

CodeAssistPreferencePage.java - CodeAssistAdvancedPreferencePage.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 *****/
11
12package org.eclipse.jdt.internal.ui.preferences;
13
14import org.eclipse.swt.widgets.Composite;
15import org.eclipse.swt.widgets.Control;
16
17import org.eclipse.core.resources.IProject;
18
19import org.eclipse.ui.PlatformUI;
20import org.eclipse.ui.preferences.IWorkbenchPreferenceContainer;
21
22import org.eclipse.jdt.internal.ui.IJavaHelpContextIds;
23
24
25/**
26 * Content Assist preference page.
27 * <p>
28 * Note: Must be public since it is referenced from plugin.xml
29 * </p>
30 *
31 * @since 3.0
32 */
33public class CodeAssistPreferencePage extends PropertyAndPreferencePage {
34
35    private CodeAssistConfigurationBlock fConfigurationBlock;
36
37    public void createControl(Composite parent) {
38        IWorkbenchPreferenceContainer container= (IWorkbenchPreferenceContainer)
39        getContainer();
40        fConfigurationBlock= new
41        CodeAssistConfigurationBlock(getNewStatusChangeListener(), container);
42
43        super.createControl(parent);
44        PlatformUI.getWorkbench().getHelpSystem().setHelp(getControl(),
45        IJavaHelpContextIds.JAVA_EDITOR_PREFERENCE_PAGE);
46    }
47
48    protected Control createPreferenceContent(Composite composite) {
49        return fConfigurationBlock.createContents(composite);
50    }
51
52    protected boolean hasProjectSpecificOptions(IProject project) {
53        return false;
54    }
55
56}

```

```

1/*****
2 * Copyright (c) 2005, 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 *****/
11
12package org.eclipse.jdt.internal.ui.preferences;
13
14import org.eclipse.swt.widgets.Composite;
15import org.eclipse.swt.widgets.Control;
16
17import org.eclipse.core.resources.IProject;
18
19import org.eclipse.ui.PlatformUI;
20import org.eclipse.ui.preferences.IWorkbenchPreferenceContainer;
21
22import org.eclipse.jdt.internal.ui.IJavaHelpContextIds;
23
24
25public final class CodeAssistAdvancedPreferencePage extends
26    PropertyAndPreferencePage {
27
28    private CodeAssistAdvancedConfigurationBlock fConfigurationBlock;
29
30
31    public void createControl(Composite parent) {
32        IWorkbenchPreferenceContainer container= (IWorkbenchPreferenceContainer)
33        getContainer();
34        fConfigurationBlock= new
35        CodeAssistAdvancedConfigurationBlock(getNewStatusChangeListener(), container);
36
37        super.createControl(parent);
38        PlatformUI.getWorkbench().getHelpSystem().setHelp(getControl(),
39        IJavaHelpContextIds.JAVA_EDITOR_PREFERENCE_PAGE);
40    }
41
42    protected Control createPreferenceContent(Composite composite) {
43        return fConfigurationBlock.createContents(composite);
44    }
45
46    protected boolean hasProjectSpecificOptions(IProject project) {
47        return false;
48    }
49
50}

```

```

52
53 protected String getPreferencePageID() {
54     return "org.eclipse.jdt.ui.preferences.CodeAssistPreferencePage";
55     //$NON-NLS-1$
56 }
57
58 protected String getPropertyPageID() {
59     return null;
60 }
61
62 /* (non-Javadoc)
63  * @see org.eclipse.jface.dialogs.DialogPage#dispose()
64  */
65 public void dispose() {
66     if (fConfigurationBlock != null) {
67         fConfigurationBlock.dispose();
68     }
69     super.dispose();
70 }
71
72 /*
73  * @see org.eclipse.jface.preference.IPreferencePage#performDefaults()
74  */
75 protected void performDefaults() {
76     super.performDefaults();
77     if (fConfigurationBlock != null) {
78         fConfigurationBlock.performDefaults();
79     }
80 }
81
82 /*
83  * @see org.eclipse.jface.preference.IPreferencePage#performOk()
84  */
85 public boolean performOk() {
86     if (fConfigurationBlock != null && !fConfigurationBlock.performOk()) {
87         return false;
88     }
89     return super.performOk();
90 }
91
92 /*
93  * @see org.eclipse.jface.preference.IPreferencePage#performApply()
94  */
95 public void performApply() {
96     if (fConfigurationBlock != null) {
97         fConfigurationBlock.performApply();
98     }
99 }
100
101

```

```

42
43 protected String getPreferencePageID() {
44     return "org.eclipse.jdt.ui.preferences.CodeAssistPreferenceAdvanced";
45     //$NON-NLS-1$
46 }
47
48 protected String getPropertyPageID() {
49     return null;
50 }
51
52 /* (non-Javadoc)
53  * @see org.eclipse.jface.dialogs.DialogPage#dispose()
54  */
55 public void dispose() {
56     if (fConfigurationBlock != null) {
57         fConfigurationBlock.dispose();
58     }
59     super.dispose();
60 }
61
62 /*
63  * @see org.eclipse.jface.preference.IPreferencePage#performDefaults()
64  */
65 protected void performDefaults() {
66     super.performDefaults();
67     if (fConfigurationBlock != null) {
68         fConfigurationBlock.performDefaults();
69     }
70 }
71
72 /*
73  * @see org.eclipse.jface.preference.IPreferencePage#performOk()
74  */
75 public boolean performOk() {
76     if (fConfigurationBlock != null && !fConfigurationBlock.performOk()) {
77         return false;
78     }
79     return super.performOk();
80 }
81
82 /*
83  * @see org.eclipse.jface.preference.IPreferencePage#performApply()
84  */
85 public void performApply() {
86     if (fConfigurationBlock != null) {
87         fConfigurationBlock.performApply();
88     }
89 }
90

```