Text and Fonts in Windows

**DrawString()**

- Graphics class member method to display a string on a graphics object
- Six overloaded forms
- All have Font as an argument
Fonts

• FONT: Typeface, style, size, attributes of characters in a character set
  – Provide control over the visual appearance of text

Categories of Fonts

• Windows System Fonts
  – Always available

• Logical Fonts – Defined in separate resource files
  – Stroke fonts
    • Consist of line/curve segments – so continuously scalable
    • Slow to draw
    • Legibility not good
  – Raster fonts--Bitmaps
    • Scaling by non-integer scaling factor difficult
    • Fast to display
    • Legibility very good
  – TrueType/OpenType (Adobe) fonts--Rasterized stroke fonts
    • Stored as strokes with hints to convert to bitmap
    • Continuously scalable
    • Fast to display
    • Legibility very good
    • Combine best of both stroke and raster fonts

• Device fonts
  – Native to output device (e.g., built-in printer fonts)
Windows System Fonts

Font = ANSI_FIXED_FONT
Font = ANSI_VAR_FONT
Font = DEVICE_DEFAULT_FONT
Font = OEM_FIXED_FONT
Font = SYSTEM_FONT
Font = SYSTEM_FIXED_FONT

Windows Stock Fonts

Some Stroke Fonts

Modern AоБаСеДеЕе
Roman АаБбCcДеЕе
Script AаБбCcДеЕе

Windows Stroke Fonts
### Some Bitmapped Fonts

<table>
<thead>
<tr>
<th>Font Style</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>Courier</td>
<td>АаБбCcDдEе</td>
</tr>
<tr>
<td>MS Serif</td>
<td>АаБбCcDдEе</td>
</tr>
<tr>
<td>MS Sans Serif</td>
<td>АаБбCcDдEе</td>
</tr>
<tr>
<td>Symbol</td>
<td>АαВβΧχΔδΕє</td>
</tr>
</tbody>
</table>

*Windows Raster Fonts*

### Some TrueType Fonts

<table>
<thead>
<tr>
<th>Font Style</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>Courier New</td>
<td>АаБбCcDдEе</td>
</tr>
<tr>
<td>Courier New Bold</td>
<td>АаБбCcDдEе</td>
</tr>
<tr>
<td>Courier New Italic</td>
<td>АаБбCcDдEе</td>
</tr>
<tr>
<td>Courier New Bold Italic</td>
<td>АаБбCcDдEе</td>
</tr>
<tr>
<td>Times New Roman</td>
<td>АаБбCcDдEе</td>
</tr>
<tr>
<td>Times New Roman Bold</td>
<td>АаБбCcDдEе</td>
</tr>
<tr>
<td>Times New Roman Italic</td>
<td>АаБбCcDдEе</td>
</tr>
<tr>
<td>Times New Roman Bold Italic</td>
<td>АаБбCcDдEе</td>
</tr>
<tr>
<td>Arial</td>
<td>АаБбCcDдEе</td>
</tr>
<tr>
<td>Arial Bold</td>
<td>АаБбCcDдEе</td>
</tr>
<tr>
<td>Arial Italic</td>
<td>АаБбCcDдEе</td>
</tr>
<tr>
<td>Arial Bold Italic</td>
<td>АаБбCcDдEе</td>
</tr>
<tr>
<td>Symbol</td>
<td>АαВβΧχΔδΕє</td>
</tr>
</tbody>
</table>

*Windows TrueType Fonts*
Changing Fonts

- Two important classes in System.Drawing:
  - FontFamily
    - Specified by a string such as “Times New Roman”
  - Font
    - A combination of a FontFamily, attributes (e.g., Bold, Italic, etc.), and a point size

- Font Class
  - Categories of Font constructors
    - Based on an existing Font object
    - Based on a Font family

Simplest Font Constructor

- Creates a new font based on an existing font
- New font is the same except for the font style
  - Font(Font font, FontStyle fs);
    - FontStyle Enumeration
      - Regular 0
      - Bold 1
      - Italic 2
      - Underline 4
      - Strikeout 8
    » Can use bitwise OR operator to combine Font styles
- Examples:
  Font f = this.Font;  // Get this form’s Font property
  Font fItalic = new Font(f, FontStyle.Italic);
  » Can now draw with this new fItalic font
**MeasureString()**

- Member of Graphics class
- Returns width and height of imaginary rectangle bounding a string
  - Several overloaded forms
  - Simplest:
    - `MeasureString(string str, Font font);`
  - Returns a SizeF structure
    - Members are width and height of bounding rectangle in pixels

---

**Font-Bold-Italic Example Program**

- Outputs text with one word bolded and another italicized
  - Creates new fonts from form’s existing font
- Uses MeasureString() to position each new word on the window’s client area
Creating Fonts by Font Family Name

- Specify a font by giving its font family name, the point size, and optionally a style

- Font Constructors:
  - Font(string strFamily, float fSizeInPoints)
    - There are about 72 “points” per inch
    - Anything smaller than 8-point is hard to read
  - Font(string strFamily, float fSizeInPoints, FontStyle fs)
    - strFamily must represent a TrueType/OpenType font that is on the system
      Font myFont = new Font(“Times New Roman”, 12);

- The Font property of a window form can be set in the form’s constructor, e.g.:
  this.Font = new Font(“Arial”, 24, FontStyle.Bold);

- Font-Name & Font-Sizes example programs
  - Note use of foreach( ) C# construct
  - And font.GetHeight(g): more general than MeasureString()
    - Could be used for a printer or screen graphics object

Font Class Properties

- All are read-only
  - string Name Font family name
  - FontFamily FontFamily Font family class
  - FontStyle Style From constructor
  - bool Bold True if boldface
  - bool Italic True if italic
  - bool Underline True if underlined
  - bool Strikeout True if strikeout
  - float Size From constructor
  - int Height Line spacing for video display
  - Others

- Example program: Font-Properties
Getting a Font from a GDI Logical Font

- Font.FromLogFont(GDI LOGFONT object)
  - Returns a new Font object
  - LOGFONT
  - typedef struct tagLOGFONT { LONG lfHeight; LONG lfWidth; LONG lfEscapement; LONG lfOrientation; LONG lfWeight; BYTE lfItalic; BYTE lfUnderline; BYTE lfStrikeOut; BYTE lfCharSet; BYTE lfOutPrecision; BYTE lfClipPrecision; BYTE lfQuality; BYTE lfPitchAndFamily; TCHAR lfFaceName[LF_FACESIZE]; } LOGFONT, *PLOGFONT;
  - Lots of possibilities

- Orientation--how much character is rotated
- Escapement--orientation of line between first & last character in a string

![Character Escapement & Orientation]