

Eclipse – Navigation – Two Traces Use Full Trace (No Slicing)

Supporting Applications : [KTreeMap, SVN]

```
import org.eclipse.swt.widgets.Tree;
import org.eclipse.jface.viewers.IContentProvider;
import org.eclipse.jface.viewers.ISelection;
import org.eclipse.swt.widgets.Display;
import org.eclipse.jface.action.Separator;
import org.eclipse.jface.action.IMenuListener;
import org.eclipse.ui.IActionBars;
import org.eclipse.ui.IViewPart;
import org.eclipse.swt.widgets.Control;
import org.eclipse.swt.graphics.RGB;
import org.eclipse.ui.IWorkbenchPart;
import org.eclipse.swt.widgets.Menu;
import org.eclipse.swt.widgets.Composite;
import org.eclipse.jface.viewers.ISelectionChangedListener;
import org.eclipse.jface.viewers.LabelProvider;
import org.eclipse.ui.IViewSite;
import org.eclipse.jface.viewers.IStructuredSelection;
import org.eclipse.jface.viewers.ITreeContentProvider;
import org.eclipse.jface.viewers.SelectionChangedEvent;
import org.eclipse.jface.action.MenuManager;
import org.eclipse.jface.action.IToolBarManager;
import org.eclipse.ui.part.DrillDownAdapter;
import org.eclipse.jface.viewers.TreeViewer;
import org.eclipse.jface.action.Action;

public class AppTreeContentProvider
implements ITreeContentProvider {

    public boolean hasChildren(object) {
    }

    public Object[] getChildren(Object) {
    }

    public void inputChanged(object && treeViewer) {
    }
}

public class AppSelectionChangedListener
implements ISelectionChangedListener {

    public void selectionChanged(SelectionChangedEvent) {
    }
}

public class AppAction
extends Action {
}

public class AppMenuListener
implements IMenuListener {
}

public class AppWorkbenchPart
implements IWorkbenchPart {

    public void createPartControl(Composite) {
        TreeViewer treeViewer = new TreeViewer(Composite,int)|||(Tree)|||(Composite);
        AppTreeContentProvider appTreeContentProvider = new AppTreeContentProvider();
        treeViewer.setContentProvider(appTreeContentProvider);
        LabelProvider labelProvider = new LabelProvider();
        treeViewer.setLabelProvider(labelProvider);
        IViewSite iviewSite = IViewPart.getViewSite(); // MAY REPEAT!
        treeViewer.setInput(iviewSite);
        AppSelectionChangedListener appSelectionChangedListener = new AppSelectionChangedListener();
        treeViewer.addSelectionChangedListener(appSelectionChangedListener); // REPEATED!
        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);
        Separator separator = new Separator(String)||();
        AppAction appAction = new AppAction();
        IActionBars iactionBars = iviewSite.getActionBars(); // MAY REPEAT!
        IToolBarManager itoolBarManager = iactionBars.getToolBarManager();
        itoolBarManager.add(appAction && separator); // REPEATED!
        DrillDownAdapter drillDownAdapter = new DrillDownAdapter(treeViewer);
        drillDownAdapter.addNavigationActions(itoolBarManager);
    }

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

public class SomeClass {

    public void someMethod() {
        appAction.setToolTipText(String);
        Display.getCurrent();
        RGB rGB = new RGB(float,float,float)|||(int,int,int); // MAY REPEAT!
        ISelection iselection = treeViewer.getSelection();
        Display.getDefault();
    }
}
```