

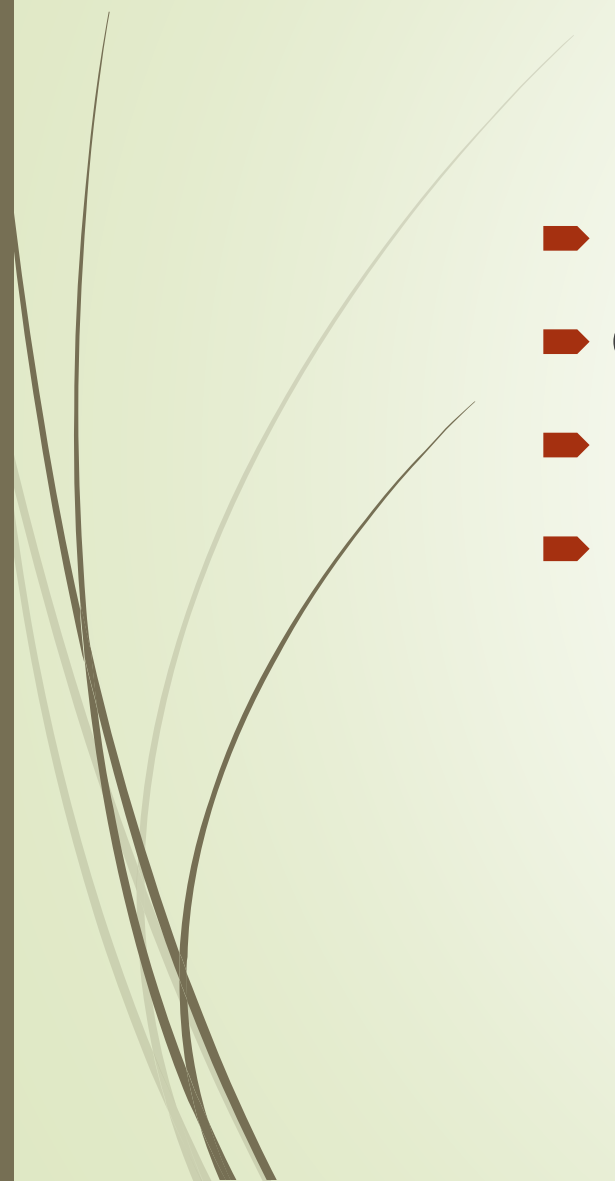


Visual Basic.NET

More Windows Control



ROAD MAP

- Introduction
 - Common Dialog Controls
 - RichTextbox Controls
 - RTF Language
- 



INTRODUCTION

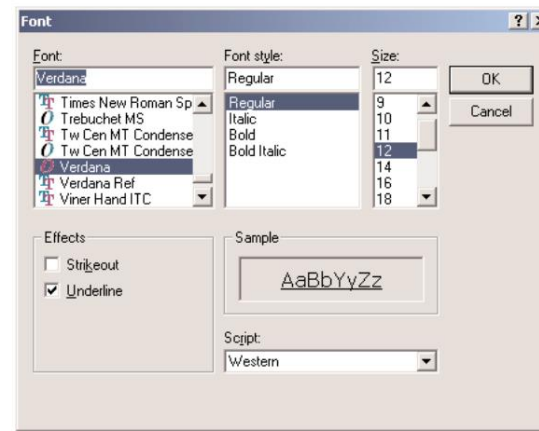
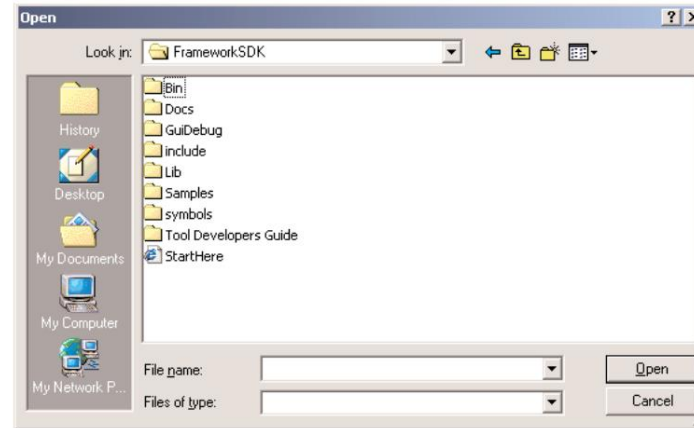


- a set of controls for displaying common dialogs
 - such as Open or Color
- common dialog controls are an essential part of a Windows application
 - they enable you to design user interfaces with the look and feel of a Windows application
- Explore the RichTextBox control

The Common Dialog Controls

- A rather tedious, but quite common, task in nearly every application is to prompt the user for filenames, font names and sizes, or colors to be used by the application

FIGURE 7.1
The Open and Font
common dialog
boxes





The Common Dialog Controls

- Invisible at runtime
- Not placed on your forms
- **OpenFileDialog**
 - Lets users select a file to open.
- **SaveFileDialog**
 - Lets users select or specify a filename in which the current document will be saved
- **ColorDialog**
 - Lets users select a color from a list of predefined colors, or specify custom color
- **FontDialog**
 - Lets users select a typeface and style to be applied to the current text selection
- **PrintDialog**
 - Lets users select and set up a printer



Using Common Dialog Controls

- Call the control's ShowDialog method

```
If OpenFileDialog1.ShowDialog = DialogResult.OK Then  
    fileName = OpenFileDialog1.FileName  
End If
```


The Color Dialog Box

- One of the simplest dialog boxes
- Has a single property, Color
 - which returns the color selected by the user or sets the initially selected color when the user opens the dialog box

```
Private Sub Button1_Click(ByVal sender As System.Object, _  
                          ByVal e As System.EventArgs) Handles Button1.Click  
    ColorDialog1.Color = Me.BackColor  
  
    ColorDialog1.AllowFullOpen = True  
    If ColorDialog1.ShowDialog = DialogResult.OK Then  
        Me.BackColor = ColorDialog1.Color  
    End If  
End Sub
```



Properties



- AllowFullOpen
 - Set True
 - users to be able to open up the dialog box and define their own custom colors
- AnyColor
 - A Boolean value
 - whether the dialog box displays all available colors in the set of basic colors
- Color
 - a Color value
 - Set to any valid color



Properties

- CustomColors

- Indicate the set of custom colors that will be shown in the common dialog
- Display 16 additional custom colors

```
Dim colors() As Integer = {222663, 35453, 7888}  
ColorDialog1.CustomColors = colors
```

- SolidColorOnly

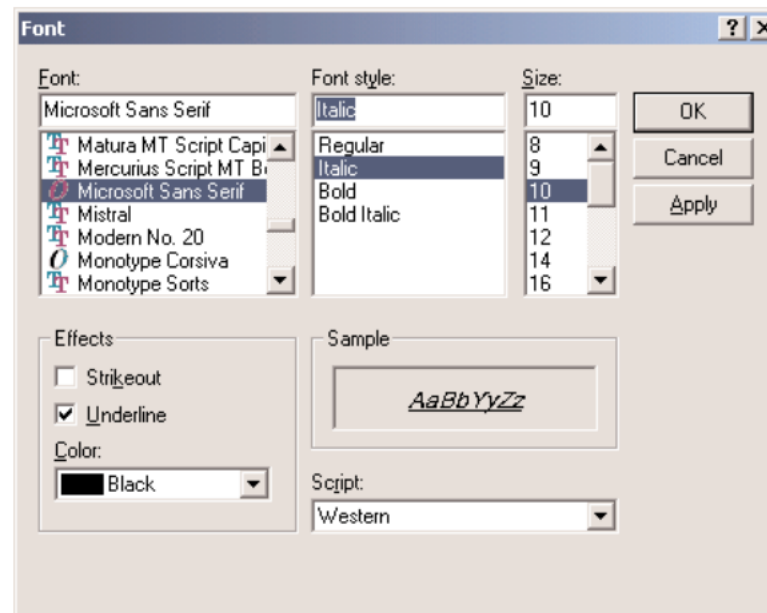
- Indicates whether the dialog box will restrict users to selecting solid colors only
- Only 256 colors

The Font Dialog Box

- Let the user review and select a font and its size and style

FIGURE 7.3

The Font common dialog box





The Font Dialog Box's Properties

- AllowScriptChange
 - a Boolean value
 - whether the Script combo box will be displayed on the Font common dialog
- AllowSimulations
 - a Boolean value
 - whether the dialog box allows the display and selection of simulated fonts
- AllowVectorFonts
 - a Boolean value
 - whether the dialog box allows the display and selection of vector fonts
- AllowVerticalFonts
 - a Boolean value
 - whether the dialog box allows the display and selection of both vertical and horizontal fonts
 - Default = False – horizontal fonts

The Font Dialog Box's Properties

- Color
 - Set or return the selected font color.
- FixedPitchOnly
 - a Boolean value
 - whether the dialog box allows only the selection of fixed-pitch fonts
 - Default = false – all fonts displayed on the common dialog
- Font
 - a Font object

```
FontDialog1.Font = TextBox1.Font  
If FontDialog1.ShowDialog = DialogResult.OK Then  
    TextBox1.Font = FontDialog1.Font  
End If
```



The Font Dialog Box's Properties

- FontMustExist
 - a Boolean value
 - whether the dialog box forces the selection of an existing font
 - Default = True
- MaxSize, MinSize
 - integers that determine the minimum and maximum point size the user can select
- ScriptsOnly
 - whether the dialog box allows selection of fonts for Symbol character sets
 - Default = True
- ShowApply
 - a Boolean value
 - whether the dialog box provides an Apply button
 - Default = False

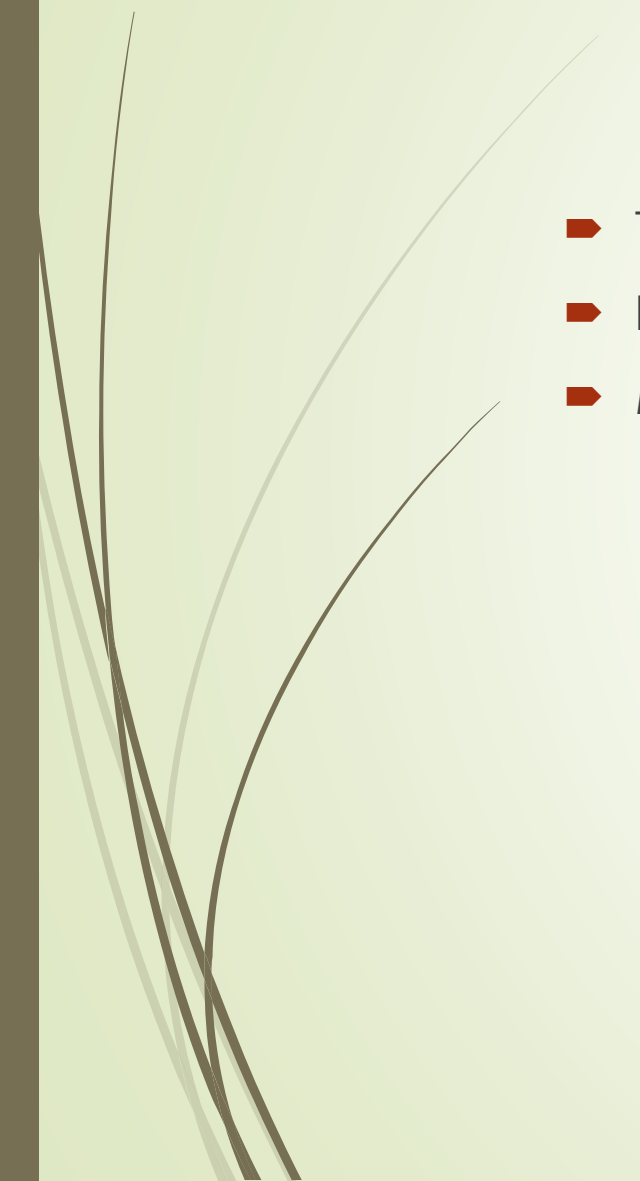


The Font Dialog Box's Properties

- ShowColor
 - a Boolean value
 - whether the dialog box allows the user to select a color for the font
- ShowEffects
 - a Boolean value
 - whether the dialog box contains controls to allow the user to specify special text effects
 - strikethrough and underline



The Open and Save As Dialog Boxes

- Two most widely used common dialog boxes
 - Implemented by the OpenFileDialog and SaveFileDialog controls
 - Most of their properties are common
- 



Properties



- **AddExtension**
 - a Boolean value
 - whether the dialog box automatically adds an extension to a filename, if the user omits it
- **CheckFileExists**
 - a Boolean value
 - whether the dialog box displays a warning if the user enters the name of a file that does not exist
- **CheckPathExists**
 - a Boolean value
 - whether the dialog box displays a warning if the user specifies a path that does not exist, as part of the user-supplied filename



Properties



- DefaultExtension
 - Set the default extension of the dialog box
 - TXT or DOC
- DereferenceLinks
 - whether the dialog box returns the location of the file referenced by the shortcut or the location of the shortcut itself
 - C:\WINDOWS\SYSTEM32\Inkstub.exe
- FileName
 - path of the file selected by the user on the control
 - The user can click OK/Cancel to select this file



Properties

➤ Filter

- specify the type(s) of files displayed on the dialog box
- "Text files | *.txt"
- multiple extensions

➤ FilterIndex

- specify more than one filter for the Open dialog box, the filter specified first in the Filter property becomes the default

➤ InitialDirectory

- set the initial directory (folder) in which files are displayed the first time the Open and Save dialog boxes are opened



Properties

- RestoreDirectory
 - a Boolean value
 - whether the dialog box restores the current directory before closing
 - Default = False
- ValidateNames
 - a Boolean value
 - whether the dialog box accepts only valid Win32 filenames
 - Default = True

TIP *The following four properties—FileNames, MultiSelect, ReadOnlyChecked, and ShowReadOnly—are properties of the OpenFileDialog control only.*



Properties

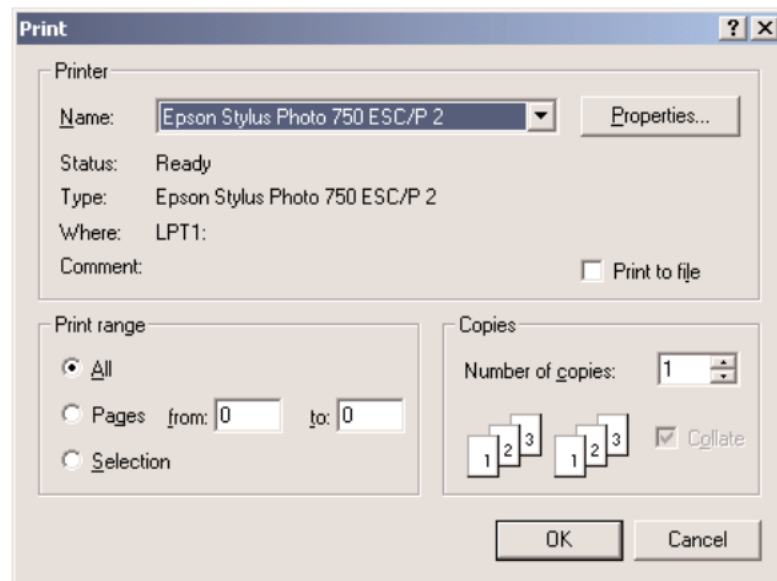
- FileNames
 - a collection
 - iterate through the filenames with an enumerator
- MultiSelect
 - a Boolean value
 - whether the user can select multiple files on the dialog box
 - Default = False
- ReadOnlyChecked
 - a Boolean value
 - whether the Read-Only check box is initially selected when the dialog box first pops up
- ShowReadOnly
 - a Boolean value
 - whether the Read-Only check box is available

The Print Dialog Box

- enables users to select a printer
- set certain properties of the printout
 - number of copies and pages to be printed
- set up a specific printer

FIGURE 7.8

The Print common dialog box





The Print Dialog Box

- **AllowPrintToFile**

- a Boolean value
- whether the user will be given the option to print to a file

- **AllowSelection**

- a Boolean value
- whether the user is allowed to print the current selection of the document

- **AllowSomePages**

- a Boolean value
- whether the Pages option on the dialog will be enabled



The RichTextBox Control

- The core of a full-blown word processor
- Provide all the functionality of a TextBox control
- Give you the capability to mix different fonts, sizes, and attributes
- Gives you precise control over the margins of the text

- No need to supply the formatting codes
- Provide simple properties that turn the selected text into bold, change the alignment of the current paragraph

NOTE *The WordPad application that comes with Windows is based on the RichTextBox control. You can easily duplicate every bit of WordPad's functionality with the RichTextBox control, as you will see later on in the section "VB.NET at Work: The RTFPad Project."*

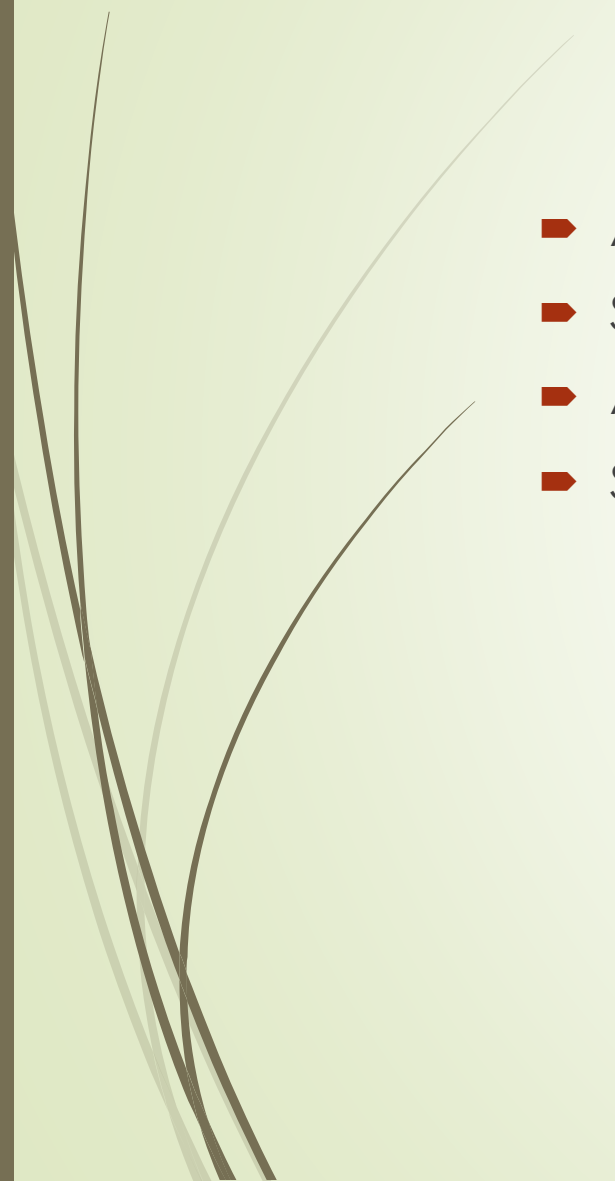


RTF Property

- RTF → *Rich Text Format*
 - a standard for storing formatting information along with the text
- The text displayed on the control
- Return the text *along with* any formatting information
- Specify the text's formatting, including paragraph indentation, font, and font size or style



The RTF Language

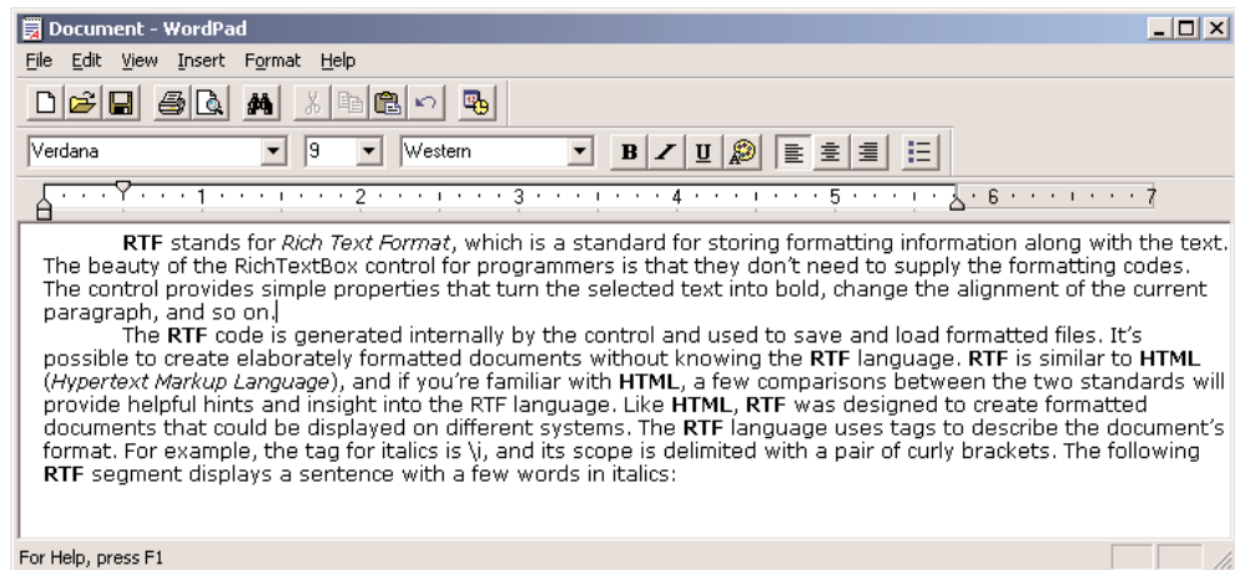
- A basic knowledge of the RTF format
 - Simple Commands
 - ASCII strings
 - Similar to HTML
- 

The RTF Language

- Open the WordPad application
- choose Start ➤ Programs ➤ Accessories ➤ WordPad
- enter a few lines of text
- File ➤ Save As
- Document.rtf

FIGURE 7.10

The formatting applied to the text using WordPad's commands is stored along with the text in RTF format.





The RTF Language

- All formatting tags are prefixed with the backslash (\) symbol
- Paragraphs are marked with the \par tag
- The entire document is enclosed in a pair of curly brackets
- The \li and \ri tags followed by a numeric value specify the amount of the left and right indentation



The RTF Language



RTF vs HTML

`{{\b RTF}` (which stands for Rich Text Format) is a `{\i` document formatting language`}` that uses simple commands to specify the formatting of the document.

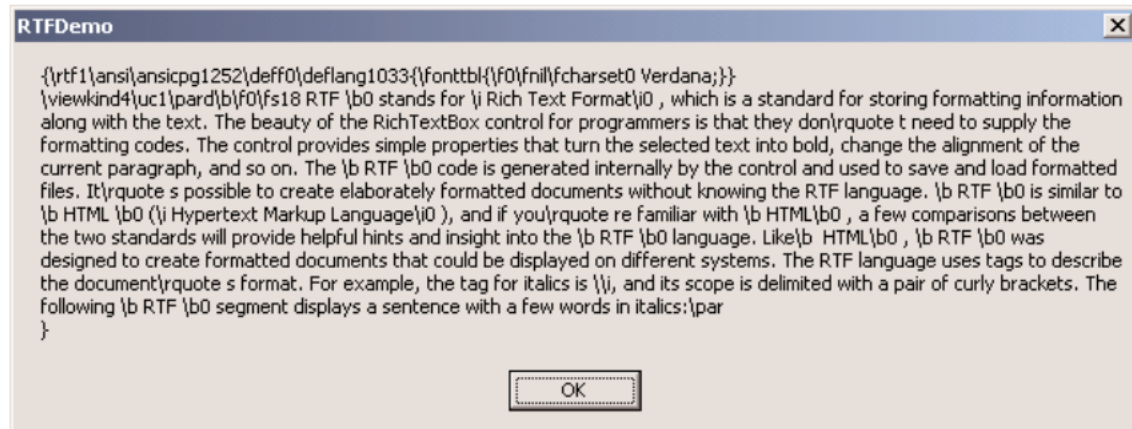
`RTF` (which stands for Rich Text Format) is a `<i>`document formatting language`</i>` that uses simple commands to specify the formatting of the document.

The RTF Code

- Simply manipulating a few properties

FIGURE 7.12

The RTF code for the formatted document shown in Figure 7.11



The RichTextBox's Properties

TABLE 7.1: RICHTEXTBOX PROPERTIES FOR MANIPULATING SELECTED TEXT

PROPERTY	WHAT IT MANIPULATES
SelectedText	The selected text
SelectedRTF	The RTF code of the selected text
SelectionStart	The position of the selected text's first character
SelectionLength	The length of the selected text
SelectionFont	The font of the selected text
SelectionColor	The color of the selected text
SelectionIndent, SelectionRightIndent, SelectionHangingIndent	The indentation of the selected text
RightMargin	The distance of the text's right margin from the left edge of the control, which is in effect the length of each text line
SelectionBullet	Whether the selected text is bulleted
BulletIndent	The amount of bullet indent for the selected text



Methods

- **SaveFile** saves the contents of the control to a disk file.
- **LoadFile** loads the control from a disk file

SaveFile

- Syntax `RichTextBox1.SaveFile(path, filetype)`
 - *path* is the path of the file in which the current document will be saved
 - the SaveFile method saves the document in RTF format and uses the RTF extension

TABLE 7.2: THE RICHTEXTBOXSTREAMTYPE ENUMERATION

FORMAT	EFFECT
PlainText	Stores the text on the control without any formatting
RichNoOLEobjs	Stores the text without any formatting and ignores any embedded OLE objects
RichText	Stores the formatted text
TextTextOLEobjs	Stores the text along with the embedded OLE objects
UnicodePlainText	Stores the text in Unicode format

LoadFile

- Syntax `RichTextBox1.LoadFile(path, filetype)`
 - `filetype` argument is optional and can have one of the values of the `RichTextBoxStreamType` enumeration

NOTE *You can't assign formatted text to the control at design time. The `Text` property is available at design time, but the text is rendered in the same format. The `RTF` property isn't available at design time. To display initially some formatted text on the control, you must either load it from a file with the `LoadFile` method, or assign the equivalent `RTF` code to the `RTF` property at runtime, usually from within the form's `Load` event.*



Select, SelectAll

- Select Method

- selects a section of the text on the control
- accepts two arguments

```
RichTextBox1.Select(start, length)
```

- SelectAll Method

- accepts no arguments and selects all the text on the control

Exercise

- (1) What are four options for each image file type?

```
OpenFileDialog1.Filter = "Bitmaps|*.BMP|GIF Images|*.GIF|JPEG" & _  
                        "Images|*.JPG|All Images|*.BMP;*.GIF;*.JPG"
```

- (2) What are the meaning of the SelectionStart in the RichTextBox's properties?