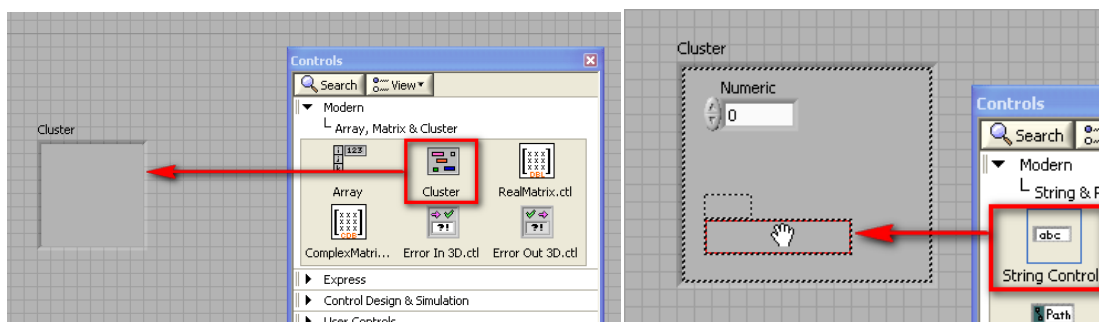


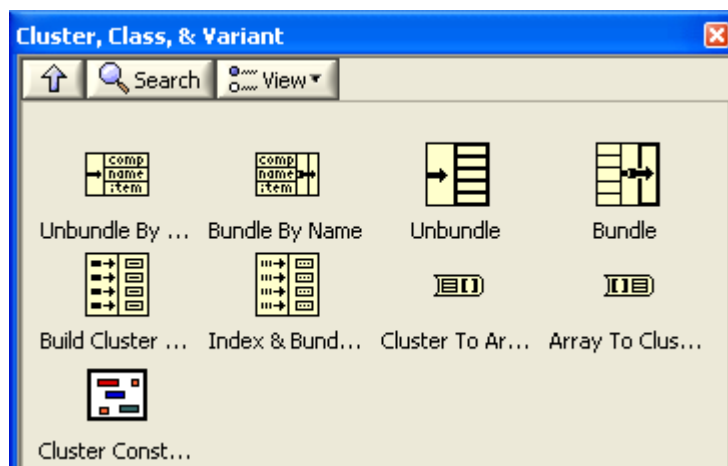
## LabVIEW Clusters

Clusters group data elements of mixed types, such as a bundle of wires, as in a telephone cable, where each wire in the cable represents a different element of the cluster.

A cluster is similar to a record or a struct in text-based programming languages. Bundling several data elements into clusters eliminates wire clutter on the block diagram and reduces the number of connector pane terminals that subVIs need. Like an array, a cluster is either a control or an indicator. A cluster cannot contain a mixture of controls and indicators.



**Cluster functions:** In the Cluster, Class & Variant subpalette on the Block Diagram we have the following Cluster functions we may use to manipulate and get data in or out of a cluster. In this example we will create clusters and use these functions.

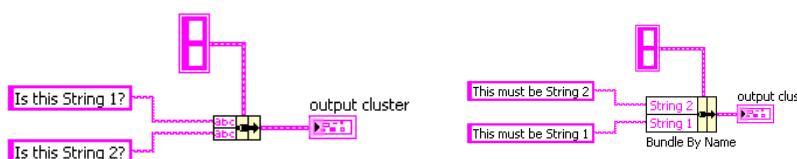
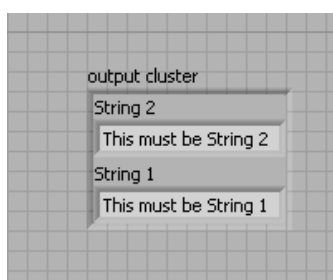
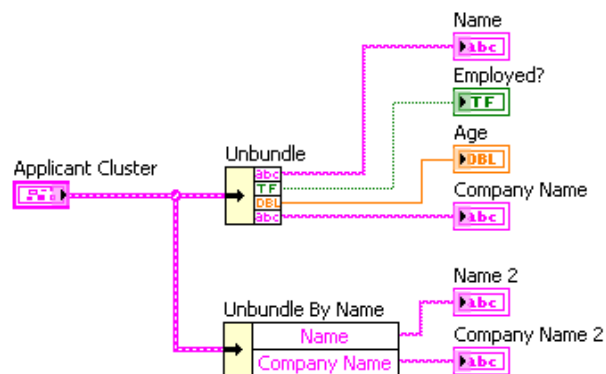


**Task: Unbundle/Unbundle By Name.** Use the **Unbundle** functions to disassemble a cluster into its individual elements. Use the **Unbundle by Name** function to return specific cluster elements you specify by name. You can also resize these functions for multiple elements using the mouse. Create the following code:

**Task: Bundle/Bundle**

**By Name.** Use the **Bundle** function to assemble a cluster from individual elements. To wire

elements into the Bundle function, use your mouse to resize the function. Create the following code:



**Task: Cluster Order.** Cluster elements have a logical order unrelated to their position in the shell. The first object you place in the cluster is element 0, the second is element 1, and so on. If you delete an element, the order adjusts automatically. The cluster order determines the order in which the elements appear as terminals on the Bundle and Unbundle functions on the block diagram. You can view and modify the cluster order by right-clicking the cluster border and selecting Reorder Controls in Cluster from the shortcut menu.

