

FindReferencesInHierarchyAction.java - FindReferencesInProjectAction.java

```

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.actions;
12
13import org.eclipse.ui.IWorkbenchSite;
14import org.eclipse.ui.PlatformUI;
15
16import org.eclipse.jdt.core.ICompilationUnit;
17import org.eclipse.jdt.core.IField;
18import org.eclipse.jdt.core.IJavaElement;
19import org.eclipse.jdt.core.ILocalVariable;
20import org.eclipse.jdt.core.IMethod;
21import org.eclipse.jdt.core.IType;
22import org.eclipse.jdt.core.ITypeParameter;
23import org.eclipse.jdt.core.JavaModelException;
24import org.eclipse.jdt.core.search.IJavaSearchScope;
25import org.eclipse.jdt.core.search.SearchEngine;
26
27import org.eclipse.jdt.ui.search.ElementQuerySpecification;
28import org.eclipse.jdt.ui.search.QuerySpecification;
29
30import org.eclipse.jdt.internal.ui.IJavaHelpContextIds;
31import org.eclipse.jdt.internal.ui.JavaPluginImages;
32import org.eclipse.jdt.internal.ui.javaeditor.JavaEditor;
33import org.eclipse.jdt.internal.ui.search.JavaSearchScopeFactory;
34import org.eclipse.jdt.internal.ui.search.SearchMessages;
35
36/**
37 * Finds references of the selected element in its hierarchy.
38 * The action is applicable to selections representing a Java element.
39 *
40 * <p>
41 * This class may be instantiated; it is not intended to be subclassed.
42 * </p>
43 *
44 * @since 2.0
45 *
46 * @noextend This class is not intended to be subclassed by clients.
47 */
48public class FindReferencesInHierarchyAction extends FindReferencesAction {
49
50    /**
51     * Creates a new <code>FindReferencesInHierarchyAction</code>. The action

```

```

1/*****
2 * Copyright (c) 2000, 2006 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.erlide.ui.search;
12
13import org.eclipse.core.resources.IFile;
14import org.eclipse.core.resources.IProject;
15import org.eclipse.ui.IEditorInput;
16import org.eclipse.ui.IFileEditorInput;
17import org.eclipse.ui.IWorkbenchSite;
18import org.erlide.core.erlang.IErElement;
19import org.erlide.ui.editors.erl.ErlangEditor;
20
21/**
22 * Finds references to the selected element in the enclosing project of the
23 * selected element. The action is applicable to selections representing a Java
24 * element.
25 *
26 * <p>
27 * This class may be instantiated; it is not intended to be subclassed.
28 * </p>
29 *
30 * @since 3.0
31 */
32public class FindReferencesInProjectAction extends FindReferencesAction {
33
34    /**
35     * Creates a new <code>FindReferencesInProjectAction</code>. The action

```

FindReferencesInHierarchyAction.java - FindReferencesInProjectAction.java

```

52  * requires that the selection provided by the site's selection provider is of
type
53  * <code>org.eclipse.jface.viewers.IStructuredSelection</code>.
54  *
55  * @param site the site providing context information for this action
56  */
57  public FindReferencesInHierarchyAction(IWorkbenchSite site) {
58      super(site);
59  }
60
61  /**
62   * Note: This constructor is for internal use only. Clients should not call
this constructor.
63   * @param editor the Java editor
64   *
65   * @noreference This constructor is not intended to be referenced by clients.
66  */
67  public FindReferencesInHierarchyAction(JavaEditor editor) {
68      super(editor);
69  }
70
71  Class[] getValidTypes() {
72      return new Class[] { ICompilationUnit.class, IType.class, IMethod.class,
IField.class, ILocalVariable.class, ITypeParameter.class };
73  }
74
75  void init() {
76      setText(SearchMessages.Search_FindHierarchyReferencesAction_label);
77      setToolTipText(SearchMessages.Search_FindHierarchyReferencesAction_tooltip);
78      setImageDescriptor(JavaPluginImages.DESC_OBJS_SEARCH_REF);
79      PlatformUI.getWorkbench().getHelpSystem().setHelp(this,
IJavaHelpContextIds.FIND_REFERENCES_IN_HIERARCHY_ACTION);
80  }
81
82  QuerySpecification createQuery(IJavaElement element) throws
JavaModelException, InterruptedException {
83      IType type= getType(element);
84      if (type == null) {
85          return super.createQuery(element);
86      }
87      JavaSearchScopeFactory factory= JavaSearchScopeFactory.getInstance();
88      IJavaSearchScope scope= SearchEngine.createHierarchyScope(type);
89      String description= factory.getHierarchyScopeDescription(type);
90      return new ElementQuerySpecification(element, getLimitTo(), scope,
description);
91  }

```

```

36  * requires that the selection provided by the site's selection provider is
of type <code>IStructuredSelection</code>.
37  *
38  *
39  * @param site
the site providing context information for this action
40  *
41  */
42  public FindReferencesInProjectAction(final IWorkbenchSite site) {
43      super(site);
44  }
45
46  /**
47   * Note: This constructor is for internal use only. Clients should not call
this constructor.
48   *
49   *
50   * @param editor
the Java editor
51   *
52  */
53  public FindReferencesInProjectAction(final ErlangEditor editor) {
54      super(editor);
55  }
56
57  @Override
58  Class<?>[] getValidTypes() {
59      return new Class[] { IErlElement.class };
60  }
61
62  @Override
63  void init() {
64      setText("Project");
65      setToolTipText("Find references in selected projects");
66      // FIXME setImageDescriptor(JavaPluginImages.DESC_OBJS_SEARCH_REF);
67      // FIXME PlatformUI.getWorkbench().getHelpSystem().setHelp(this,
IJavaHelpContextIds.FIND_REFERENCES_IN_PROJECT_ACTION);
68
69  }
70
71  @Override
72  protected String[] getScope() {
73      final IEditorInput editorInput = getEditor().getEditorInput();
74      if (editorInput instanceof IFileEditorInput) {
75          final IFileEditorInput input = (IFileEditorInput) editorInput;
76          final IFile file = input.getFile();
77          final IProject project = file.getProject();
78          return SearchUtil.getProjectScope(project);
79      }
80      return null;
81  }

```

FindReferencesInHierarchyAction.java - FindReferencesInProjectAction.java

92

93}

94

82

```
83 // QuerySpecification createQuery(IJavaElement element)
84 // throws JavaModelException {
85 //   JavaSearchScopeFactory factory = JavaSearchScopeFactory.getInstance();
86 //   JavaEditor editor = getEditor();
87 //   //
88 //   IJavaSearchScope scope;
89 //   String description;
90 //   final boolean isInsideJRE = factory.isInsideJRE(element);
91 //   if (editor != null) {
92 //     scope = factory.createJavaProjectSearchScope(editor
93 //       .getEditorInput(), isInsideJRE);
94 //     description = factory.getProjectScopeDescription(editor
95 //       .getEditorInput(), isInsideJRE);
96 //   } else {
97 //     scope = factory.createJavaProjectSearchScope(element
98 //       .getJavaProject(), isInsideJRE);
99 //     description = factory.getProjectScopeDescription(element
100 //       .getJavaProject(), isInsideJRE);
101 //   }
102 //   return new ElementQuerySpecification(element, getLimitTo(), scope,
103 //     description);
104 // }
```

105

106}

107