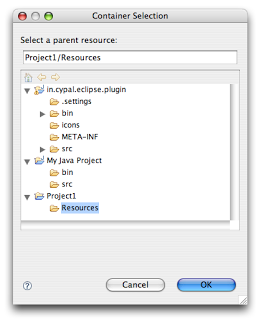
[**Selection Dialogs in Eclipse**](http://blog.eclipse-tips.com/2008/07/selection-dialogs-in-eclipse.html)

If you are an Eclipse Plug-in developer, you must have used the MessageDialog. There are many other Dialogs provided by Eclipse Platform are reusable and part of the API. I'll try to explain the various selection dialogs that I know of. In case I've missed any, add a comment and do let me know. Will add it to the list

***ContainerSelectionDialog***:

     You can use this dialog when you want to select a container (Project/Folder) for your new resource.

[](http://1.bp.blogspot.com/_hsp14iFkRLs/SH8a6f7eQsI/AAAAAAAACg8/rK4dVs8lWbI/s1600-h/Container_Selection.png)

[JavaDoc](http://help.eclipse.org/ganymede/topic/org.eclipse.platform.doc.isv/reference/api/org/eclipse/ui/dialogs/ContainerSelectionDialog.html)

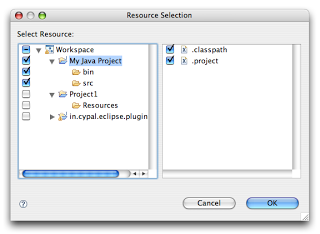
Example:

ContainerSelectionDialog dialog = new ContainerSelectionDialog(window.getShell(), null, true, "Select a parent:");  
  
dialog.setTitle("Container Selection");  
  
dialog.open();

You can even restrict the resource to be within a project/folder by passing the respective object as the second parameter for the constructor.

***ResourceSelectionDialog:***

The ContainerSelectionDialog allowed you to select only one resource that too it should be a container. If you want to select multiple resources including files, this is the dialog you should be using.

[](http://3.bp.blogspot.com/_hsp14iFkRLs/SH8hlCn98rI/AAAAAAAAChM/WlNOVqX5ZdA/s1600-h/Resource_Selection_Dialog.png)

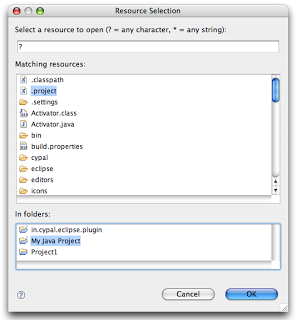
[JavaDoc](http://help.eclipse.org/ganymede/topic/org.eclipse.platform.doc.isv/reference/api/org/eclipse/ui/dialogs/ResourceSelectionDialog.html)

Example:

ResourceSelectionDialog dialog = new ResourceSelectionDialog(window.getShell(), ResourcesPlugin.getWorkspace().getRoot(), "Select Resource:");  
  
dialog.setTitle("Resource Selection");  
  
dialog.open();

***ResourceListSelectionDialog:***

The above dialog is good when you want to present the entire set of resources under a parent and allow the user to select multiple resources. But if you have a set of resources and want the user to select only one from that, then probably this dialog is the one you should be using.

[](http://3.bp.blogspot.com/_hsp14iFkRLs/SH8kcZLgNnI/AAAAAAAAChU/_6hEDSVsiBA/s1600-h/Resource_List_Selection_Dialog.png)

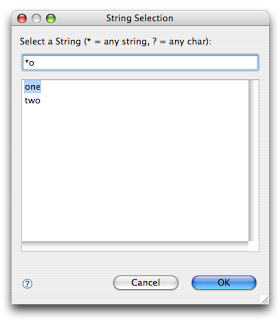
[JavaDoc](http://help.eclipse.org/ganymede/topic/org.eclipse.platform.doc.isv/reference/api/org/eclipse/ui/dialogs/ResourceListSelectionDialog.html).

Example:

ResourceListSelectionDialog dialog = new ResourceListSelectionDialog(window.getShell(), resourcesArray);  
dialog.setTitle("Resource Selection");  
dialog.open();

***ElementListSelectionDialog:***

Alright. So far we have been looking at selecting something from the workspace. But what if I have some elements on my own and I want to select from that? The first dialog you would be using is ElementListSelectionDialog. The user can select an element from the set. You have to pass the elements as an array and supply a  label provider to render the element. The user can filter using wildcards as well:

[](http://1.bp.blogspot.com/_hsp14iFkRLs/SH8n2FZLVDI/AAAAAAAAChc/e6JexFgLQDk/s1600-h/Element_List_Selection_Dialog.png)

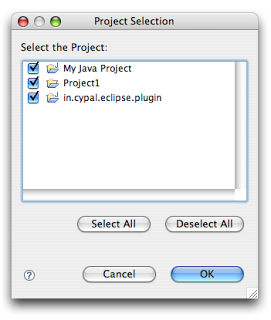
[JavaDoc](http://help.eclipse.org/ganymede/topic/org.eclipse.platform.doc.isv/reference/api/org/eclipse/ui/dialogs/ElementListSelectionDialog.html)

Example:

ElementListSelectionDialog dialog = new ElementListSelectionDialog(window.getShell(), new LabelProvider());  
  
dialog.setTitle("String Selection");  
  
dialog.setMessage("Select a String (\* = any string, ? = any char):");  
  
dialog.setElements(new Object[] { "one", "two", "three" });  
  
dialog.open();

***ListSelectionDialog:***

That holds good for selecting a single element. What if you want to select multiple elements from the given set? ListSelectionDialog is the answer to it. It is basically a single column TableViewer with SWT.CHECK style applied. You have to supply your own ContentProvider and LabelProvider for the TableViewer. It also has Select All & Deselect All buttons. (In the example I've used workspace and the associated label & content providers, but this dialog is not tied to IResource in anyway

[](http://3.bp.blogspot.com/_hsp14iFkRLs/SH8ebY8Y3yI/AAAAAAAAChE/aGI0GriTRO0/s1600-h/List_Selection_Dialog.png)

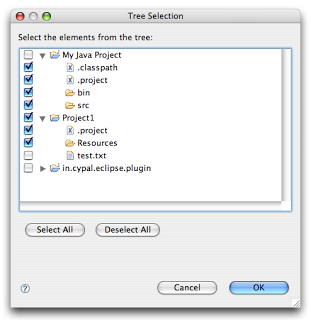
[JavaDoc](http://help.eclipse.org/ganymede/topic/org.eclipse.platform.doc.isv/reference/api/org/eclipse/ui/dialogs/ListSelectionDialog.html)

Example:

ListSelectionDialog dlg = new ListSelectionDialog(window.getShell(), ResourcesPlugin.getWorkspace().getRoot(), new BaseWorkbenchContentProvider(), new WorkbenchLabelProvider(), "Select the Project:");  
dlg.setTitle("Project Selection");  
dlg.open();

***CheckedTreeSelectionDialog:***

If you have your items in a tree structure and want to select few elements from them, then CheckedTreeSelectionDialog is your choice. You have to bring your own content & label provider and the input. Again, due to sheer laziness, I'm using the workspace as the input and associated content & label providers, but remember that this dialog can work well with your own data as well.

[](http://2.bp.blogspot.com/_hsp14iFkRLs/SH8rZnM2jKI/AAAAAAAAChk/Fyuvq7DMqqk/s1600-h/Checked_Tree_Selection_Dialog.png)

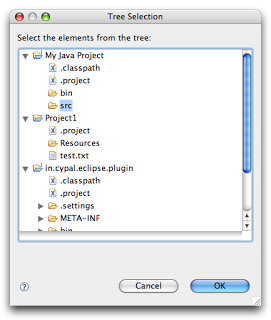
[JavaDoc](http://help.eclipse.org/ganymede/topic/org.eclipse.platform.doc.isv/reference/api/org/eclipse/ui/dialogs/CheckedTreeSelectionDialog.html)

Example:

CheckedTreeSelectionDialog dialog = new CheckedTreeSelectionDialog(window.getShell(), new WorkbenchLabelProvider(), new BaseWorkbenchContentProvider());  
  
dialog.setTitle("Tree Selection");  
  
dialog.setMessage("Select the elements from the tree:");  
  
dialog.setInput(ResourcesPlugin.getWorkspace().getRoot());  
  
dialog.open();

***ElementTreeSelectionDialog:***

This is the same as the above except that it will allow you to select a single element in the whole tree rather than multiple elements. Again, not tied to workspace & resources.

[](http://1.bp.blogspot.com/_hsp14iFkRLs/SH8sNwHA34I/AAAAAAAAChs/rLCynZAlgdc/s1600-h/Element_Tree_Selection_Dialog.png)

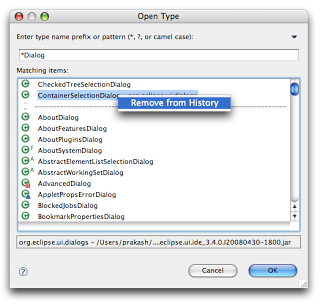
[JavaDoc](http://help.eclipse.org/ganymede/topic/org.eclipse.platform.doc.isv/reference/api/org/eclipse/ui/dialogs/ElementTreeSelectionDialog.html)

Example:

ElementTreeSelectionDialog dialog = new ElementTreeSelectionDialog(window.getShell(), new WorkbenchLabelProvider(), new BaseWorkbenchContentProvider());  
dialog.setTitle("Tree Selection");  
dialog.setMessage("Select the elements from the tree:");  
dialog.setInput(ResourcesPlugin.getWorkspace().getRoot());  
dialog.open();

***FilteredItemsSelectionDialog:***

Have you used the Open Type (Ctrl + Shift + T) or Open Resource (Ctrl+Shift+R) dialog? Its similar to the ElementListSelectionDialog, but it has more features. It can select multiple items, display a detail pane about the item that is currently selected, it can even remember your previous selections; store them in history and present them before the other choices.

[](http://3.bp.blogspot.com/_hsp14iFkRLs/SH8vC1N263I/AAAAAAAACh0/hKCWlyWFv4Q/s1600-h/Filtered_Items_Selection_Dialog.png)

FilteredItemsSelectionDialog is the key. Its an abstract class and you have to extend it and create a concrete implementation to use it. I'll resever the details of extending it for a separate tip, till then you can go thru the [JavaDoc](http://help.eclipse.org/ganymede/topic/org.eclipse.platform.doc.isv/reference/api/org/eclipse/ui/dialogs/FilteredItemsSelectionDialog.html).  
  
  
  
[Update]: This has been added to the [Eclipse User Interface Guidelines](http://wiki.eclipse.org/User_Interface_Guidelines#Selection_Dialogs)