

## ***Drag and Drop Sample Help***

Sample Description: [Drag and Drop](#)

### **Points of Interest**

[Drag and Drop Files From the FileManager](#)

[Restricting Input in a TextBox](#)

[Resizing the Form](#)

[Viewing Bitmaps](#)

### **Control**

Image

ListBox

For Help on Help, Press F1

**Drag and Drop**

The Drag and Drop sample contains two primary objects; A hypercontrol named PagingControl and the main drag and drop form itself named PagingForm. The purpose of the application is to drag and drop several Windows bitmap files (.BMP) onto the Image viewer control, then use the paging buttons to view the various bitmaps. The PagingForm may be resized to various sizes to facilitate viewing. The image will always be fit into the viewing area.

**PagingControl**

The PagingControl object is a form containing four buttons: Cancel, Back, Next, and Go To. These buttons are used to page back and forth between various images. To make the hypercontrol reusable, the click event of each of the buttons sends a corresponding "event." These events were defined as: ValidateCancel, ValidateBack, ValidateNext, ValidateGoTo. In addition, a Resize event is defined to automatically resize the width of the buttons as well as locate them. The height of the hypercontrol is fixed.

**PagingForm**

The PagingForm is a form that contains an embedded copy of the PagingControl hyper-control. In addition, an Image control is used to present the bitmap files that are dropped onto the form. A TextBox is available for the user to enter a specific bitmap to view with the GoTo button. A hidden ListBox is used to capture the names of the files that are dropped on the form.

### **Drag and Drop Files From the FileManager**

The form contains a DragAndDrop event handler, which is used to determine if the file being dropped on the form is a .BMP file. If it is, the filename is then added to the hidden ListBox and displayed in the Image control by setting the FileName property of the BitmapSource object, which is the Picture source of the ImgBitmapDisplay object.

### **Restricting Input in a TextBox**

The TextBox TxtPageNumber contains a KeyPress event handler that filters keyboard input. Numbers 0-9 are permitted as well as a Backspace character. All other keyboard characters are blocked by having the KeyAscii value set to 0.

**Resizing the Form**

The PagingForm contains a Resize event handler, which automatically locates the hypercontrol PagingControl in the lower right hand corner of the form. In addition, the width and height of the Image view and the ReadMe button are also adjusted automatically.

### **Viewing Bitmaps**

Event handlers for the various hypercontrol events have been created to handle the paging functions. An integer property "CurrentPage" has been added to the PagingForm to indicate the number of the current page being viewed. These various paging methods check to make sure there are valid pages loaded. In the case of the GoTo event handler, the method must also make sure the page number the user entered is valid. Once all validations are completed, the CurrentPage property is changed to indicate the page to turn to, then the UpdateDisplay method is called. The UpdateDisplay method extracts the corresponding filename from the Hidden ListBox and sets the BitmapSource object's FileName property.

