb(240, 240, 240);
  font: Verdana 8px;
}
.MTrimBar {
  background-color: #777777 #373737 #202020 50% 50%;
  color: white;
  font: Verdana 8px;
}
```

Some Things you cannot style (yet)

- Menu bar background
- Table headers

Partly implemented:

- Gradients

My Wish:

- Having similar capabilities compared with WebKit’s gradients

How to enable CSS Styling (1)

Create a contribution to the extension point org.eclipse.e4.ui.css.swt.theme

```
<extension
      point="org.eclipse.e4.ui.css.swt.theme">
   <theme
      basestylesheeturi="css/blue.css"
      id="org.eclipse.e4.tutorial.contacts.themes.blue"
      label="Blue Theme">
   </theme>
</extension>
```

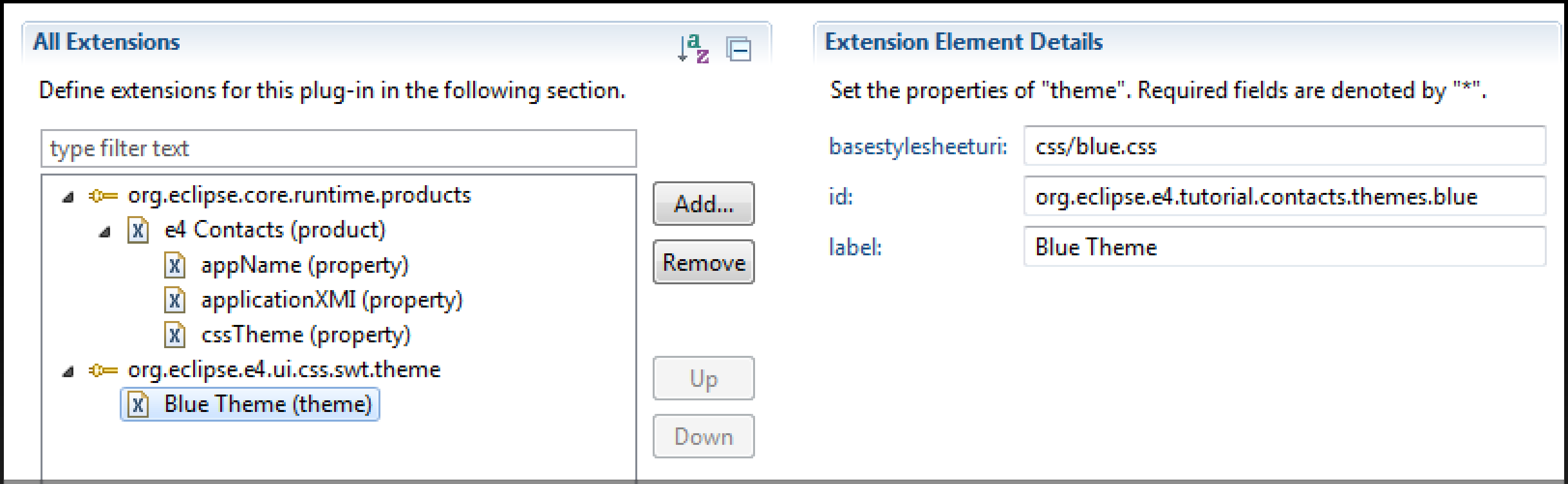
How to enable CSS Styling (2)

Create a contribution to the extension point
org.eclipse.core.runtime.products

```
<extension
   id="product"
   point="org.eclipse.core.runtime.products">
   <product
     application="org.eclipse.e4.ui.workbench.swt.application"
     name="e4 Contacs">
     <property
        name="applicationCSS"
        value="org.eclipse.e4.tutorial.contacts.themes.blue">
     </property>
   </product>
</extension>
```

How to enable CSS Styling (3)

- Extension point org.eclipse.ui.css.swt.theme



All Extensions

Define extensions for this plug-in in the following section.

```
type filter text
◢ org.eclipse.core.runtime.products
    ◢ e4 Contacts (product)
        appName (property)
        applicationXMI (property)
        cssTheme (property)
◢ org.eclipse.e4.ui.css.swt.theme
    Blue Theme (theme)
```

Extension Element Details

Set the properties of "theme". Required fields are denoted by "*".

basestylesheeturi:	css/blue.css
id:	org.eclipse.e4.tutorial.contacts.themes.blue
label:	Blue Theme

How to use custom attributes?

- Java:

```
Label label = new Label(parent, SWT.NONE);
label.setData("org.eclipse.e4.ui.css.id",  "SeparatorLabel");
```

- Workbench Model: Give the element an id
- CSS:

```
#SeparatorLabel {
    color: #f08d00;
}
```

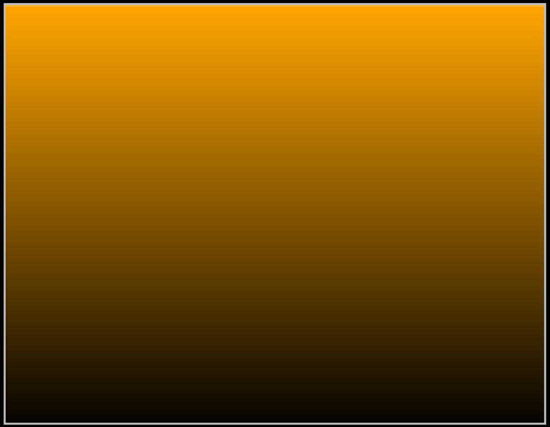
e4 CSS Editors

- CSS has a well known syntax
- But which UI elements can be styled?
- Which CSS attributes does a specific element support?

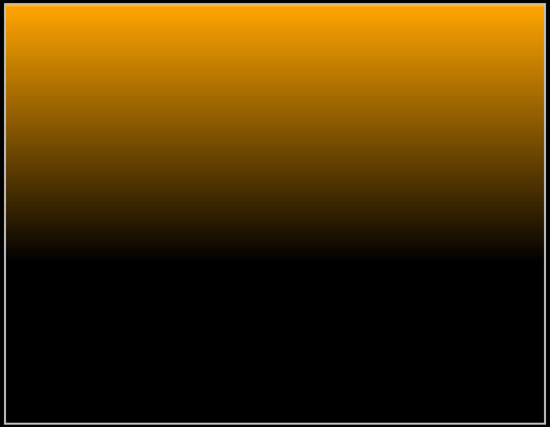
- At Eclipse Summit Europe 2010 Sven Efftinge and Sebastian Zarnekow showed a prototype of an Xtext-based editor, with content assist for both elements and attributes
- Let’s see, what Eclipse 4.1 will bring

Gradient Examples

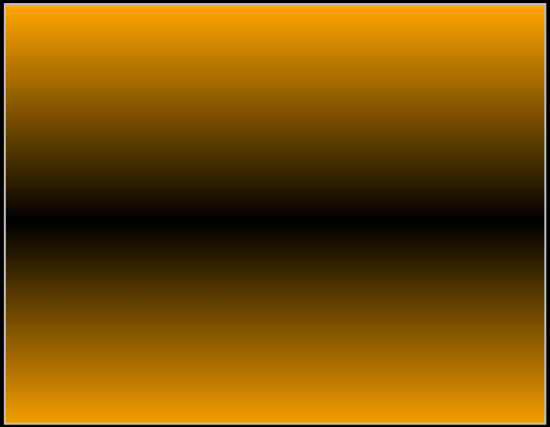
linear orange black



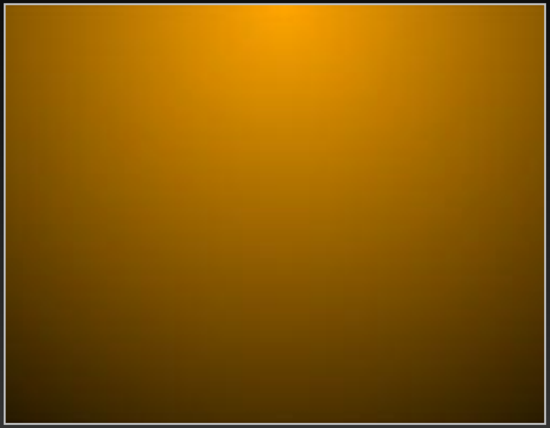
linear orange black 60%



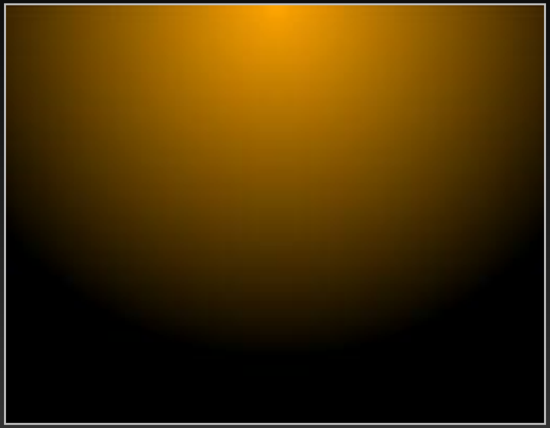
linear orange black orange 50% 100%



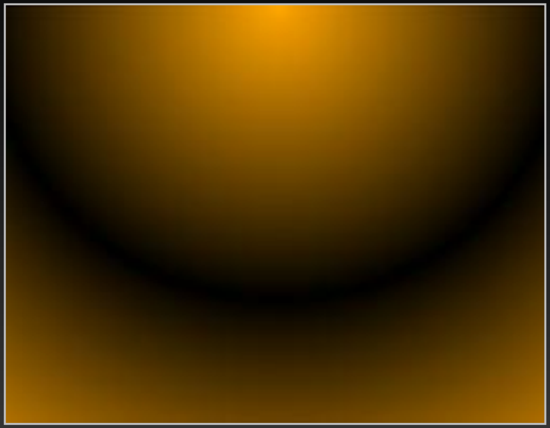
radial orange black



radial orange black 60%



radial orange black orange 50% 100%



Dynamic Theme Switching

- It is possible to change the CSS based theme at runtime
- Good for accessibility
- Good for user preferences

Picture from
http://www.sxc.hu/photo/823108

Theme Switching

- DI of IThemeEngine
- IThemeEngine provides API for applying styles and theme management

```
@Execute
public void execute( IThemeEngine engine ) {
    engine.setTheme(
        "org.eclipse.e4.demo.contacts.themes.darkgradient");
}
```

CSS Overview

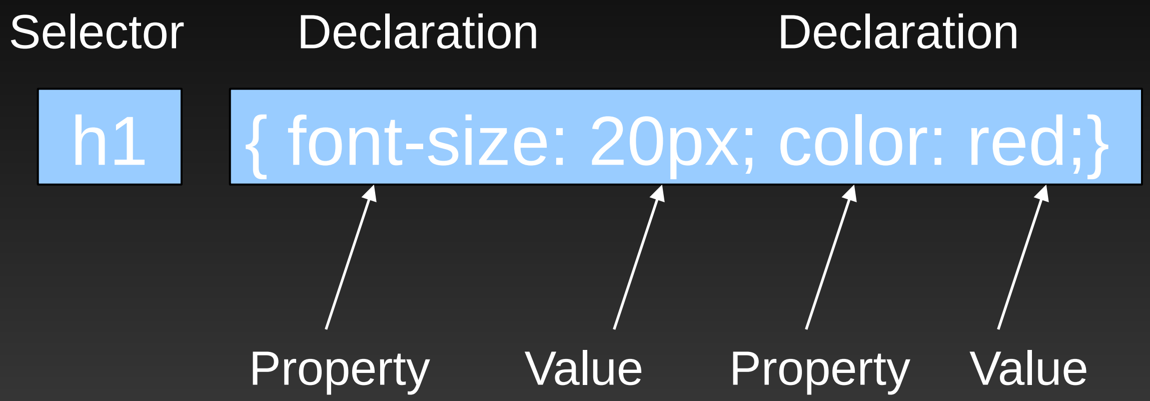
Picture from
http://www.sxc.hu/photo/1089931

CSS Overview

- CSS Syntax
- Element, Class and ID Selectors
- CSS Properties
- Cascade

CSS Syntax

- CSS stylesheets usually contain many rules
- rules are made up of selectors and declarations
- declarations are made up of a property and a value



Element, Class and ID selectors

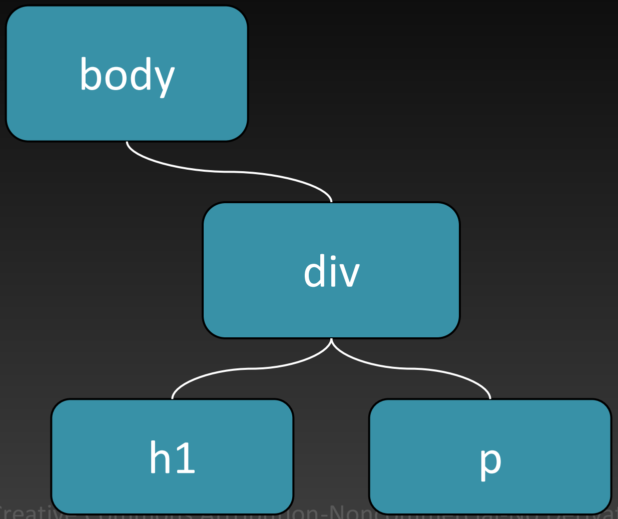
- Element type selectors in HTML matches HTML elements
  p {background-color: black;}
- Element selectors in the SWT styling world match the widget classnames
  Composite {background-color: red;}
- CSS allows you to specify your own selectors named *id* and *class*
- the *id* selector is used to specify the style for a single, unique element
  #myId {background-color: gold;}
- the class selector is used to specify the style for a number of elements
  .myClass {background-color: blue;}

Selectors combined

- It is possible to combine selectors:
- Matches all p elements with a class attribute “warning”:
  p.warning { STYLES }
- Matches all div elements with class attributes “foo” and “bar”:
  div.foo.bar { STYLES }
- Matches any element with class attributes “more”, “detail”, “please”
  .more.detail.please { STYLES }

Group and Position Selectors

- A group selector allows you to apply the same rules to different selectors
  selector, selector, etc { STYLES }
- To select a descendant element, use whitespace between the selectors.
  selector selector selector etc { STYLES }



Group and Position Selectors (2)

- A child selector selects elements that are children of the previous selector
  selector > selector > etc { STYLES }

CSS Properties

- property values come in the following forms:
  - text
  - numbers
  - lengths
  - percentages
  - comma-delimited lists of values
  - space-delimited lists of values
  - functions

CSS Property Examples

- color: white;
- color: rgb(255, 255, 255);
- color: #ffffff;
- border: 4px double black;
- margin: 0;
- background-image: url(“myPic.jpg”);
- line-height: 150%;
- padding: 0.25em;

!important Declarations

- A declaration that has the “!important” keyword added takes precedence over a normal declaration
- This can be used to used to add weight to declarations (in order to aid in cascade resolution)

```
p { font-size: 10px !important; }
```

Cascade

- In CSS it is possible to have cases where an element matches to multiple styles
- In order to determine which style to use, a resolution process is used
- the CSS Cascade looks at importance, origin, specificity and the source order of the style in question and assigns a weight
- declaration with the highest weight takes precedence

Cascade (2)

- In terms of selectors, the order of importance (from least to most) is:
  - Universal selectors - *
  - Element selectors
  - Class, Attribute and Pseudo Selectors
  - ID selectors
  - Above list repeated in order with !important added

CSS Engine API

Parsing
parseStyleSheet(), parseStyleDeclaration(), parsePropertyValue()

Apply Styles
applyStyles(), applyStyleDeclaration(), applyCSSProperty()

Retrieve Info
getViewCSS(), getDocumentCSS()

Utils
reset(), setErrorHandler()

SWT CSS Engine

- Implements CSS Engine API
- Extensible widget styling provided through the use of ElementProviders and PropertyHandlers
- Makes use of a number of utility classes to facilitate translation between text and SWT colors, fonts, images, gradients
- SWT provides a reskin() method in Widget that will send out a Skin event to all controls to notify that some value of the style has been changed

CSS Stylable Element

- CSSStylableElement interface defines methods to retrieve styling information about a particular element
- The CSS SWT engine has a hierarchy of element adapters that mirrors the SWT widget hierarchy (see WidgetElement)
- Element providers and the widgets that they support are registered through an extension point
- This allows for the extension of the CSS engine by adding custom property handlers and associating them with an element provider

Property Handlers

- ICSSPropertyHandler is used to apply a CSS property value to an element
- Property handlers perform the actual work of calling the methods on the widgets
- Property handlers are registered through an extension point
- Customized properties can be easily extended through this mechanism (ex. new CTabFolder properties)

Available Handlers

- Controls
  - background-color, background-image
  - cursor
  - font, font-style, font-weight, font-size, font-family
  - color
- Button, Label
  - alignment
- CTabFolder
  - border-visibe, maximized, minimized, maximize-visible, minimize-visible, mru-visible
  - simple, single, unselected-close-visible
  - tab-renderer, tab-height

Available Handlers (2)

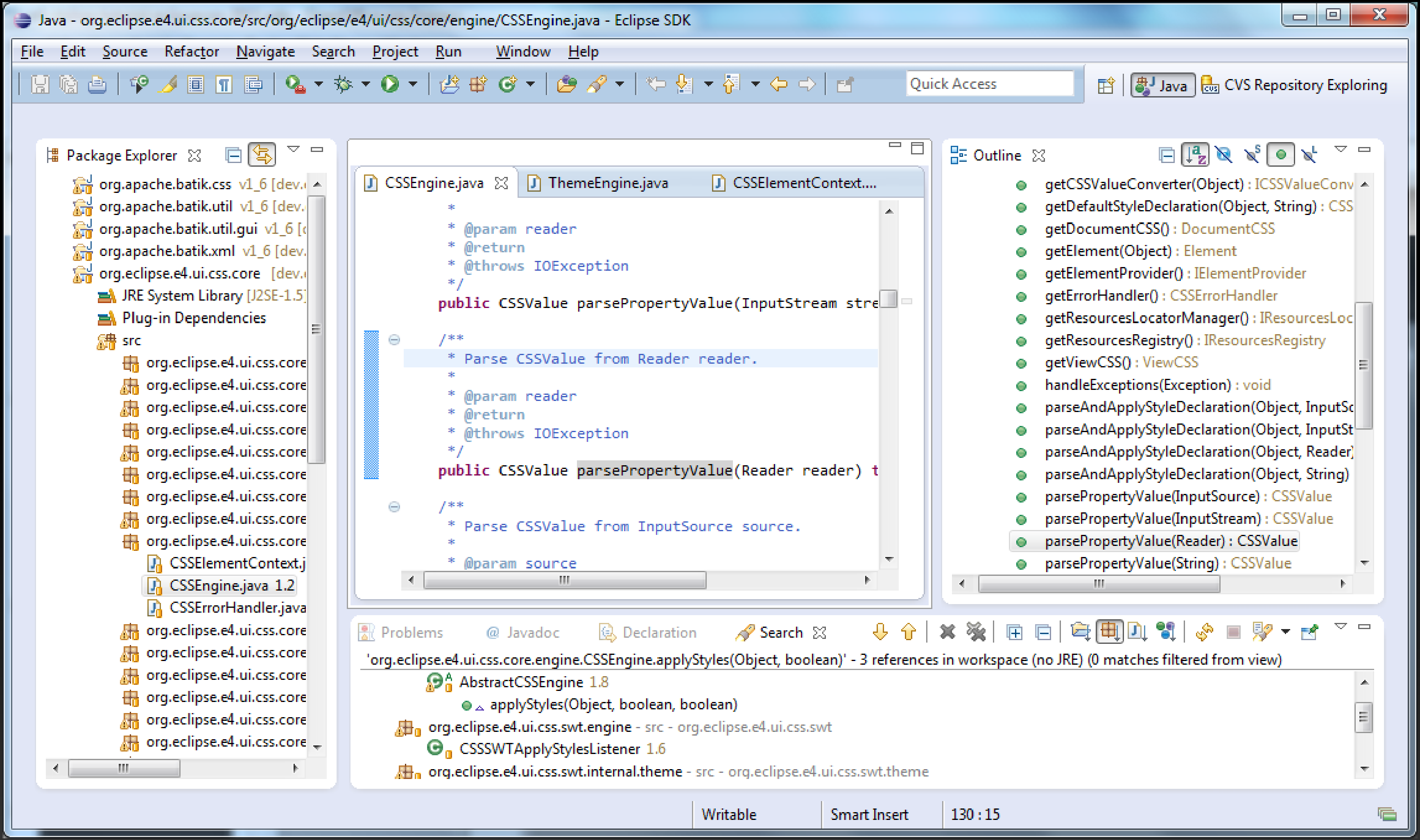
- For a complete list, check out the e4 wiki:

  http://wiki.eclipse.org/E4/CSS/SWT_Mapping

e4 Workbench

- The e4 workbench makes use of the CSS Engine to contribute a number of different looks
- IThemeEngine defines methods to register and manage themes as well as to get and applyStyles
- e4 provides an IThemeEngine implementation that creates an instance of the SWT CSS Engine
- e4 also provides an IStylingEngine which makes use of the IThemeEngine implementation to set classnames, set ids, and style widgets
- IStylingEngine can be obtained from a context, service registry or through DI (as shown in the contacts example)

4.x SDK on Windows 7



Java - org.eclipse.e4.ui.css.core/src/org/eclipse/e4/ui/css/core/engine/CSSEngine.java - Eclipse SDK

org.apache.batik.css v1_6 [dev.
org.apache.batik.util v1_6 [dev.
org.apache.batik.util.gui v1_6 [
org.apache.batik.xml v1_6 [dev.
org.eclipse.e4.ui.css.core [dev.
JRE System Library [J2SE-1.5]
Plug-in Dependencies
src
org.eclipse.e4.ui.css.core
org.eclipse.e4.ui.css.core
org.eclipse.e4.ui.css.core
org.eclipse.e4.ui.css.core
org.eclipse.e4.ui.css.core
org.eclipse.e4.ui.css.core
org.eclipse.e4.ui.css.core
org.eclipse.e4.ui.css.core
org.eclipse.e4.ui.css.core
CSSElementContext.j
CSSEngine.java 1.2
CSSErrorHandler.java
org.eclipse.e4.ui.css.core
org.eclipse.e4.ui.css.core
org.eclipse.e4.ui.css.core
org.eclipse.e4.ui.css.core
org.eclipse.e4.ui.css.core
org.eclipse.e4.ui.css.core

CSSEngine.java | ThemeEngine.java | CSSElementContext....

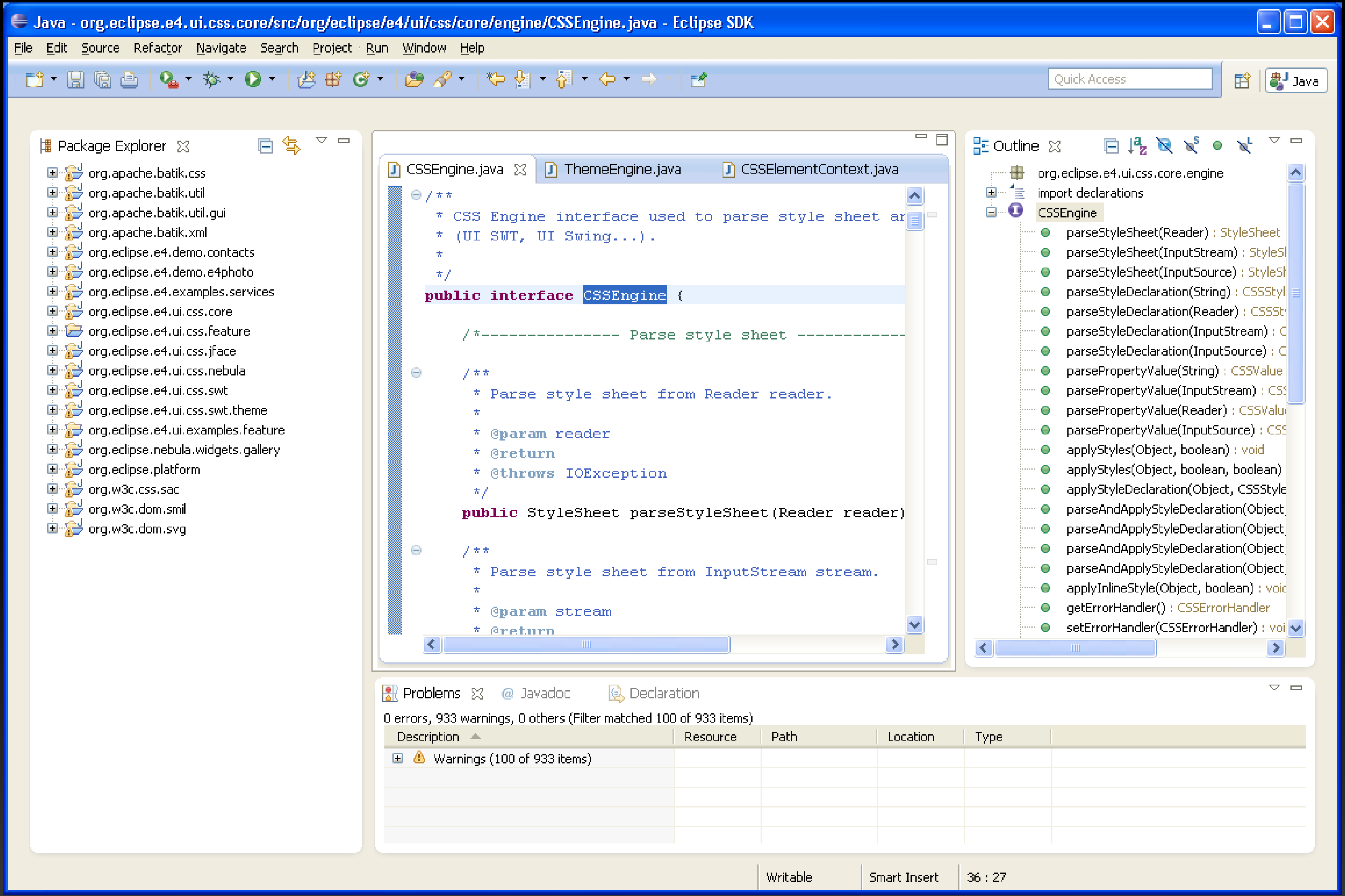
```
 *
 * @param reader
 * @return
 * @throws IOException
 */
public CSSValue parsePropertyValue(InputStream stre

/**
 * Parse CSSValue from Reader reader.
 *
 * @param reader
 * @return
 * @throws IOException
 */
public CSSValue parsePropertyValue(Reader reader) t

/**
 * Parse CSSValue from InputSource source.
 *
 * @param source
```

getCSSValueConverter(Object) : ICSSValueConv
getDefaultStyleDeclaration(Object, String) : CSS
getDocumentCSS() : DocumentCSS
getElement(Object) : Element
getElementProvider() : IElementProvider
getErrorHandler() : CSSErrorHandler
getResourcesLocatorManager() : IResourcesLoc
getResourcesRegistry() : IResourcesRegistry
getViewCSS() : ViewCSS
handleExceptions(Exception) : void
parseAndApplyStyleDeclaration(Object, InputSo
parseAndApplyStyleDeclaration(Object, InputSt
parseAndApplyStyleDeclaration(Object, Reader)
parseAndApplyStyleDeclaration(Object, String)
parsePropertyValue(InputSource) : CSSValue
parsePropertyValue(InputStream) : CSSValue
parsePropertyValue(Reader) : CSSValue
parsePropertyValue(String) : CSSValue

'org.eclipse.e4.ui.css.core.engine.CSSEngine.applyStyles(Object, boolean)' - 3 references in workspace (no JRE) (0 matches filtered from view)
AbstractCSSEngine 1.8
applyStyles(Object, boolean, boolean)
org.eclipse.e4.ui.css.swt.engine - src - org.eclipse.e4.ui.css.swt
CSSSWTApplyStylesListener 1.6
org.eclipse.e4.ui.css.swt.internal.theme - src - org.eclipse.e4.ui.css.swt.theme

Writable | Smart Insert | 130 : 15

4.x SDK on XP (Blue)


Java - org.eclipse.e4.ui.css.core/src/org/eclipse/e4/ui/css/core/engine/CSSEngine.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Package Explorer
CSSEngine.java
ThemeEngine.java
CSSElementContext.java
Outline
Problems
Javadoc
Declaration
Writable
Smart Insert
36 : 27

Package Explorer:

- org.apache.batik.css
- org.apache.batik.util
- org.apache.batik.util.gui
- org.apache.batik.xml
- org.eclipse.e4.demo.contacts
- org.eclipse.e4.demo.e4photo
- org.eclipse.e4.examples.services
- org.eclipse.e4.ui.css.core
- org.eclipse.e4.ui.css.feature
- org.eclipse.e4.ui.css.jface
- org.eclipse.e4.ui.css.nebula
- org.eclipse.e4.ui.css.swt
- org.eclipse.e4.ui.css.swt.theme
- org.eclipse.e4.ui.examples.feature
- org.eclipse.nebula.widgets.gallery
- org.eclipse.platform
- org.w3c.css.sac
- org.w3c.dom.smil
- org.w3c.dom.svg

```
/**
 * CSS Engine interface used to parse style sheet ar
 * (UI SWT, UI Swing...).
 *
 */
public interface CSSEngine {

    /*--------------- Parse style sheet ------------

    /**
     * Parse style sheet from Reader reader.
     *
     * @param reader
     * @return
     * @throws IOException
     */
    public StyleSheet parseStyleSheet(Reader reader)

    /**
     * Parse style sheet from InputStream stream.
     *
     * @param stream
     * @return
```

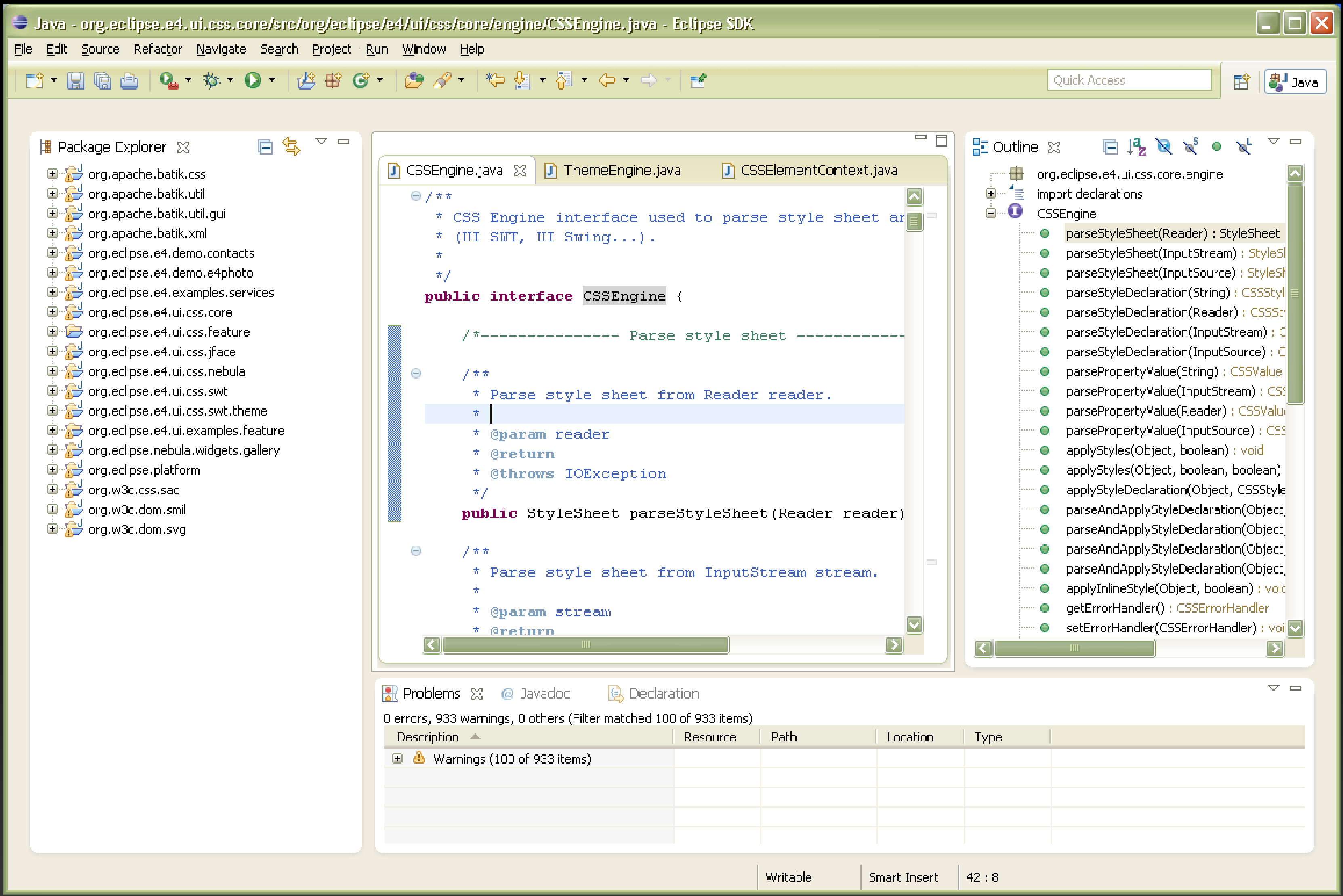
Outline:

```
org.eclipse.e4.ui.css.core.engine
import declarations
CSSEngine
    parseStyleSheet(Reader) : StyleSheet
    parseStyleSheet(InputStream) : StyleSl
    parseStyleSheet(InputSource) : StyleSl
    parseStyleDeclaration(String) : CSSStyl
    parseStyleDeclaration(Reader) : CSSSt
    parseStyleDeclaration(InputStream) : C
    parseStyleDeclaration(InputSource) : C
    parsePropertyValue(String) : CSSValue
    parsePropertyValue(InputStream) : CSS
    parsePropertyValue(Reader) : CSSValu
    parsePropertyValue(InputSource) : CSS
    applyStyles(Object, boolean) : void
    applyStyles(Object, boolean, boolean)
    applyStyleDeclaration(Object, CSSStyle
    parseAndApplyStyleDeclaration(Object,
    parseAndApplyStyleDeclaration(Object,
    parseAndApplyStyleDeclaration(Object,
    parseAndApplyStyleDeclaration(Object,
    applyInlineStyle(Object, boolean) : voic
    getErrorHandler() : CSSErrorHandler
    setErrorHandler(CSSErrorHandler) : voi
```

0 errors, 933 warnings, 0 others (Filter matched 100 of 933 items)

Description	Resource	Path	Location	Type
Warnings (100 of 933 items)				

4.x SDK on XP (Olive)


Java - org.eclipse.e4.ui.css.core/src/org/eclipse/e4/ui/css/core/engine/CSSEngine.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Package Explorer
CSSEngine.java
ThemeEngine.java
CSSElementContext.java
Outline
Problems
Javadoc
Declaration

Package Explorer:

- org.apache.batik.css
- org.apache.batik.util
- org.apache.batik.util.gui
- org.apache.batik.xml
- org.eclipse.e4.demo.contacts
- org.eclipse.e4.demo.e4photo
- org.eclipse.e4.examples.services
- org.eclipse.e4.ui.css.core
- org.eclipse.e4.ui.css.feature
- org.eclipse.e4.ui.css.jface
- org.eclipse.e4.ui.css.nebula
- org.eclipse.e4.ui.css.swt
- org.eclipse.e4.ui.css.swt.theme
- org.eclipse.e4.ui.examples.feature
- org.eclipse.nebula.widgets.gallery
- org.eclipse.platform
- org.w3c.css.sac
- org.w3c.dom.smil
- org.w3c.dom.svg

```
/**
 * CSS Engine interface used to parse style sheet ar
 * (UI SWT, UI Swing...).
 *
 */
public interface CSSEngine {

    /*--------------- Parse style sheet ------------

    /**
     * Parse style sheet from Reader reader.
     *
     * @param reader
     * @return
     * @throws IOException
     */
    public StyleSheet parseStyleSheet(Reader reader)

    /**
     * Parse style sheet from InputStream stream.
     *
     * @param stream
     * @return
```

Outline:

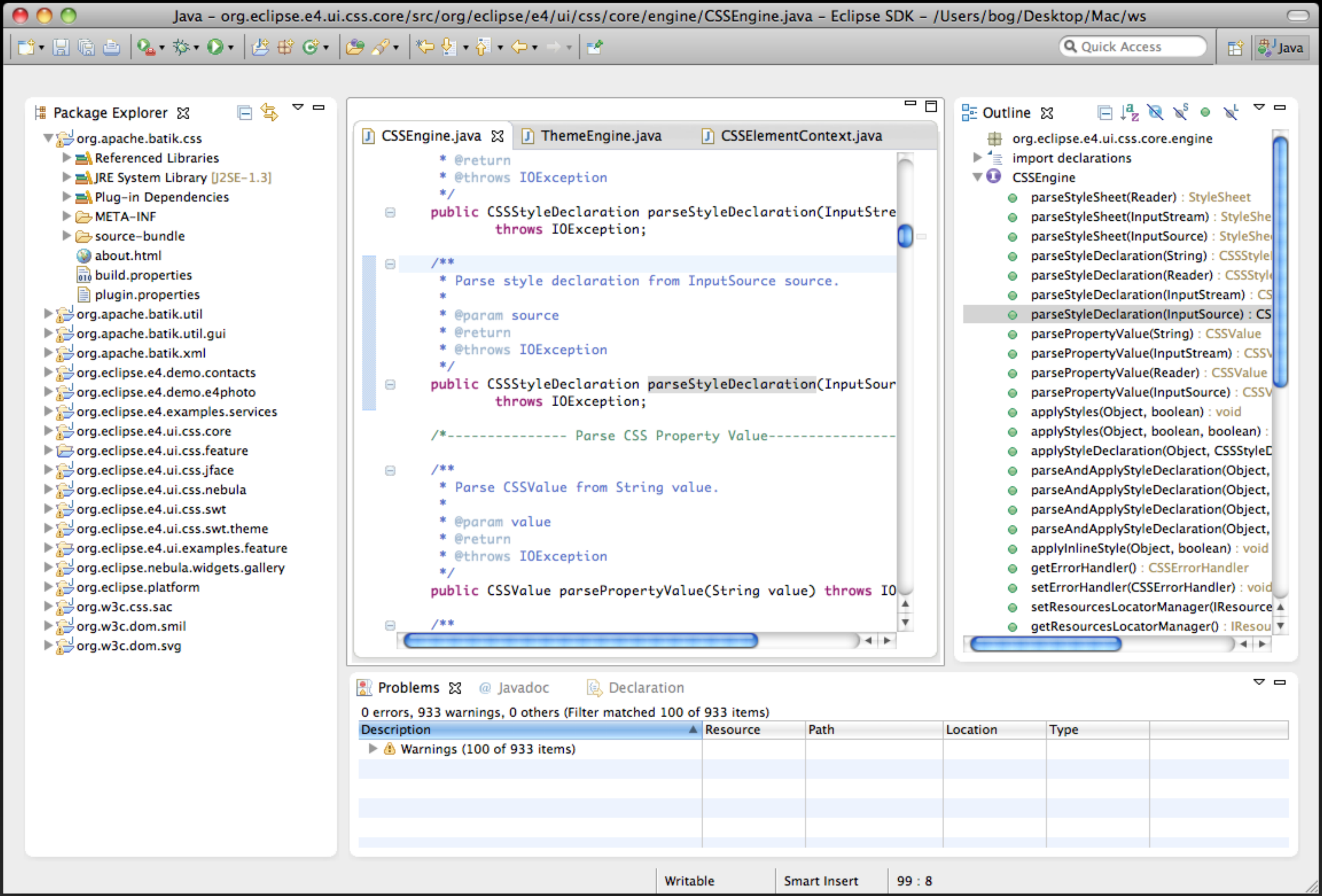
- org.eclipse.e4.ui.css.core.engine
- import declarations
- CSSEngine
  - parseStyleSheet(Reader) : StyleSheet
  - parseStyleSheet(InputStream) : StyleSl
  - parseStyleSheet(InputSource) : StyleSl
  - parseStyleDeclaration(String) : CSSStyl
  - parseStyleDeclaration(Reader) : CSSSt
  - parseStyleDeclaration(InputStream) : C
  - parseStyleDeclaration(InputSource) : C
  - parsePropertyValue(String) : CSSValue
  - parsePropertyValue(InputStream) : CSS
  - parsePropertyValue(Reader) : CSSValu
  - parsePropertyValue(InputSource) : CSS
  - applyStyles(Object, boolean) : void
  - applyStyles(Object, boolean, boolean)
  - applyStyleDeclaration(Object, CSSStyle
  - parseAndApplyStyleDeclaration(Object,
  - parseAndApplyStyleDeclaration(Object,
  - parseAndApplyStyleDeclaration(Object,
  - parseAndApplyStyleDeclaration(Object,
  - applyInlineStyle(Object, boolean) : voic
  - getErrorHandler() : CSSErrorHandler
  - setErrorHandler(CSSErrorHandler) : voi

0 errors, 933 warnings, 0 others (Filter matched 100 of 933 items)

Description	Resource	Path	Location	Type
Warnings (100 of 933 items)				

Writable | Smart Insert | 42 : 8

4.x SDK on MAC



Java – org.eclipse.e4.ui.css.core/src/org/eclipse/e4/ui/css/core/engine/CSSEngine.java – Eclipse SDK – /Users/bog/Desktop/Mac/ws

Package Explorer:

- org.apache.batik.css
  - Referenced Libraries
  - JRE System Library [J2SE-1.3]
  - Plug-in Dependencies
  - META-INF
  - source-bundle
  - about.html
  - build.properties
  - plugin.properties
- org.apache.batik.util
- org.apache.batik.util.gui
- org.apache.batik.xml
- org.eclipse.e4.demo.contacts
- org.eclipse.e4.demo.e4photo
- org.eclipse.e4.examples.services
- org.eclipse.e4.ui.css.core
- org.eclipse.e4.ui.css.feature
- org.eclipse.e4.ui.css.jface
- org.eclipse.e4.ui.css.nebula
- org.eclipse.e4.ui.css.swt
- org.eclipse.e4.ui.css.swt.theme
- org.eclipse.e4.ui.examples.feature
- org.eclipse.nebula.widgets.gallery
- org.eclipse.platform
- org.w3c.css.sac
- org.w3c.dom.smil
- org.w3c.dom.svg

Editor tabs: CSSEngine.java | ThemeEngine.java | CSSElementContext.java

```
     * @return
     * @throws IOException
     */
    public CSSStyleDeclaration parseStyleDeclaration(InputStre
            throws IOException;

    /**
     * Parse style declaration from InputSource source.
     *
     * @param source
     * @return
     * @throws IOException
     */
    public CSSStyleDeclaration parseStyleDeclaration(InputSour
            throws IOException;

    /*-------------- Parse CSS Property Value---------------

    /**
     * Parse CSSValue from String value.
     *
     * @param value
     * @return
     * @throws IOException
     */
    public CSSValue parsePropertyValue(String value) throws IO

    /**
```

Outline:

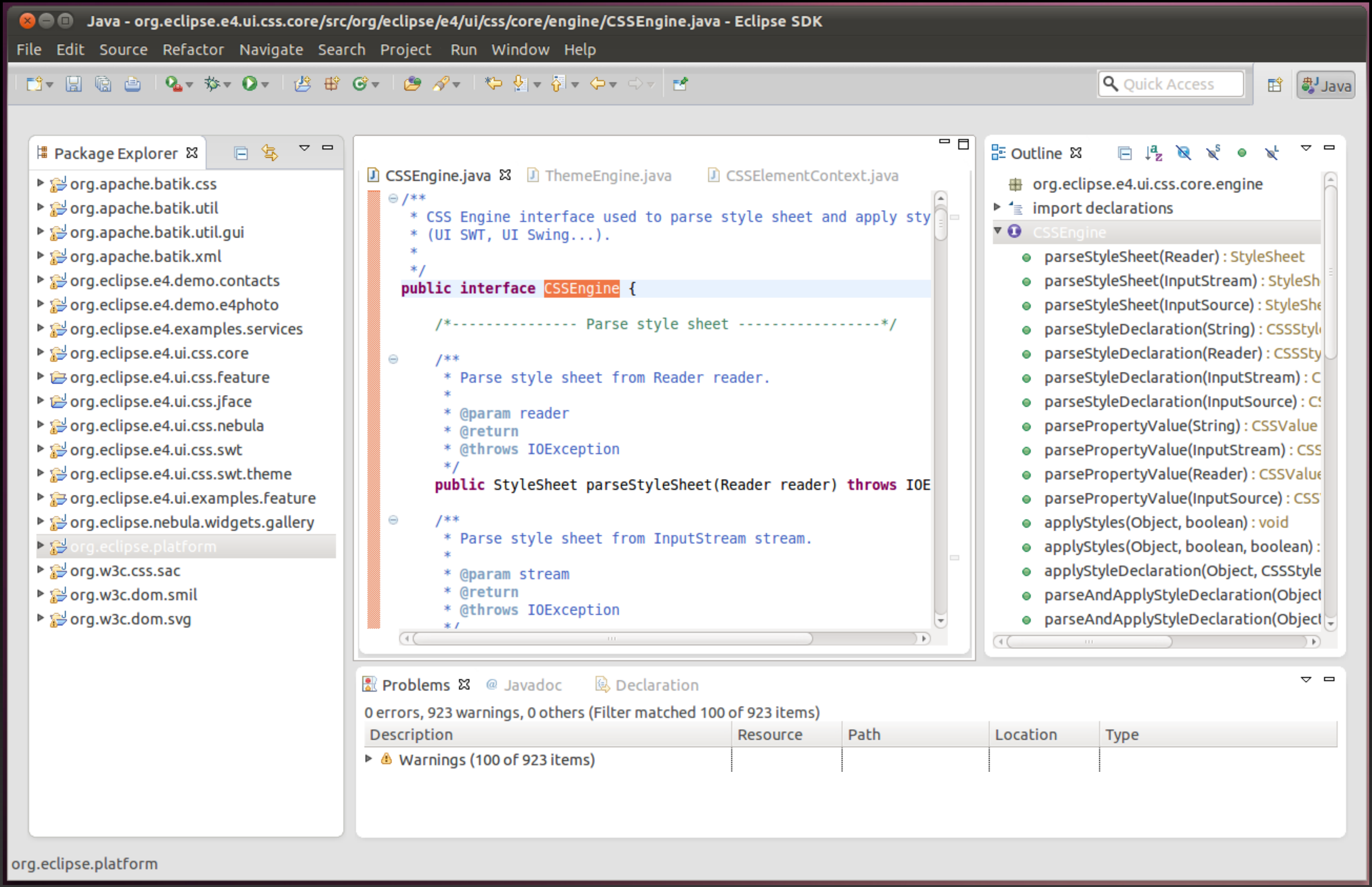
- org.eclipse.e4.ui.css.core.engine
- import declarations
- CSSEngine
  - parseStyleSheet(Reader) : StyleSheet
  - parseStyleSheet(InputStream) : StyleShe
  - parseStyleSheet(InputSource) : StyleShe
  - parseStyleDeclaration(String) : CSSStyle
  - parseStyleDeclaration(Reader) : CSSStyl
  - parseStyleDeclaration(InputStream) : CS
  - parseStyleDeclaration(InputSource) : CS
  - parsePropertyValue(String) : CSSValue
  - parsePropertyValue(InputStream) : CSSV
  - parsePropertyValue(Reader) : CSSValue
  - parsePropertyValue(InputSource) : CSSV
  - applyStyles(Object, boolean) : void
  - applyStyles(Object, boolean, boolean) :
  - applyStyleDeclaration(Object, CSSStyleD
  - parseAndApplyStyleDeclaration(Object,
  - parseAndApplyStyleDeclaration(Object,
  - parseAndApplyStyleDeclaration(Object,
  - parseAndApplyStyleDeclaration(Object,
  - applyInlineStyle(Object, boolean) : void
  - getErrorHandler() : CSSErrorHandler
  - setErrorHandler(CSSErrorHandler) : void
  - setResourcesLocatorManager(IResource
  - getResourcesLocatorManager() : IResou

Problems: 0 errors, 933 warnings, 0 others (Filter matched 100 of 933 items)

Description	Resource	Path	Location	Type
Warnings (100 of 933 items)				

Writable | Smart Insert | 99 : 8

4.x SDK on Linux



Java - org.eclipse.e4.ui.css.core/src/org/eclipse/e4/ui/css/core/engine/CSSEngine.java - Eclipse SDK

- org.apache.batik.css
- org.apache.batik.util
- org.apache.batik.util.gui
- org.apache.batik.xml
- org.eclipse.e4.demo.contacts
- org.eclipse.e4.demo.e4photo
- org.eclipse.e4.examples.services
- org.eclipse.e4.ui.css.core
- org.eclipse.e4.ui.css.feature
- org.eclipse.e4.ui.css.jface
- org.eclipse.e4.ui.css.nebula
- org.eclipse.e4.ui.css.swt
- org.eclipse.e4.ui.css.swt.theme
- org.eclipse.e4.ui.examples.feature
- org.eclipse.nebula.widgets.gallery
- org.eclipse.platform
- org.w3c.css.sac
- org.w3c.dom.smil
- org.w3c.dom.svg

```java
/**
 * CSS Engine interface used to parse style sheet and apply sty
 * (UI SWT, UI Swing...).
 *
 */
public interface CSSEngine {

    /*--------------- Parse style sheet -----------------*/

    /**
     * Parse style sheet from Reader reader.
     *
     * @param reader
     * @return
     * @throws IOException
     */
    public StyleSheet parseStyleSheet(Reader reader) throws IOE

    /**
     * Parse style sheet from InputStream stream.
     *
     * @param stream
     * @return
     * @throws IOException
     */
```

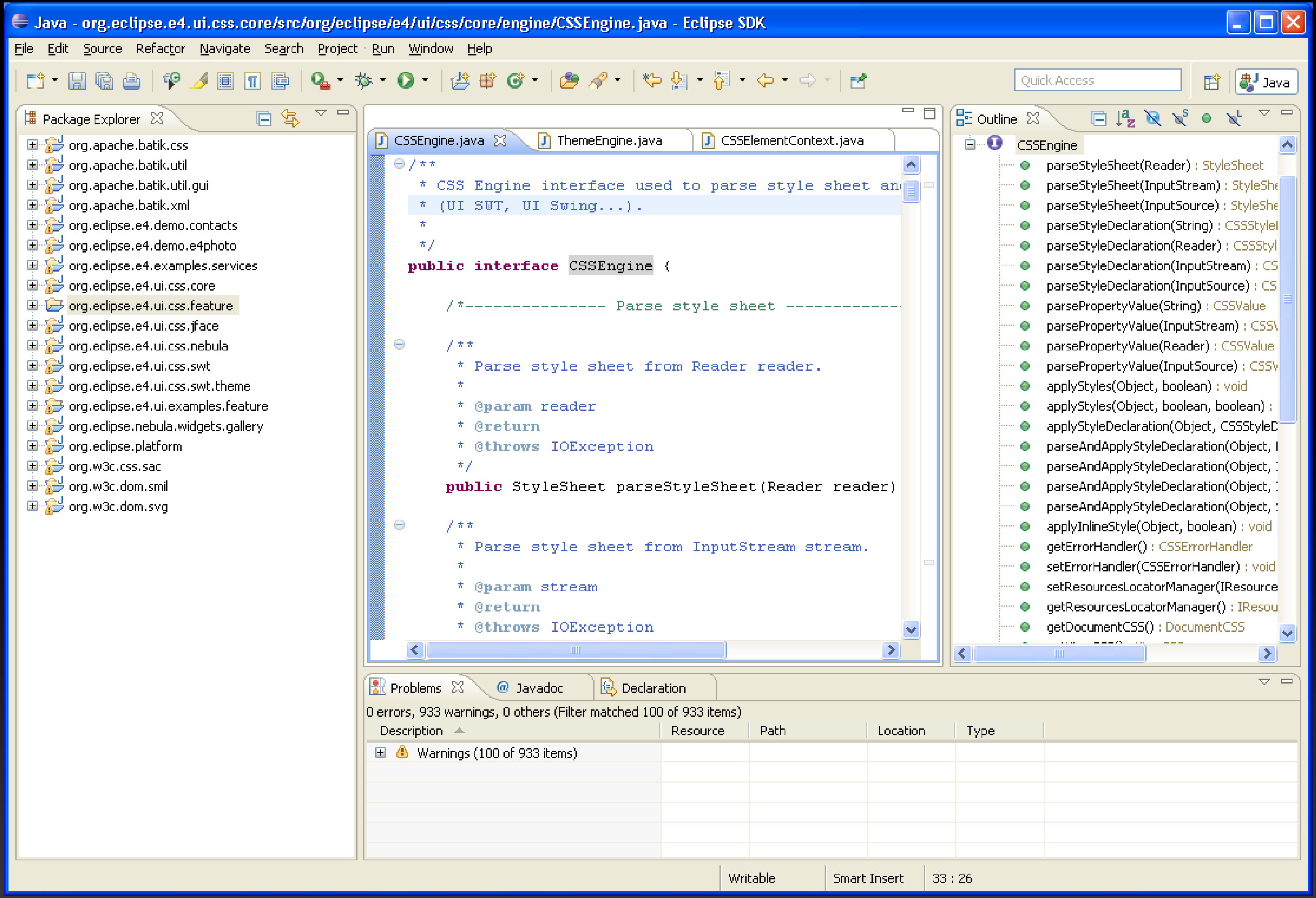
- org.eclipse.e4.ui.css.core.engine
- import declarations
- CSSEngine
  - parseStyleSheet(Reader) : StyleSheet
  - parseStyleSheet(InputStream) : StyleSh
  - parseStyleSheet(InputSource) : StyleShe
  - parseStyleDeclaration(String) : CSSStyl
  - parseStyleDeclaration(Reader) : CSSSty
  - parseStyleDeclaration(InputStream) : C
  - parseStyleDeclaration(InputSource) : C
  - parsePropertyValue(String) : CSSValue
  - parsePropertyValue(InputStream) : CSS
  - parsePropertyValue(Reader) : CSSValue
  - parsePropertyValue(InputSource) : CSS
  - applyStyles(Object, boolean) : void
  - applyStyles(Object, boolean, boolean) :
  - applyStyleDeclaration(Object, CSSStyle
  - parseAndApplyStyleDeclaration(Object
  - parseAndApplyStyleDeclaration(Object

0 errors, 923 warnings, 0 others (Filter matched 100 of 923 items)

Description	Resource	Path	Location	Type
Warnings (100 of 923 items)				

org.eclipse.platform

4.x SDK on XP with “Retro” styling


Java - org.eclipse.e4.ui.css.core/src/org/eclipse/e4/ui/css/core/engine/CSSEngine.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Package Explorer
CSSEngine.java
ThemeEngine.java
CSSElementContext.java
Outline
Problems
Javadoc
Declaration

```
org.apache.batik.css
org.apache.batik.util
org.apache.batik.util.gui
org.apache.batik.xml
org.eclipse.e4.demo.contacts
org.eclipse.e4.demo.e4photo
org.eclipse.e4.examples.services
org.eclipse.e4.ui.css.core
org.eclipse.e4.ui.css.feature
org.eclipse.e4.ui.css.jface
org.eclipse.e4.ui.css.nebula
org.eclipse.e4.ui.css.swt
org.eclipse.e4.ui.css.swt.theme
org.eclipse.e4.ui.examples.feature
org.eclipse.nebula.widgets.gallery
org.eclipse.platform
org.w3c.css.sac
org.w3c.dom.smil
org.w3c.dom.svg
```

```java
/**
 * CSS Engine interface used to parse style sheet an
 * (UI SWT, UI Swing...).
 *
 */
public interface CSSEngine {

    /*--------------- Parse style sheet ------------

    /**
     * Parse style sheet from Reader reader.
     *
     * @param reader
     * @return
     * @throws IOException
     */
    public StyleSheet parseStyleSheet(Reader reader)

    /**
     * Parse style sheet from InputStream stream.
     *
     * @param stream
     * @return
     * @throws IOException
```

```
CSSEngine
    parseStyleSheet(Reader) : StyleSheet
    parseStyleSheet(InputStream) : StyleShe
    parseStyleSheet(InputSource) : StyleShe
    parseStyleDeclaration(String) : CSSStyle
    parseStyleDeclaration(Reader) : CSSStyl
    parseStyleDeclaration(InputStream) : CS
    parseStyleDeclaration(InputSource) : CS
    parsePropertyValue(String) : CSSValue
    parsePropertyValue(InputStream) : CSSV
    parsePropertyValue(Reader) : CSSValue
    parsePropertyValue(InputSource) : CSSV
    applyStyles(Object, boolean) : void
    applyStyles(Object, boolean, boolean) :
    applyStyleDeclaration(Object, CSSStyleD
    parseAndApplyStyleDeclaration(Object, I
    parseAndApplyStyleDeclaration(Object,
    parseAndApplyStyleDeclaration(Object,
    parseAndApplyStyleDeclaration(Object,
    applyInlineStyle(Object, boolean) : void
    getErrorHandler() : CSSErrorHandler
    setErrorHandler(CSSErrorHandler) : void
    setResourcesLocatorManager(IResource
    getResourcesLocatorManager() : IResou
    getDocumentCSS() : DocumentCSS
```

0 errors, 933 warnings, 0 others (Filter matched 100 of 933 items)

Description	Resource	Path	Location	Type
Warnings (100 of 933 items)				

Writable | Smart Insert | 33 : 26

Lab: Let's take a look!

- Start up your e4 SDK build and create a new workspace
- File>Import>Existing Projects into Workspace>Select archive file:
- Browse to where you saved the “e4styling.zip” archive , select it>Open
- Click Finish in the wizard to complete the import

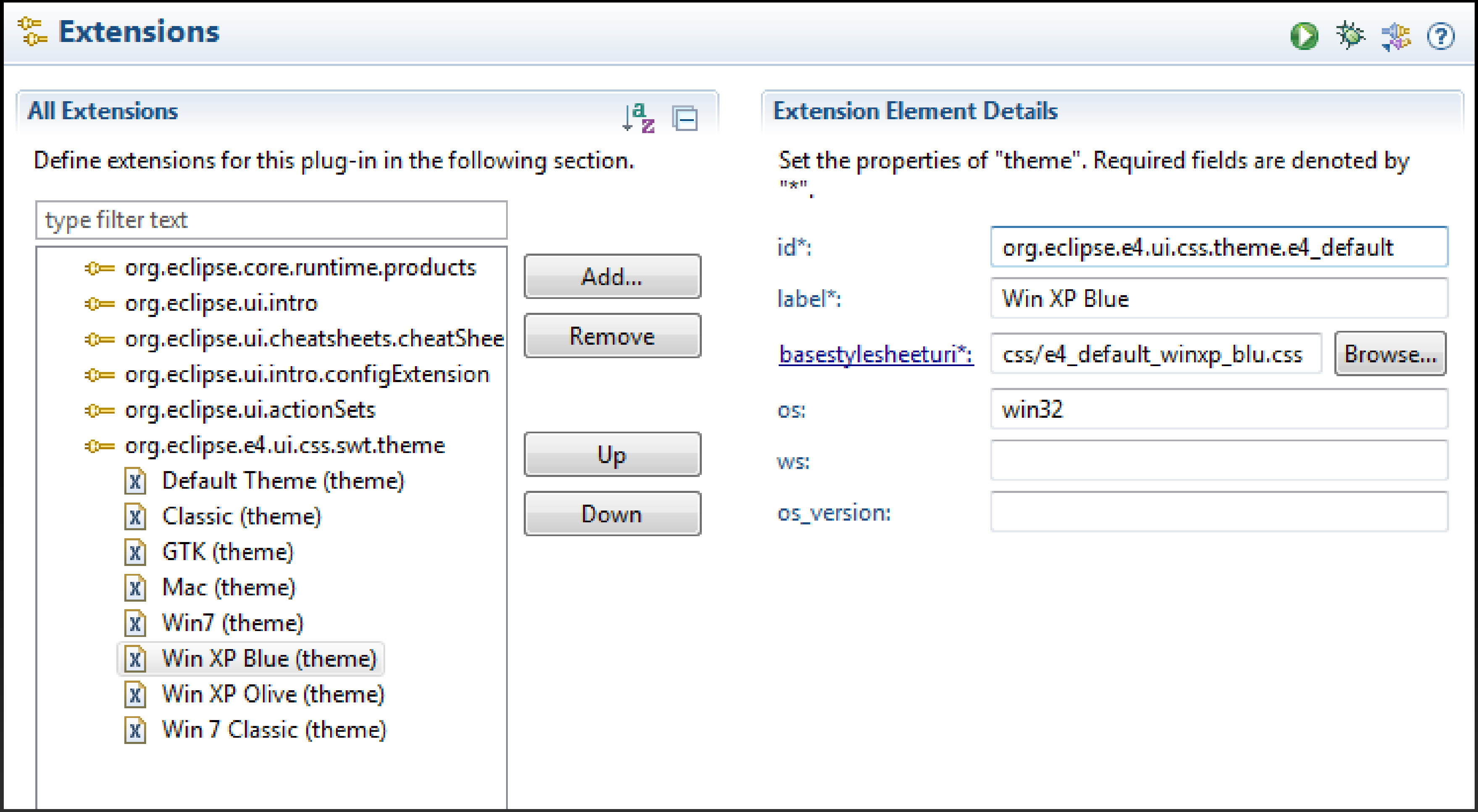
Lab: e4 Contacts Demo Styling

- Double click the file contacts.product in the project org.eclipse.e4.demo.contacts
- Click on „Launch an Eclipse Application“ =Y the contacts demo starts
- Switch the theme to “Blue Gradient Theme”
- Edit the css file css/dark-gradient.css
- Switch the theme back to “Black Gradient Theme”

Lab: Workbench Styling

- Open up the org.eclipse.platform project and open the plugin.xml
- Go to the extensions tab and expand org.eclipse.e4.ui.css.swt.theme
- These are the themes that come with e4

Extension Point „Theme“



Define extensions for this plug-in in the following section.

type filter text

- org.eclipse.core.runtime.products
- org.eclipse.ui.intro
- org.eclipse.ui.cheatsheets.cheatShee
- org.eclipse.ui.intro.configExtension
- org.eclipse.ui.actionSets
- org.eclipse.e4.ui.css.swt.theme
  - Default Theme (theme)
  - Classic (theme)
  - GTK (theme)
  - Mac (theme)
  - Win7 (theme)
  - Win XP Blue (theme)
  - Win XP Olive (theme)
  - Win 7 Classic (theme)

Extension Element Details

Set the properties of "theme". Required fields are denoted by "*".

id*:	org.eclipse.e4.ui.css.theme.e4_default
label*:	Win XP Blue
basestylesheeturi*:	css/e4_default_winxp_blu.css
os:	win32
ws:	
os_version:	

e4 Stylesheets

- In the org.eclipse.platform project, the 'css' folder contains the various stylesheets used by the workbench
- Let's take a closer look at one of them:
- e4_default_win.css

e4_defaut_win7.css

```
MTrimmedWindow {
    background-color: #E1E6F6;
    margin-top: 2px;
    margin-bottom: 2px;
    margin-left: 2px;
    margin-right: 2px;
}

.MTrimmedWindow.topLevel {
    margin-top: 24px;
    margin-bottom: 2px;
    margin-left: 12px;
    margin-right: 12px;
}
```

e4_defaut_win7.css

```
.MPartStack {
         tab-renderer:
        url('platform:/…..CTabRendering');
        unselected-tabs-color:
          #FFFFFF #FFFFFF #FFFFFF 100% 100%;
        outer-keyline-color: #FFFFFF;
        inner-keyline-color: #FFFFFF;
        Font-size: 9;
        font-family: 'Segoe UI';
        simple: true;
}
```

e4_defaut_win7.css

```
.MPartStack.active {
	unselected-tabs-color: #F3F9FF #D0DFEE
#CEDDED #CEDDED #D2E1F0 #D2E1F0
#FFFFFF 20% 45% 60% 70% 100% 100%;
	outer-keyline-color: #B6BCCC;
	inner-keyline-color: #FFFFFF;
	tab-outline: #B6BCCC;
}
```

e4_defaut_win7.css

```
.MTrimBar {
   background-color: #E1E6F6;
}

.MTrimBar#org-eclipse-ui-main-toolbar {
   background-image:  url(./win7.PNG);
}

#PerspectiveSwitcher  {
        background-color: #F5F7FC #E1E6F6 100%;
        perspective-keyline-color: #AAB0BF #AAB0BF;
}
```

Lab: Make a Theme (1)

- Make a copy of e4_default_win7.css and call it e4_mytheme.css
- Open the plugin.xml, go to the extensions tab, select the org.eclipse.e4.ui.css.swt.theme extension point, right click>New>theme
- Change the id and label, and for the basestyleuri field, browse to the location of your stylesheet and
  select it (leave os, ws and os_version for now)

Lab: Make a Theme (2)

- Run as>Eclipse Application
- Once the inner workbench opens up, go to Window>Preferences>General>Appearance
- You will notice that your Theme shows up
- Select it and hit OK

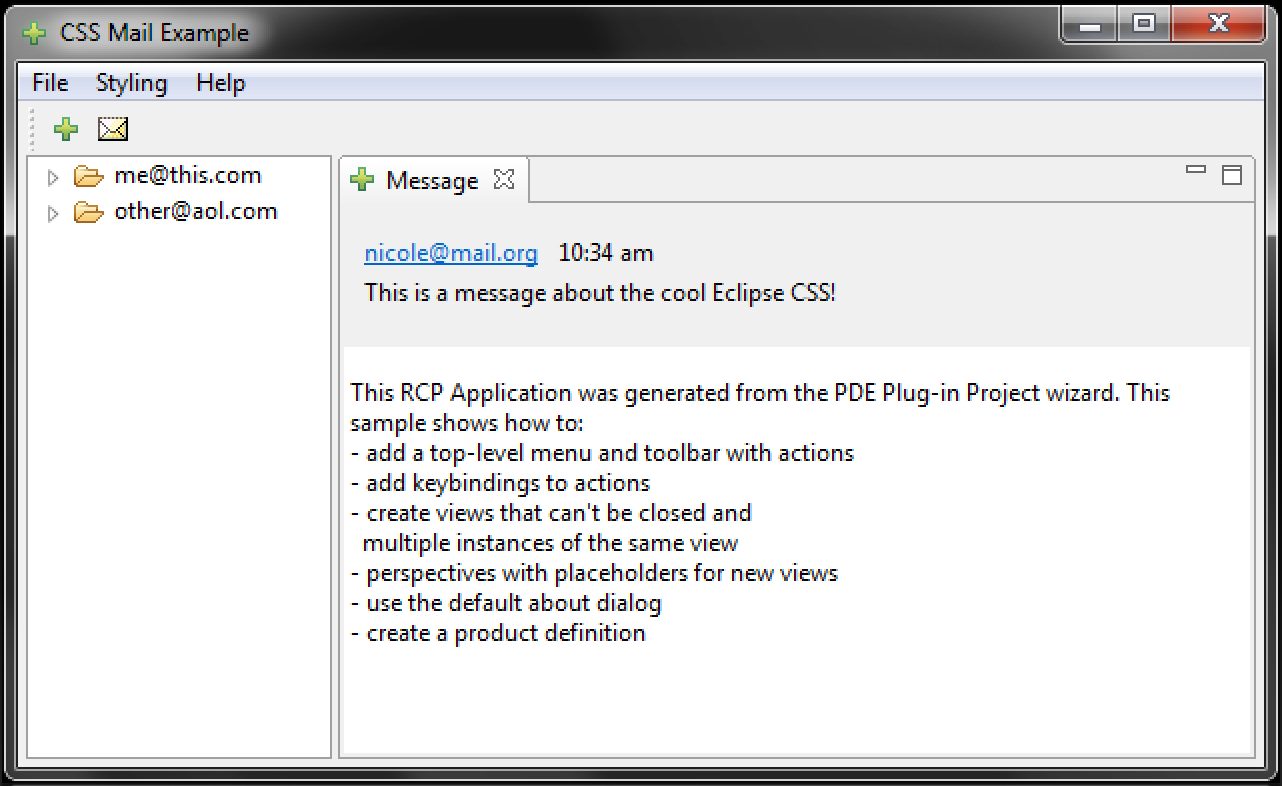
What Classnames to use?

- All widgets used in the workbench currently get a class name and id when created by a workbench renderer
- Currently, there is no spy type of application to get these class names and ids (nice to have)
- As a hack, you can add a println to setCSSInfo() in SWTPartRenderer

Styling an 3.x RCP Application

Picture from
http://www.sxc.hu/photo/1089931

3.X RCP Mail

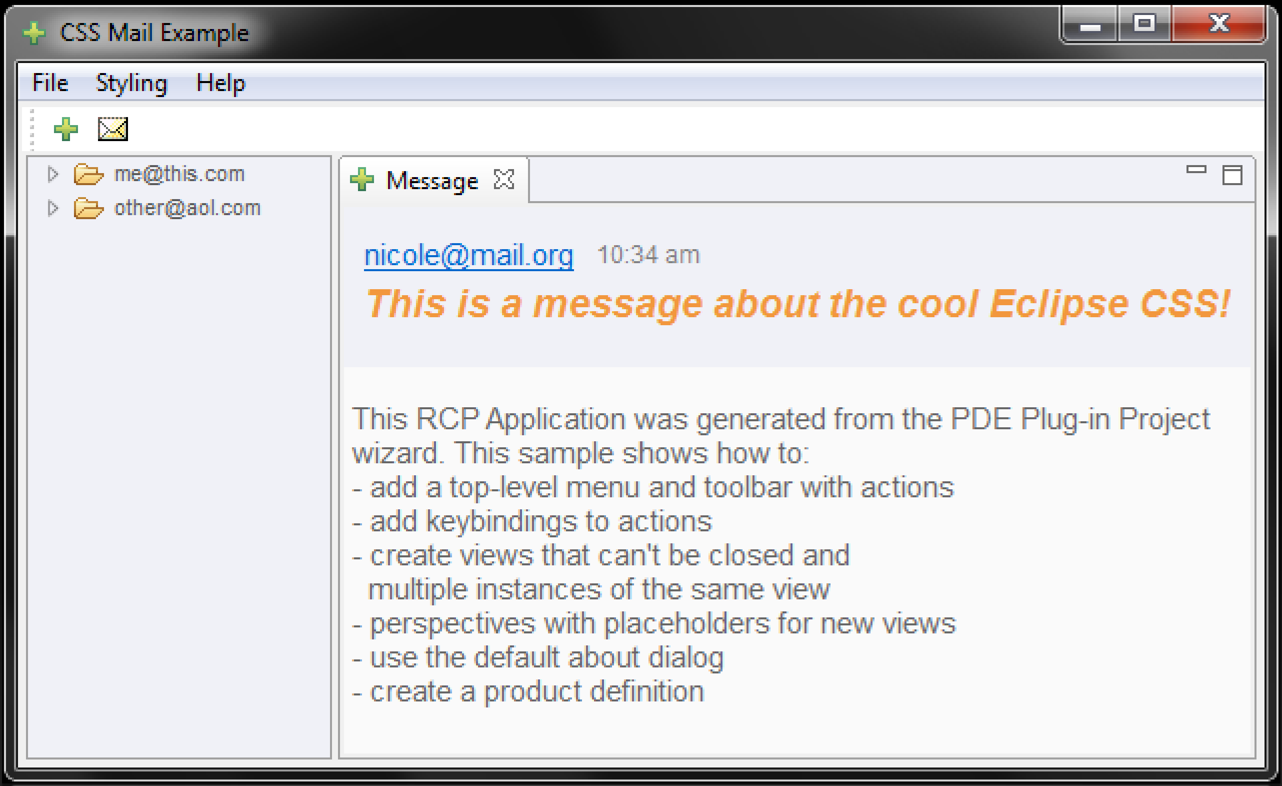


nicole@mail.org 10:34 am

This is a message about the cool Eclipse CSS!

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
- create a product definition

3.x RCP Mail with CSS Styling

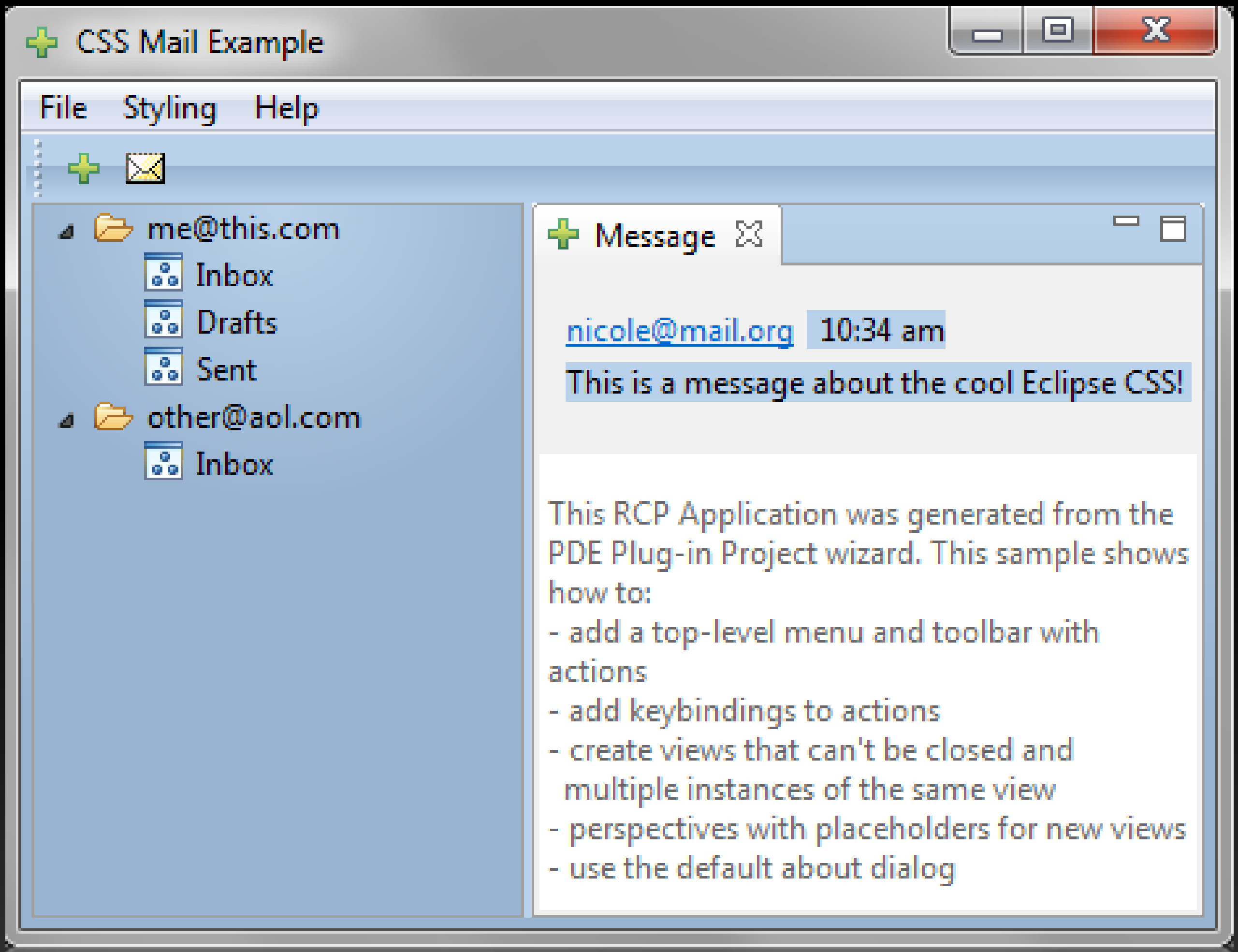


nicole@mail.org 10:34 am

This is a message about the cool Eclipse CSS!

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
- create a product definition

3.x RCP Mail with blue CSS Styling

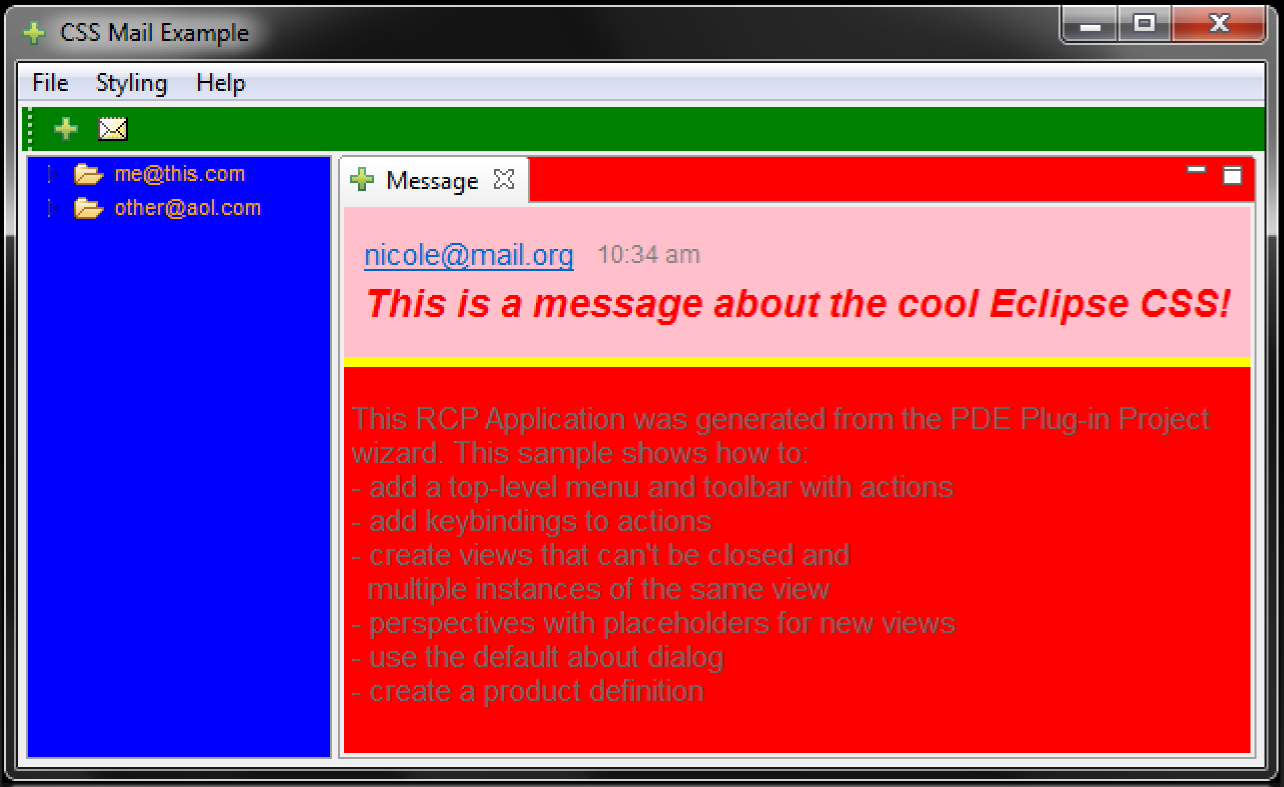


nicole@mail.org 10:34 am

This is a message about the cool Eclipse CSS!

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

3.x RCP Mail CSS Color Demonstrator



nicole@mail.org 10:34 am

This is a message about the cool Eclipse CSS!

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
- create a product definition

Needed Bundles

- org.apache.batik.css
- org.apache.batik.util
- org.apache.batik.util.gui
- org.apache.batik.xml
- org.eclipse.e4.ui.css.core
- org.eclipse.e4.ui.css.jface
- org.eclipse.e4.ui.css.legacy
- org.eclipse.e4.ui.css.swt
- org.eclipse.e4.ui.css.swt.theme
- org.eclipse.e4.ui.examples.css.rcp
- org.eclipse.e4.ui.widgets
- org.w3c.css.sac
- org.w3c.dom.smil
- org.w3c.dom.svg

The RCP Mail Demo
with CSS Styling

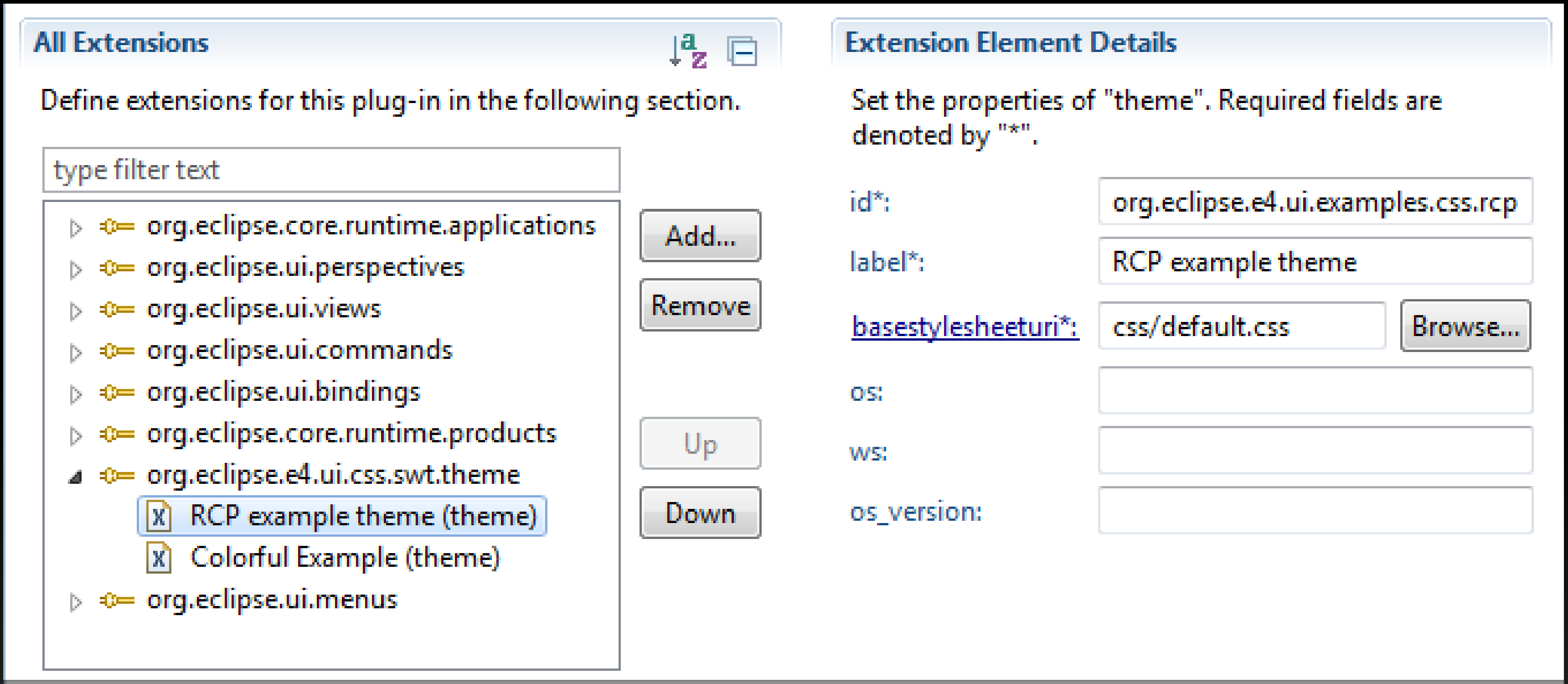
How to get the bundles

- Checkout :pserver:anonymous@dev.eclipse.org:/cvsroot/eclipse/e4/releng
- Import Project Set /releng/org.eclipse.e4.ui.releng/e4.ui.css.psf
- Checkout e4/org.eclipse.e4.ui/bundles/org.eclipse.e4.ui.widgets
- Delete all projects that are not on the previous slide
- Cleanup org.eclipse.e4.ui.css.swt
  - Organize Imports
  - Delete dependencies to org.eclipse.e4.core.*
- Start the CSS RCP Mail demo

How to enable CSS Styling in 3.x

- Provide a CSS theme
- Provide an extension to org.exlipse.e4.ui.css.swt.theme
- Enable theming support in the initialize method of ApplicationWorkbenchAdvisor

Extension Point for Themes



All Extensions

Define extensions for this plug-in in the following section.

type filter text

- org.eclipse.core.runtime.applications
- org.eclipse.ui.perspectives
- org.eclipse.ui.views
- org.eclipse.ui.commands
- org.eclipse.ui.bindings
- org.eclipse.core.runtime.products
- org.eclipse.e4.ui.css.swt.theme
  - RCP example theme (theme)
  - Colorful Example (theme)
- org.eclipse.ui.menus

Extension Element Details

Set the properties of "theme". Required fields are denoted by "*".

id*:	org.eclipse.e4.ui.examples.css.rcp
label*:	RCP example theme
basestylesheeturi*:	css/default.css
os:	
ws:	
os_version:	

ApplicationWorkbenchAdvisor

```
Bundle b = FrameworkUtil.getBundle(getClass());
BundleContext context = b.getBundleContext();
ServiceReference serviceRef = context
      .getServiceReference(IThemeManager.class.getName());
IThemeManager themeManager =
     (IThemeManager) context.getService(serviceRef);

final IThemeEngine engine =
     themeManager.getEngineForDisplay(Display.getCurrent());
engine.setTheme("org.eclipse.e4.ui.examples.css.rcp", true);
…
```

Lab: Play around with CSS Mail

- Install the Eclipse 3.7M6 SDK
- Import all projects from archive file “3xstyling.zip”
- Open plugin.xml of project org.eclipse.e4.ui.examples.css.rcp
- Start css mail application
- Play around with css theming

Discussion

Picture from
http://www.sxc.hu/photo/922004

License & Acknowledgements

Picture Index

Many thanks to the authors of the following pictures:

- Slide “UI Styling”: http://www.sxc.hu/photo/1089931
- Slide “Dynamic Theme Switching”: http://www.sxc.hu/photo/823108
- Slide “Discussion”: http://www.sxc.hu/photo/922004