

LibraryFilter.java - ContainedLibraryFilter.java

```

1/*****
2 * Copyright (c) 2000, 2007 IBM Corporation and others.
3 * All rights reserved. This program and the accompanying materials
4 * are made available under the terms of the Eclipse Public License v1.0
5 * which accompanies this distribution, and is available at
6 * http://www.eclipse.org/legal/epl-v10.html
7 *
8 * Contributors:
9 *   IBM Corporation - initial API and implementation
10 *****/
11 package org.eclipse.jdt.internal.ui.viewsupport;
12
13 import org.eclipse.jface.viewers.Viewer;
14 import org.eclipse.jface.viewers.ViewerFilter;
15
16 import org.eclipse.jdt.core.IPackageFragmentRoot;
17
18 /**
19  * Filters out all elements which libraries
20  */
21 public class LibraryFilter extends ViewerFilter {
22
23     /**
24      * Returns the result of this filter, when applied to the
25      * given inputs.
26      *
27      * @param viewer the viewer
28      * @param parent the parent element
29      * @param element the element to test
30      * @return Returns true if element should be included in filtered set
31      */
32     public boolean select(Viewer viewer, Object parent, Object element) {
33         if (element instanceof IPackageFragmentRoot)
34             return !((IPackageFragmentRoot)element).isArchive();
35
36         return true;
37     }
38 }

```

```

1/*****
2 * Copyright (c) 2000, 2006 IBM Corporation and others.
3 * All rights reserved. This program and the accompanying materials
4 * are made available under the terms of the Eclipse Public License v1.0
5 * which accompanies this distribution, and is available at
6 * http://www.eclipse.org/legal/epl-v10.html
7 *
8 * Contributors:
9 *   IBM Corporation - initial API and implementation
10 *****/
11 package org.eclipse.jdt.internal.ui.filters;
12
13 import org.eclipse.core.resources.IProject;
14 import org.eclipse.core.resources.IResource;
15
16 import org.eclipse.jface.viewers.Viewer;
17 import org.eclipse.jface.viewers.ViewerFilter;
18
19 import org.eclipse.jdt.core.IPackageFragmentRoot;
20
21 /**
22  * The LibraryFilter is a filter used to determine whether
23  * a Java internal library is shown
24  */
25 public class ContainedLibraryFilter extends ViewerFilter {
26
27     /** (non-Javadoc)
28      * Method declared on ViewerFilter.
29      */
30
31     public boolean select(Viewer viewer, Object parentElement, Object element) {
32         if (element instanceof IPackageFragmentRoot) {
33             IPackageFragmentRoot root= (IPackageFragmentRoot)element;
34             if (root.isArchive()) {
35                 // don't filter out JARs contained in the project itself
36                 IResource resource= root.getResource();
37                 if (resource != null) {
38                     IProject jarProject= resource.getProject();
39                     IProject container= root.getJavaProject().getProject();
40                     return !container.equals(jarProject);
41                 }
42             }
43         }
44         return true;
45     }
46 }
47

```