

okButton

Accessing Workspace selection

Managing changes ok:

revertButton
window

selectionCount
selectionPathsInto:separator:

revert:
textDidChange:
touch:

ok:sender

Implement in your subclass to commit the changes that the user has made to the selected item. The Inspector panel sends an ok: message when the user clicks it.

This method is optional, but if you implement it, you must send the same message to super as part of your implementation:

revert:, touch:

okButton

Returns the id of the Inspector's OK button. This can be useful if you want to alter its title, for example.

revertButton:

revert:sender

Implement in your subclass to load data into the inspector's display. The Workspace Manager sends an inspector object whenever the inspector's display might need to be updated for example, when the inspector is opened or when the selection changes in the File Viewer.

Your subclass must implement this method, and it must send the same message to super as part of its implementation:

ok:, touch:

Returns the number of items selected in the File Viewer. You can use this information to determine how many inspectors should be displayed. For example, most inspectors can give information on only one file at a time. If your inspector has revert methods, they would have this test:

`selectionPathsInto:separator:`

`selectionPathsInto:(char *)pathString separator:(char)character`

Returns the paths of the files selected in the File Viewer. The paths are placed in the string `pathString` separated from the previous one by `character`. For example, if `character` is ``:'`, `pathString` could contain `test2:/me/test3°`.

If your inspector acts on only one file at a time (see `selectionCount`), the file's path can be identified by the last character in the string.

`selectionCount`

`textDidChange:sender`

Sends the `WMInspector` a `touch:` message on behalf of some `Text` object in the Inspector panel.

By making your inspector object the delegate of any `Text` object in your inspector's display, the Inspector is updated appropriately as the user alters the panel's contents.

`touch:`

`touch:sender`

Changes the image in the Inspector panel's close box to a broken `°X°` to indicate that the contents are not valid. This enables the OK and Revert buttons.

`textDidChange:`

`window`

Returns the id of the window that contains the user interface for the inspector.