

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

Supporting Applications : [ConcernMapper, Sample Tree Viewer, LDAP Browser]

```
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.swt.widgets.Composite;
import org.eclipse.jface.viewers.IBaseLabelProvider;
import org.eclipse.ui.PlatformUI;
import org.eclipse.swt.widgets.Tree;
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.ui.IWorkbenchPart;
import org.eclipse.swt.widgets.Menu;
import org.eclipse.jface.viewers.ViewerSorter;
import org.eclipse.ui.IViewSite;
import org.eclipse.jface.viewers.IDoubleClickListener;
import org.eclipse.jface.viewers.ITreeContentProvider;
import org.eclipse.jface.action.MenuManager;
import org.eclipse.jface.action.IToolBarManager;
import org.eclipse.jface.viewers.TreeViewer;
import org.eclipse.jface.action.Action;

public class AppTreeContentProvider
implements ITreeContentProvider {

    public void inputChanged(TreeViewer) {
    }

    public void dispose() {
    }

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

public class AppWorkbenchPart
implements IWorkbenchPart {

    public void createPartControl(Composite) {
        LabelProvider labelProvider = new LabelProvider();
        TreeViewer treeViewer = new TreeViewer(Composite,int)|||(Tree)|||(Composite);
        ViewerSorter viewerSorter = new ViewerSorter(Collator)||();
        treeViewer.setSorter(viewerSorter);
        AppTreeContentProvider appTreeContentProvider = new AppTreeContentProvider();
        treeViewer.setContentProvider(appTreeContentProvider);
        treeViewer.setLabelProvider(IBaseLabelProvider);
        Control control = treeViewer.getControl(); // REPEATED!
        Separator separator = new Separator(String)||(); // MAY REPEAT!
        AppDoubleClickListener appDoubleClickListener = new AppDoubleClickListener();
        MenuManager menuManager = new MenuManager(String)|||(String,String)||();
        menuManager.setRemoveAllWhenShown(boolean);
        Menu menu = menuManager.createContextMenu(control);
        control.setMenu(menu);
        AppMenuListener appMenuListener = new AppMenuListener();
        menuManager.addMenuListener(appMenuListener);
        IWorkbenchPartSite iworkbenchPartSite = IWorkbenchPart.getSite(); // REPEATED!
        IViewSite iviewSite = IViewPart.getViewSite(); // REPEATED!
        treeViewer.setInput(iviewSite);
        IActionBars iactionBars = iviewSite.getActionBars(); // MAY REPEAT!
        IMenuManager imenuManager = iactionBars.getMenuManager();
        IToolBarManager itoolBarManager = iactionBars.getToolBarManager();
        itoolBarManager.add(appAction && separator); // REPEATED!
    }

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

public class AppAction
extends Action {
}

public class AppDoubleClickListener
implements IDoubleClickListener {
}

public class AppMenuListener
implements IMenuListener {
}

public class SomeClass {

    public void someMethod() {
        treeViewer.addDoubleClickListener(appDoubleClickListener);
        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 negative for the instruction **extends ViewPart**.
- The class AppTreeContentProvider should also implement the method **getChildren()**. So, we have another false negative for this instruction.