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

```
Supporting Applications : [LDAP Connections, Sample Table Viewer]

import org.eclipse.ui.ISharedImages;
import java.text.Collator;
import org.eclipse.jface.action.Separator;
import org.eclipse.jface.action.IMenuListener;
import org.eclipse.ui.IActionBars;
import org.eclipse.jface.action.IMenuManager;
import org.eclipse.jface.resource.ImageDescriptor;
import org.eclipse.jface.viewers.TableViewer;
import org.eclipse.swt.widgets.Composite;
import org.eclipse.ui.PlatformUI;
import org.eclipse.jface.viewers.LabelProvider;
import org.eclipse.ui.IViewPart;
import org.eclipse.ui.IWorkbenchPartSite;
import org.eclipse.swt.widgets.Control;
import org.eclipse.jface.viewers.IStructuredContentProvider;
import org.eclipse.ui.IWorkbenchPart;
import org.eclipse.swt.widgets.Menu;
import org.eclipse.jface.viewers.ViewerSorter;
import org.eclipse.jface.viewers.StructuredViewer;
import org.eclipse.ui.IViewSite;
import org.eclipse.jface.viewers.IDoubleClickListener;
import org.eclipse.jface.action.IToolBarManager;
import org.eclipse.jface.action.MenuManager;
import org.eclipse.swt.widgets.Table;
import org.eclipse.jface.action.Action;

public class AppWorkbenchPart
implements IWorkbenchPart  {

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

      public  createPartControl() {
            LabelProvider labelProvider = new LabelProvider();
            ViewerSorter viewerSorter = new ViewerSorter(Collator)||();
            StructuredViewer.setSorter(viewerSorter);
            Separator separator = new Separator(String)||();
            AppDoubleClickListener appDoubleClickListener = new AppDoubleClickListener();
            MenuManager menuManager = new MenuManager(String)||(String,String)||();
            menuManager.setRemoveAllWhenShown(boolean);
            AppMenuListener appMenuListener = new AppMenuListener();
            menuManager.addMenuListener(appMenuListener);
            tableViewer.setContentProvider(appStructuredContentProvider);
            tableViewer.setLabelProvider(labelProvider);
            /*                      Cyclic Statements                  */
            IToolBarManager itoolBarManager = iactionBars.getToolBarManager();
            Menu menu = menuManager.createContextMenu(control);
            control.setMenu(menu);
            IViewSite iviewSite = IViewPart.getViewSite();
            Control control = tableViewer.getControl();
            IMenuManager imenuManager = iactionBars.getMenuManager();
            imenuManager.add(appAction && separator);
            IActionBars iactionBars = iworkbenchPartSite.getActionBars();
            IWorkbenchPartSite iworkbenchPartSite = IWorkbenchPart.getSite();
            tableViewer.setInput(iviewSite);
      }
}

public class AppStructuredContentProvider
implements IStructuredContentProvider  {

      public void inputChanged(tableViewer && iworkbenchPartSite && iviewSite) {
      }

      public Object[] getElements(iviewSite) {
      }

      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);
            tableViewer.addDoubleClickListener(appDoubleClickListener);
            AppStructuredContentProvider appStructuredContentProvider = new AppStructuredContentProvider();
            AppAction appAction = new AppAction();
            appAction.setText(String);
            appAction.setToolTipText(String);
      }
}
```

Description of False Negatives:

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