

## Eclipse – Table Viewer – Three Traces Use Full Trace (No Slicing)

Supporting Applications : [LDAP Connections, Editor List, Sample Table Viewer]

```
import org.eclipse.ui.PlatformUI;
import java.text.Collator;
import org.eclipse.jface.viewers.LabelProvider;
import org.eclipse.jface.action.IMenuListener;
import org.eclipse.ui.IActionBars;
import org.eclipse.ui.IViewPart;
import org.eclipse.ui.IWorkbenchPartSite;
import org.eclipse.swt.widgets.Control;
import org.eclipse.ui.IWorkbenchPart;
import org.eclipse.jface.viewers.IStructuredContentProvider;
import org.eclipse.swt.widgets.Menu;
import org.eclipse.jface.viewers.ViewersSorter;
import org.eclipse.jface.viewers.TableViewer;
import org.eclipse.swt.widgets.Composite;
import org.eclipse.jface.viewers.StructuredViewer;
import org.eclipse.jface.viewers.IBaseLabelProvider;
import org.eclipse.ui.IViewSite;
import org.eclipse.jface.action.MenuManager;
import org.eclipse.jface.action.IToolBarManager;
import org.eclipse.swt.widgets.Table;
import org.eclipse.jface.action.Action;

public class AppWorkbenchPart
implements IWorkbenchPart {

    public setFocus() {
        Control control = tableViewer.getControl();
        boolean app_boolean = control.setFocus();
    }

    public void createPartControl(Composite) {
        LabelProvider labelProvider = new LabelProvider();
        tableViewer.setLabelProvider(IBaseLabelProvider);
        MenuManager menuManager = new MenuManager(String||(String,String)||());
        menuManager.setRemoveAllWhenShown(boolean);
        AppMenuListener appMenuListener = new AppMenuListener();
        menuManager.addMenuListener(appMenuListener);
        tableViewer.setContentProvider(appStructuredContentProvider);
        IViewSite iViewSite = IViewPart.getViewSite(); // MAY REPEAT!
        tableViewer.setInput(iViewSite); // MAY REPEAT!
        Control control = tableViewer.getControl(); // REPEATED!
        Menu menu = menuManager.createContextMenu(control);
        control.setMenu(menu);
        /* Cyclic Statements */
        IToolBarManager itoolBarManager = iactionBars.getToolBarManager(); // MAY REPEAT!
        itoolBarManager.add(appAction); // REPEATED!
        IActionBars iactionBars = iworkbenchPartSite.getActionBars(); // MAY REPEAT!
        IWorkbenchPartSite iworkbenchPartSite = IWorkbenchPart.getSite(); // REPEATED!
    }
}

public class AppStructuredContentProvider
implements IStructuredContentProvider {

    public void inputChanged(tableViewer) {
    }

    public Object[] getElements(Object) {
    }

    public void dispose() {
    }
}

public class AppAction
extends Action {
}

public class AppMenuListener
implements IMenuListener {
}

public class SomeClass {

    public void someMethod() {
        TableViewer tableViewer = new TableViewer(Composite,int)|||(Table)|||(Composite);
        AppStructuredContentProvider appStructuredContentProvider = new AppStructuredContentProvider();
        AppAction appAction = new AppAction();
        appAction.setText(String);
        appAction.setToolTipText(String);
        ViewersSorter viewersSorter = new ViewersSorter(Collator)||();
        StructuredViewer.setSorter(ViewersSorter);
    }
}
```

### Description of False Negatives:

The class AppWorkbenchPart should extend ViewPart. So, we have one false negative (i.e., **extends** ViewPart).