

TToolsPalette:

This is a small palette providing several basic, useful objects.

The Timer object is fairly straightforward; it emits messages on a timed entry and holds some state so that other items may obtain values based on the timed entry.

The SwitchView object is more complex. It is mostly an InterfaceBuilder toy. To use it, hook up a number of views to be swapped into the SwitchView, by control-dragging connections from the SwitchView. You'll find that the custom connection inspector will allow you to add as many as you like. When you've set up all the content views, then connect an ordinary control - a matrix of buttoncells, popuplist, button, or even TextField - to the view, so that the action taken is "takeViewNumberFrom". Now, using the Attributes inspector of the SwitchView instance, make sure the "MessageToSend" matrix is set to "Automatic". Under Test Interface mode, you should see logical results - pressing the nth button in your matrix control should result in the nth view swapped in, or in the case of a Button, the tag of the button will be used to determine what view in the list should be shown.

The Ranker class allows one to control-drag cells around in a matrix. Its main purpose in life, however, is to show the advanced palette builder how to properly subclass and use Matrix within IB.

The SortedList class is a simple subclass of List that adds items in sorted order. Of interest to advanced users of IB is the ListEditor, which allows one to set up arbitrary lists of objects from within IB, which of course are archived properly when your nib file is saved, and unarchived correctly when the nib file is loaded. This class is used indirectly in TTools to keep the items in the class browser sorted.

The ClassHierarchy category of Object implements a few methods that allow you to get a list of the classes that directly inherit from a given class, which was useful in implementing the class browser used in the palette. Like SortedList, this class is not directly available in IB, but is included in the library and header files, as it is of fairly general use.