JFace – Content Assist – Three Traces
Use Full Trace (No Slicing)

```
Supporting Applications : [JSP Editor, Example Java Editor, HTML Editor]

import org.eclipse.jface.text.contentassist.IContentAssistProcessor;
import org.eclipse.jface.text.IDocumentPartitioner;
import org.eclipse.jface.action.IAction;
import org.eclipse.jface.action.Separator;
import org.eclipse.jface.viewers.ISelectionProvider;
import org.eclipse.jface.action.IMenuManager;
import org.eclipse.jface.text.IDocument;
import org.eclipse.jface.text.source.IAnnotationHover;
import org.eclipse.jface.text.contentassist.ContentAssistant;
import org.eclipse.jface.text.IDocumentExtension3;
import org.eclipse.jface.text.source.SourceViewerConfiguration;
import org.eclipse.jface.viewers.ISelectionChangedListener;
import org.eclipse.swt.graphics.Color;
import org.eclipse.jface.text.ITextViewer;
import org.eclipse.jface.text.DocumentCommand;
import org.eclipse.jface.text.contentassist.ICompletionProposal;
import org.eclipse.jface.text.ITextHover;
import org.eclipse.jface.text.TextAttribute;
import org.eclipse.jface.text.IAutoEditStrategy;
import org.eclipse.jface.text.source.ISourceViewer;
import org.eclipse.jface.text.DefaultPositionUpdater;
import org.eclipse.jface.action.Action;

public class AppContentAssistProcessor
implements IContentAssistProcessor  {

      public ICompletionProposal[] computeCompletionProposals(ITextViewer,int) {
            CompletionProposal completionProposal = new CompletionProposal(String,int,int,int)||
                                                                          (String,int,int,int,Image,String,IContextInformation,String);
      }

      public String getErrorMessage() {
      }

      public char[] getCompletionProposalAutoActivationCharacters() {
      }

      public char[] getContextInformationAutoActivationCharacters() {
      }
}

public class AppAutoEditStrategy
implements IAutoEditStrategy  {

      public void customizeDocumentCommand(IDocument,DocumentCommand) {
      }
}

public class AppTextHover
implements ITextHover  {
}

public class AppAction
extends Action {

      public  AppAction() {
            appAction.setEnabled(boolean);
      }
}

public class AppSelectionChangedListener
implements ISelectionChangedListener  {
}

public class AppAnnotationHover
implements IAnnotationHover  {
}

public class SomeClass {

      public void someMethod() {
            IMenuManager imenuManager = IMenuManager.findMenuUsingPath(String);     // MAY REPEAT!
            AppSelectionChangedListener appSelectionChangedListener = new AppSelectionChangedListener();
            ISelectionProvider.addSelectionChangedListener(appSelectionChangedListener);  // MAY REPEAT!
            AppAnnotationHover appAnnotationHover = new AppAnnotationHover();
            SourceViewerConfiguration sourceViewerConfiguration = new SourceViewerConfiguration();
            DefaultPositionUpdater defaultPositionUpdater = new DefaultPositionUpdater(String);
            IAction.setActionDefinitionId(String);    // REPEATED!
            Separator separator = new Separator(String)||();      // REPEATED!
            imenuManager.add(separator);  // REPEATED!
            int app_int = IDocument.getLineOffset(int);     // MAY REPEAT!
            TextAttribute textAttribute = new TextAttribute(Color)||(Color,Color,int);
            AppAction appAction = new AppAction();
            IDocumentExtension3.setDocumentPartitioner(String,IDocumentPartitioner);      // MAY REPEAT!
            IDocumentPartitioner.connect(IDocument);  // MAY REPEAT!
            AppTextHover appTextHover = new AppTextHover();
            String string = sourceViewerConfiguration.getConfiguredDocumentPartitioning(ISourceViewer);
            ContentAssistant contentAssistant = new ContentAssistant();
            contentAssistant.setDocumentPartitioning(string);
            contentAssistant.enableAutoActivation(boolean);
            contentAssistant.setAutoActivationDelay(int);
            contentAssistant.setProposalPopupOrientation(int);
            contentAssistant.setContextInformationPopupOrientation(int);
            AppContentAssistProcessor appContentAssistProcessor = new AppContentAssistProcessor();
            contentAssistant.setContentAssistProcessor(appContentAssistProcessor);

      }
}
```

Description of False Negatives:

We have the following three false negatives. The first two are because of not implementing the following class since the third sample application did not implement it:

```
public class AppSourceViewerConfiguration
extends SourceViewerConfiguration {

      public IContentAssistant getContentAssistant(ISourceViewer) {
      }

}
```

The third false negative is because of not calling the following instruction since it is in org.eclipse.ui, not in JFace:

```
TextEditor.setSourceViewerConfiguration(appSourceViewerConfiguration);
```