JFace - Context Menu – One Trace
Use Concept Trace Slicing

```
Supporting Applications : [Console]

import org.eclipse.jface.text.IDocumentPartitioner;
import org.eclipse.jface.util.PropertyChangeEvent;
import org.eclipse.jface.action.IContributionManager;
import org.eclipse.swt.custom.StyledText;
import org.eclipse.jface.text.Document;
import org.eclipse.jface.action.Separator;
import org.eclipse.jface.text.IFindReplaceTarget;
import org.eclipse.jface.action.IMenuListener;
import org.eclipse.jface.viewers.ISelectionProvider;
import org.eclipse.jface.text.IPositionUpdater;
import org.eclipse.jface.text.IDocument;
import org.eclipse.jface.util.IPropertyChangeListener;
import org.eclipse.swt.widgets.Control;
import org.eclipse.jface.resource.ImageDescriptor;
import org.eclipse.jface.text.IDocumentPartitionerExtension;
import org.eclipse.swt.widgets.Menu;
import org.eclipse.jface.viewers.ISelectionChangedListener;
import org.eclipse.jface.resource.JFaceResources;
import org.eclipse.jface.text.ITextViewer;
import org.eclipse.jface.text.IDocumentListener;
import org.eclipse.jface.action.MenuManager;
import org.eclipse.jface.text.ITextListener;
import org.eclipse.jface.text.ITextOperationTarget;
import org.eclipse.jface.action.Action;

public class AppMenuListener
implements IMenuListener  {

      public void menuAboutToShow(menuManager) {
            Separator separator = new Separator(String)||();      // REPEATED!
            IDocument idocument = itextOperationTarget.getDocument();   // REPEATED!
            AppAction appAction = new AppAction(iselectionProvider && itextOperationTarget);
            menuManager.add(separator && appAction);  // REPEATED!
      }
}

public class AppAction
extends Action {

      public  AppAction() {
            boolean app_boolean = appAction.isEnabled();
            appAction.setEnabled(app_boolean);
            appAction.firePropertyChange(String,Object,Object):()||(PropertyChangeEvent):();
            appAction.setToolTipText(String);
            appAction.setHoverImageDescriptor(ImageDescriptor);
            appAction.setDisabledImageDescriptor(ImageDescriptor);
            appAction.setImageDescriptor(ImageDescriptor);
            /*                      Cyclic Statements                   */
            IFindReplaceTarget ifindReplaceTarget = itextOperationTarget.getFindReplaceTarget();
            boolean app_boolean1 = itextOperationTarget.canDoOperation(int);
            ITextOperationTarget itextOperationTarget = ITextViewer.getTextOperationTarget();
      }
}

public class AppDocumentListener
implements IDocumentListener  {
}

public class AppPropertyChangeListener
implements IPropertyChangeListener  {

      public  AppPropertyChangeListener() {
            AppSelectionChangedListener appSelectionChangedListener = new AppSelectionChangedListener(appPropertyChangeListener);
            AppTextListener appTextListener = new AppTextListener(appPropertyChangeListener);
      }
}

public class AppDocumentPartitioner
implements IDocumentPartitioner  {
}

public class AppDocumentPartitionerExtension
implements IDocumentPartitionerExtension  {
}

public class AppSelectionChangedListener
implements ISelectionChangedListener  {
}

public class AppTextListener
implements ITextListener  {
}

public class AppPositionUpdater
implements IPositionUpdater  {
}

public class SomeClass {

      public void someMethod() {
            AppDocumentPartitioner appDocumentPartitioner = new AppDocumentPartitioner();
            AppDocumentPartitionerExtension appDocumentPartitionerExtension = new AppDocumentPartitionerExtension();
            AppPropertyChangeListener appPropertyChangeListener = new AppPropertyChangeListener();
            AppMenuListener appMenuListener = new AppMenuListener(appPropertyChangeListener);
            AppDocumentListener appDocumentListener = new AppDocumentListener();
            AppPositionUpdater appPositionUpdater = new AppPositionUpdater();
            MenuManager menuManager = new MenuManager(String)||(String,String)||(); // REPEATED!
            menuManager.setRemoveAllWhenShown(boolean);     // REPEATED!
            menuManager.addMenuListener(appMenuListener);   // REPEATED!
            Document document = new Document(String)||();   // REPEATED!
            document.addPositionCategory(String);     // REPEATED!
            appDocumentPartitionerExtension.connect(document);    // REPEATED!
            ITextViewer.setDocument(document);  // REPEATED!
            JFaceResources.getFontRegistry();   // REPEATED!
            JFaceResources.getColorRegistry();  // REPEATED!
            document.addListener(appPropertyChangeListener);      // REPEATED!
            document.removeListener(appPropertyChangeListener);   // REPEATED!
            /*                      Cyclic Statements                   */
```

```
            idocument.addPositionUpdater(appPositionUpdater);     // REPEATED!
            Menu menu = menuManager.createContextMenu(styledText && control); // REPEATED!
            appAction.setText(String);    // REPEATED!
            IContributionManager.appendToGroup(appAction);  // REPEATED!
            appAction.setActionDefinitionId(String);  // REPEATED!
            appAction.setDescription(String);    // REPEATED!
            appAction.setToolTipText(String);    // REPEATED!
            idocument.addDocumentListener(appDocumentListener);   // REPEATED!
            iselectionProvider.removeSelectionChangedListener(appSelectionChangedListener);     // REPEATED!
            AppAction appAction = new AppAction();
            appAction.setImageDescriptor(ImageDescriptor);  // REPEATED!
            ISelectionProvider iselectionProvider = itextOperationTarget.getSelectionProvider();      // REPEATED!
            iselectionProvider.addTextListener(appTextListener);  // REPEATED!
            menuManager.dispose();  // REPEATED!
            IDocument idocument = iselectionProvider.getDocument();     // REPEATED!
            StyledText styledText = iselectionProvider.getTextWidget(); // REPEATED!
            Control control = iselectionProvider.getControl();    // REPEATED!
            iselectionProvider.removeTextListener(appTextListener);     // REPEATED!
            iselectionProvider.addSelectionChangedListener(appSelectionChangedListener);  // REPEATED!
            int app_int = idocument.getLength();      // REPEATED!
        }
}
```

Description of False Negatives:

The following instruction is missing (One false negatives) since it resides in org.eclipse.ui, not in JFace:

```
control.setMenu(menu);
```