

## NewEnumWizardPage.java – NewInterfaceWizardPage.java

```

1/*****
2 * Copyright (c) 2004, 2008 IBM Corporation and others.
3 * All rights reserved. This program and the accompanying materials
4 * are made available under the terms of the Eclipse Public License v1.0
5 * which accompanies this distribution, and is available at
6 * http://www.eclipse.org/legal/epl-v10.html
7 *
8 * Contributors:
9 *   IBM Corporation - initial API and implementation
10 *****/
11package org.eclipse.jdt.ui.wizards;
12
13import org.eclipse.swt.SWT;
14import org.eclipse.swt.layout.GridLayout;
15import org.eclipse.swt.widgets.Composite;
16
17import org.eclipse.core.runtime.IStatus;
18
19import org.eclipse.jface.dialogs.Dialog;
20import org.eclipse.jface.viewers.IStructuredSelection;
21
22import org.eclipse.ui.PlatformUI;
23
24import org.eclipse.jdt.core.IJavaElement;
25
26import org.eclipse.jdt.internal.ui.IJavaHelpContextIds;
27import org.eclipse.jdt.internal.ui.wizards.NewWizardMessages;
28
29/**
30 * Wizard page to create a new enum type.
31 * <p>
32 * Note: This class is not intended to be subclassed, but clients can instantiate.
33 * To implement a different kind of a new enum wizard page, extend
34 * <code>NewTypeWizardPage</code>.
35 * </p>
36 * @since 3.1
37 *
38 * @noextend This class is not intended to be subclassed by clients.
39 */
40public class NewEnumWizardPage extends NewTypeWizardPage {
41
42    private final static String PAGE_NAME= "NewEnumWizardPage"; //$NON-NLS-1$
43    private final static int TYPE = NewTypeWizardPage.ENUM_TYPE;
44
45    /**
46     * Create a new <code>NewEnumWizardPage</code>
47     */
48    public NewEnumWizardPage() {
49        super(TYPE, PAGE_NAME);
50
51        setTitle(NewWizardMessages.NewEnumWizardPage_title);
52        setDescription(NewWizardMessages.NewEnumWizardPage_description);
53    }
54
55    // ----- Initialization -----

```

```

1/*****
2 * Copyright (c) 2000, 2008 IBM Corporation and others.
3 * All rights reserved. This program and the accompanying materials
4 * are made available under the terms of the Eclipse Public License v1.0
5 * which accompanies this distribution, and is available at
6 * http://www.eclipse.org/legal/epl-v10.html
7 *
8 * Contributors:
9 *   IBM Corporation - initial API and implementation
10 *****/
11package org.eclipse.jdt.ui.wizards;
12
13import org.eclipse.swt.SWT;
14import org.eclipse.swt.layout.GridLayout;
15import org.eclipse.swt.widgets.Composite;
16
17import org.eclipse.core.runtime.IStatus;
18
19import org.eclipse.jface.dialogs.Dialog;
20import org.eclipse.jface.viewers.IStructuredSelection;
21
22import org.eclipse.ui.PlatformUI;
23
24import org.eclipse.jdt.core.IJavaElement;
25
26import org.eclipse.jdt.internal.ui.IJavaHelpContextIds;
27import org.eclipse.jdt.internal.ui.wizards.NewWizardMessages;
28
29/**
30 * Wizard page to create a new interface.
31 * <p>
32 * Note: This class is not intended to be subclassed, but clients can instantiate.
33 * To implement a different kind of a new interface wizard page, extend
34 * <code>NewTypeWizardPage</code>.
35 * </p>
36 * @since 2.0
37 *
38 * @noextend This class is not intended to be subclassed by clients.
39 */
40public class NewInterfaceWizardPage extends NewTypeWizardPage {
41
42    private final static String PAGE_NAME= "NewInterfaceWizardPage"; //$NON-NLS-1$
43
44    /**
45     * Create a new <code>NewInterfaceWizardPage</code>
46     */
47    public NewInterfaceWizardPage() {
48        super(false, PAGE_NAME);
49
50        setTitle(NewWizardMessages.NewInterfaceWizardPage_title);
51        setDescription(NewWizardMessages.NewInterfaceWizardPage_description);
52    }
53
54    // ----- Initialization -----

```

## NewEnumWizardPage.java - NewInterfaceWizardPage.java

```

56
57 /**
58  * The wizard owning this page is responsible for calling this method with the
59  * current selection. The selection is used to initialize the fields of the
60  * wizard
61  * page.
62  *
63  * @param selection used to initialize the fields
64  */
65 public void init(IStructuredSelection selection) {
66     IJavaElement jelem= getInitialJavaElement(selection);
67
68     initContainerPage(jelem);
69     initTypePage(jelem);
70
71     doStatusUpdate();
72 }
73 // ----- validation -----
74
75 private void doStatusUpdate() {
76     // all used component status
77     IStatus[] status= new IStatus[] {
78         fContainerStatus,
79         isEnclosingTypeSelected() ? fEnclosingTypeStatus : fPackageStatus,
80         fTypeNameStatus,
81         fModifierStatus,
82         fSuperInterfacesStatus
83     };
84
85     // the mode severe status will be displayed and the OK button
86     enabled/disabled.
87     updateStatus(status);
88 }
89
90 /**
91  * @see NewContainerWizardPage#handleFieldChanged
92  */
93 protected void handleFieldChanged(String fieldName) {
94     super.handleFieldChanged(fieldName);
95
96     doStatusUpdate();
97 }
98
99 // ----- UI -----
100
101 /**
102  * @see WizardPage#createControl
103  */
104 public void createControl(Composite parent) {
105     initializeDialogUnits(parent);
106
107     Composite composite= new Composite(parent, SWT.NONE);
108     composite.setFont(parent.getFont());
109

```

```

55
56 /**
57  * The wizard owning this page is responsible for calling this method with the
58  * current selection. The selection is used to initialize the fields of the
59  * wizard
60  * page.
61  *
62  * @param selection used to initialize the fields
63  */
64 public void init(IStructuredSelection selection) {
65     IJavaElement jelem= getInitialJavaElement(selection);
66
67     initContainerPage(jelem);
68     initTypePage(jelem);
69
70     doStatusUpdate();
71 }
72 // ----- validation -----
73
74 private void doStatusUpdate() {
75     // all used component status
76     IStatus[] status= new IStatus[] {
77         fContainerStatus,
78         isEnclosingTypeSelected() ? fEnclosingTypeStatus : fPackageStatus,
79         fTypeNameStatus,
80         fModifierStatus,
81         fSuperInterfacesStatus
82     };
83
84     // the mode severe status will be displayed and the OK button
85     enabled/disabled.
86     updateStatus(status);
87 }
88
89 /**
90  * @see NewContainerWizardPage#handleFieldChanged
91  */
92 protected void handleFieldChanged(String fieldName) {
93     super.handleFieldChanged(fieldName);
94
95     doStatusUpdate();
96 }
97
98 // ----- UI -----
99
100 /**
101  * @see WizardPage#createControl
102  */
103 public void createControl(Composite parent) {
104     initializeDialogUnits(parent);
105
106     Composite composite= new Composite(parent, SWT.NONE);
107     composite.setFont(parent.getFont());

```

## NewEnumWizardPage.java – NewInterfaceWizardPage.java

```

110
111     int nColumns= 4;
112
113     GridLayout layout= new GridLayout();
114     layout.numColumns= nColumns;
115     composite.setLayout(layout);
116
117     createContainerControls(composite, nColumns);
118     createPackageControls(composite, nColumns);
119     createEnclosingTypeControls(composite, nColumns);
120
121     createSeparator(composite, nColumns);
122
123     createTypeNameControls(composite, nColumns);
124     createModifierControls(composite, nColumns);
125
126     createSuperInterfacesControls(composite, nColumns);
127
128     createCommentControls(composite, nColumns);
129     enableCommentControl(true);
130
131     setControl(composite);
132
133     Dialog.applyDialogFont(composite);
134     PlatformUI.getWorkbench().getHelpSystem().setHelp(composite,
135     IJavaHelpContextIds.NEW_ENUM_WIZARD_PAGE);
136
137     /*
138     * @see WizardPage#becomesVisible
139     */
140     public void setVisible(boolean visible) {
141         super.setVisible(visible);
142         if (visible) {
143             setFocus();
144         }
145     }
146
147 }
148

```

```

108
109     int nColumns= 4;
110
111     GridLayout layout= new GridLayout();
112     layout.numColumns= nColumns;
113     composite.setLayout(layout);
114
115     createContainerControls(composite, nColumns);
116     createPackageControls(composite, nColumns);
117     createEnclosingTypeControls(composite, nColumns);
118
119     createSeparator(composite, nColumns);
120
121     createTypeNameControls(composite, nColumns);
122     createModifierControls(composite, nColumns);
123
124     createSuperInterfacesControls(composite, nColumns);
125
126     createCommentControls(composite, nColumns);
127     enableCommentControl(true);
128
129     setControl(composite);
130
131     Dialog.applyDialogFont(composite);
132     PlatformUI.getWorkbench().getHelpSystem().setHelp(composite,
133     IJavaHelpContextIds.NEW_INTERFACE_WIZARD_PAGE);
134
135     /*
136     * @see WizardPage#becomesVisible
137     */
138     public void setVisible(boolean visible) {
139         super.setVisible(visible);
140         if (visible) {
141             setFocus();
142         }
143     }
144
145 }
146

```