

Eclipse – Tree Viewer – Two Traces

Use Full Trace (No Slicing)

Supporting Applications : [LDAP Connections, Sample Tree Viewer]

```
import org.eclipse.swt.graphics.Image;
import org.eclipse.jface.viewers.ITreeContentProvider;
import org.eclipse.jface.viewers.ViewerSorter;
import org.eclipse.jface.action.MenuManager;
import org.eclipse.jface.resource.ImageDescriptor;
import org.eclipse.ui.IViewSite;
import org.eclipse.ui.IWorkbenchPartSite;
import org.eclipse.ui.IViewPart;
import org.eclipse.jface.action.IToolBarManager;
import org.eclipse.swt.widgets.Menu;
import org.eclipse.jface.viewers.IDoubleClickListener;
import org.eclipse.jface.action.IMenuListener;
import org.eclipse.swt.widgets.Tree;
import org.eclipse.jface.action.IMenuManager;
import org.eclipse.jface.viewers.LabelProvider;
import org.eclipse.ui.PlatformUI;
import java.text.Collator;
import org.eclipse.swt.widgets.Composite;
import org.eclipse.swt.widgets.Control;
import org.eclipse.jface.action.Separator;
import org.eclipse.jface.action.Action;
import org.eclipse.ui.IWorkbenchPart;
import org.eclipse.core.runtime.IAdaptable;
import org.eclipse.ui.IActionBars;
import org.eclipse.jface.viewers.TreeViewer;

public class AppTreeContentProvider
implements ITreeContentProvider {

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

    public Object[] getElements(iviewSite) {
        Object[] objectArray1 = appTreeContentProvider.getChildren(Object);
    }

    public void dispose() {
    }
}

public class AppLabelProvider
extends LabelProvider {

    public Image getImage(appAdaptable) {
    }

    public String getText(appAdaptable) {
    }
}

public class AppWorkbenchPart
implements IWorkbenchPart {

    public void setFocus() {
        Control control = treeViewer.getControl();
        boolean app_boolean1 = control.setFocus();
    }

    public void createPartControl(Composite) {
        Separator separator = new Separator(String)();
        ViewerSorter viewerSorter = new ViewerSorter(Collator)();
        AppTreeContentProvider appTreeContentProvider = new AppTreeContentProvider();
        TreeViewer treeViewer = new TreeViewer(Composite,int)()(Tree)()(Composite);
        treeViewer.setSorter(viewerSorter);
        treeViewer.setContentProvider(appTreeContentProvider);
        AppLabelProvider appLabelProvider = new AppLabelProvider();
        treeViewer.setLabelProvider(appLabelProvider);
        MenuManager menuManager = new MenuManager(String)()(String,String)();
        Menu menu = menuManager.createContextMenu(control);
        control.setMenu(menu);
        menuManager.setRemoveAllWhenShown(boolean);
        AppDoubleClickListener appDoubleClickListener = new AppDoubleClickListener();
        AppMenuItemListener appMenuItemListener = new AppMenuItemListener();
        menuManager.addMenuItemListener(appMenuItemListener);
        /* Cyclic Statements */
        IWorkbenchPartSite iworkbenchPartSite = IWorkbenchPartPart.getSite();
        IActionBars iactionBars = iviewSite.getActionBars();
        IToolBarManager itoolBarManager = iactionBars.getToolBarManager();
        treeViewer.setInput(iviewSite);
        imenuManager.add(appAction && separator);
        IMenuManager imenuManager = iactionBars.getMenuManager();
        IViewSite iviewSite = IViewPart.getViewSite();
    }
}

public class AppAction
extends Action {
}

public class AppAdaptable
implements IAdaptable {
}

public class AppMenuItemListener
implements IMenuItemListener {
}

public class AppDoubleClickListener
implements IDoubleClickListener {
}

public class SomeClass {

    public void someMethod() {
        AppAction appAction = new AppAction();
        appAction.setText(String);
        appAction.setToolTipText(String);
        AppAdaptable appAdaptable = new AppAdaptable();
        treeViewer.addDoubleClickListener(appDoubleClickListener);
    }
}
```

Description of False Negatives:

- The class AppWorkbenchPart should extend ViewPart. So, we have one false negative for the instruction **extends ViewPart**.
- The class AppTreeContentProvider should also implement the method **getChildren()**. So, we have another false negative for this instruction.