

Font và String

Font và String

- ▶ ☒ Các khái niệm cơ bản
- ▶ ☒ Font
 - ▶ - Các khái niệm về Font
 - ▶ - Lớp Font
 - ▶ - Lớp FontFamily
- ▶ ☒ String
 - ▶ - Một số thành phần liên quan đến chuỗi ký tự
 - ▶ - Lớp StringFormat
 - ▶ - Lớp TextRenderer

Các khái niệm cơ bản

- Bộ ký tự
- Trang mã
- Bộ ký tự Unicode

Bộ ký tự và Trang mã

Bộ ký tự (character set)

Bộ mã bàn phím cố định mà một hệ máy tính cụ thể đang sử dụng.

Trang mã (code page)

Trang mã chứa 1 bộ các ký tự ứng với hệ bộ ký tự nào đó của một hoặc nhiều ngôn ngữ.

Phần lớn các trang mã chứa 256 ký tự.

Character Set và Code Page

Character Set Name	Character Set ID	Code Page	Sử dụng
SHIFTJIS_CHARSET	128	932, Japanese	Japan
HANGUL_CHARSET	129	949, Korean	Korea
JOHAB_CHARSET	130	1361	
GB2312_CHARSET	134	936, Simplified Chinese	China, Singapore
CHINESEBIG5_CHARSET	136	950, Traditional Chinese	Taiwan, Hong Kong
GREEK_CHARSET	161	1253, Windows Greek	
TURKISH_CHARSET	162	1254, Windows Latin 5	Turkish
VIETNAMESE_CHARSET	163	1258, Windows Vietnamese	
HEBREW_CHARSET	177	1255, Windows Hebrew	
ARABIC_CHARSET	178	1256, Windows Arabic	
BALTIC_CHARSET	186	1257, Windows Baltic Rim	
RUSSIAN_CHARSET	204	1251, Windows Cyrillic	Slavic
THAI_CHARSET	222	874	
EASTEUROPE_CHARSET	238	1250, Windows Latin 2	Central Europe
ANSI_CHARSET	0	1252, Windows Latin 1	U.S., U.K., Canada, etc.
		1250, Windows Latin 2	Hungarian, Polish, etc.
		1256, Windows Arabic	Iraq, Egypt, Yemen, etc.

Ví dụ về Code Page

Windows Latin 1 code page (1252)

20		!	"	#	\$	%	_	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	□
80	€	□	,	f	„	...	†	‡	^	‰	Š	‹	Œ	□	Ž	□
90	□	‘	’	“	”	•	—	—	~	™	š	›	œ	□	ž	ÿ
A0		ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯
B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿

Ví dụ về Code Page (tt)

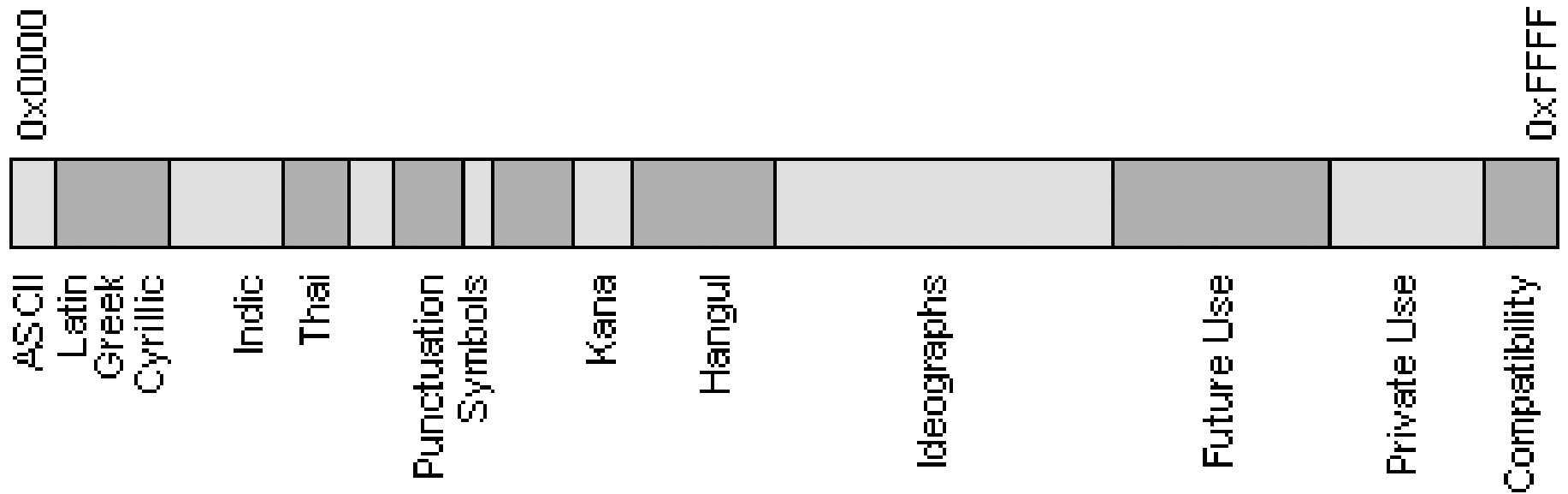
Windows Cyrillic code page (1251)

60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	□
80	Ђ	Ѓ	,	ѓ	„	…	†	‡	€	‰	Љ	‹	Њ	Ќ	Ћ	Ќ
90	ђ	‘	’	“	”	•	—	—	□	™	љ	›	њ	ќ	ћ	џ
A0		Ÿ	ÿ	Ј	Ѡ	Ѓ	Ѕ	§	Ё	©	€	«	¬	-	®	Ї
B0	°	±	І	і	г	μ	¶	·	ё	№	€	»	ј	Ѕ	ѕ	ї
C0	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D0	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
E0	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F0	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

Bộ ký tự Unicode

- Unicode là 1 bộ ký tự 16 bit chứa tất cả các ký tự thường dùng trong xử lý thông tin.
- Unicode là 1 sự đồng ý về cách lưu trữ ký tự, 1 chuẩn được sự hỗ trợ bởi các thành viên của cộng đồng Unicode
- Unicode cung cấp 1 số duy nhất cho mỗi ký tự, không liên quan đến hệ điều hành, chương trình hay ngôn ngữ nào.

Ví dụ bộ ký tự Unicode



Ví dụ bộ ký tự Unicode (tt)

0100

Latin Extended-A

017F

	010	011	012	013	014	015	016	017
0	Ā 0100	Đ 0110	Ġ 0120	İ 0130	ı· 0140	Ŏ 0150	Š 0160	Ů 0170
1	ā 0101	đ 0111	ġ 0121	ı 0131	Ł 0141	ố 0151	š 0161	ů 0171
2	Ǻ 0102	Ē 0112	Ģ 0122	IJ 0132	ł 0142	Œ 0152	Ŧ 0162	Ų 0172
3	ǻ 0103	ē 0113	ģ 0123	ij 0133	Ŋ 0143	œ 0153	ţ 0163	ų 0173
4	Ǽ 0104	Ě 0114	Ĥ 0124	Ĵ 0134	ń 0144	Ŕ 0154	Ţ 0164	Ŵ 0174

Lập trình trực quan

Các khái niệm về Font

- Định nghĩa Font
- Kiểu Font và Họ Font
- Các loại Font trên HĐH Windows
- Các thông số Font
- Độ đo Font

Định nghĩa Font

Tập hợp hoàn chỉnh

- các chữ cái
- các dấu câu
- các con số
- các ký tự đặc biệt

theo một - kiểu loại

- trọng lượng (thường hoặc đậm nét)
- dáng bộ (thẳng hoặc nghiêng)

với kích cỡ phù hợp và có thể phân biệt khác nhau.

Định nghĩa Font (tt)

- Một font có thể được cung cấp bởi 1 hoặc nhiều trang mã
- Mỗi font thường được lưu trong 1 file font
- Mỗi font thường được đặt 1 tên. Ví dụ font Times New Roman Bold, font Courier New Italic

Kiểu Font và Họ Font

Tên Font = Họ Font + (Kiểu Font)

Font name = Font family + (Typestyle)


Ví dụ:

Họ font Times New Roman chứa 4 font khác nhau:

- Times New Roman
- Times New Roman Italic
- Times New Roman Bold
- Times New Roman Bold Italic

Kiểu Font và Họ Font (tt)

Ví dụ:

Roman	Roman	Roman	Roman
Swiss	Swiss	Swiss	Swiss
Modern	Modern	Modern	Modern
<i>Script</i>	Script	<i>Script</i>	Script
Decorative	 ecorative	Decorative	DECORATIVE

Cac loại Font trên HĐH Windows

- Bitmap Font
- Vector Font
- TrueType Font
- OpenType Font

Bitmap Font

- Dùng hình ảnh để hiển thị các điểm ảnh của 1 ký tự
- Thường là những file có dạng *.fon
- Khi hiển thị ký tự có kích thước lớn thường bị nát hình



Vector Font

- Dùng các đoạn thẳng nối với nhau để hiển thị ký tự
- Thường là những file có dạng *.fnt
- Mặc dù hiển thị ký tự có kích thước lớn tốt hơn bitmap font nhưng vẫn chưa có được độ sắc nét cao



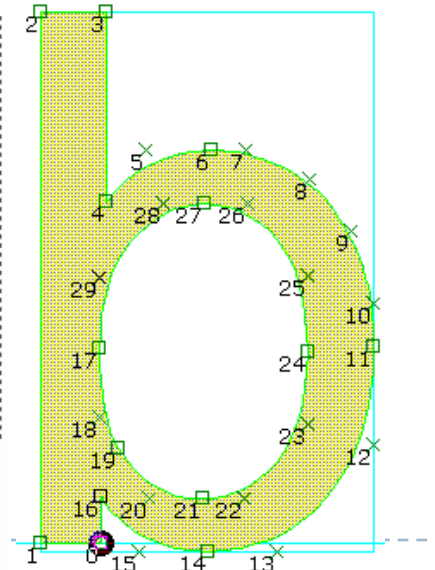
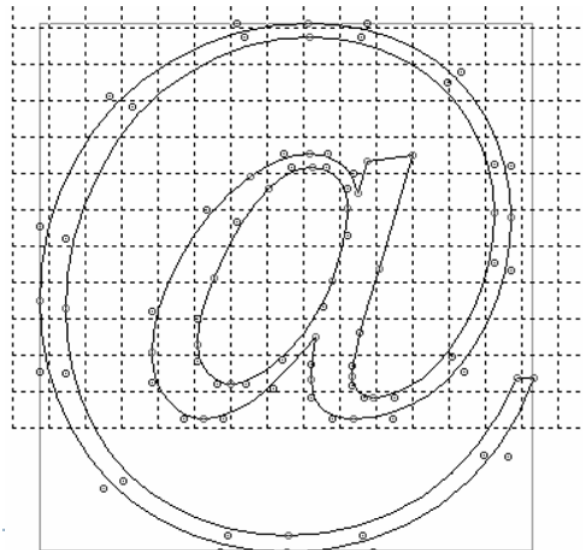
TrueType Font

- Dùng các đoạn thẳng và đoạn cong nối với nhau để hiển thị ký tự
- Thường là những file có dạng *.ttf
- Hiển thị ký tự có kích thước lớn với độ sắc nét cao

Bitmap



TrueType



OpenType Font

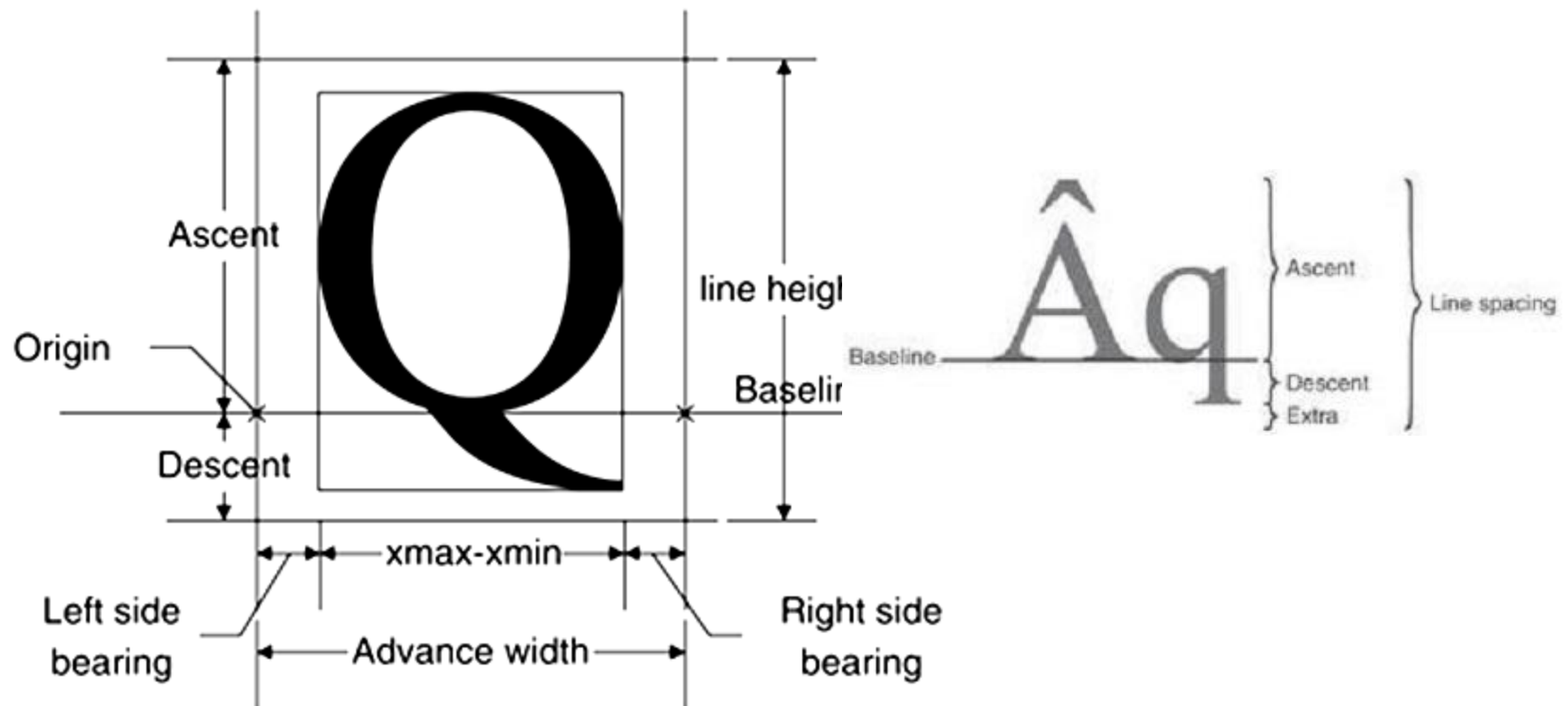
- Là chuẩn định dạng font kết hợp 2 loại định dạng font có sẵn: Type 1 (PostScript) font và TrueType font
- Độc lập với hệ điều hành (cross-platform)
- Hỗ trợ nhiều loại ngôn ngữ trong 1 font
- Thường là những file có dạng *.otf
- Trong Windows, TrueType Font có biểu tượng là chữ TT, còn OpenType Font là chữ O



Các thông số Font (Font metrics)



Font metrics (tt)



Độ đo Font

Font được đo bởi nhiều độ đo:

- pixel: phần tử nhỏ nhất của ảnh mà 1 thiết bị có thể hiển thị (màn hình, máy in)
- point: 1 point = $1/72$ inch trong in ấn
- em: độ rộng của chữ M ứng với kiểu chữ đang dùng
- design unit: dùng để đo kích cỡ 1 họ font bằng độ đo point khi bị thay đổi kích thước

Class Font

- Mô tả lớp Font
- Các thuộc tính lớp Font
- Các hàm khởi tạo lớp Font
 - + Tập hợp FontStyle
 - + Tập hợp GraphicsUnit
- Các phương thức lớp Font
- Các ví dụ

Mô tả lớp Font

- Dùng để xác định cách định dạng văn bản
- Bao gồm các thuộc tính cách thể hiện, kích thước, kiểu dáng
- Không cho phép kế thừa

- **Namespace:** System.Drawing
- **Assembly:** System.Drawing (in system.drawing.dll)

Font Properties

Name	Description
Bold	Gets a value that indicates whether this Font object is bold.
FontFamily	Gets the FontFamily object associated with this Font object.
GdiCharSet	Gets a byte value that specifies the GDI character set that this Font object uses.
GdiVerticalFont	Gets a Boolean value that indicates whether this Font object is derived from a GDI vertical font.
Height	Gets the line spacing of this font.
Italic	Gets a value that indicates whether this Font object is italic.

Font Properties (tt)

Name	Description
Name	Gets the face name of this Font object.
Size	Gets the em-size of this Font object measured in the unit of this Font object.
SizeInPoints	Gets the em-size, in points, of this Font object.
Strikeout	Gets a value that indicates whether this Font object specifies a horizontal line through the font.
Style	Gets style information for this Font object.
Underline	Gets a value that indicates whether this Font object is underlined.
Unit	Gets the unit of measure for this Font object.

Font Constructor

Public Constructors

Name	Description
Font	Overloaded. Initializes a new Font that uses the specified existing Font and FontStyle .

Font Constructor (tt)

Overload List

Font (IntPtr)

Font (Font, FontStyle)

Font (FontFamily, Single)

Font (String, Single)

Font (FontFamily, Single, FontStyle)

Font (FontFamily, Single, GraphicsUnit)

Font (String, Single, FontStyle)

Font (String, Single, GraphicsUnit)

Font (FontFamily, Single, FontStyle, GraphicsUnit)

Font (String, Single, FontStyle, GraphicsUnit)

Font (FontFamily, Single, FontStyle, GraphicsUnit, Byte)

Font (String, Single, FontStyle, GraphicsUnit, Byte)

Font (FontFamily, Single, FontStyle, GraphicsUnit, Byte, Boolean)

Font (String, Single, FontStyle, GraphicsUnit, Byte, Boolean)

Tập hợp FontStyle

Member Name	Value	Description
Bold	0	Bold text
Italic	1	Italic text
Regular	2	Normal text
Strikeout	4	Text with a line through the middle
Underline	8	Underlined text

Tập hợp GraphicsUnit

Member Name	Value	Description
World	0	Units of world coordinates
Display	1	Won't work with font constructor
Pixel	2	Units of pixels
Point	3	Units of 1/72 inch
Inch	4	Units of inch
Document	5	Units of 1/300 inch
Milimeter	6	Units of milimeters

Font Methods

Public Methods

Method	Description
Clone	Creates an exact copy of this Font object.
CreateObjRef	Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object.
Dispose	Releases all resources used by this Font object.
Equals	Overridden. Indicates whether the specified object is a Font object and is identical to this Font object.
FromHdc	Creates a Font object from the specified Windows handle to a device context.

Font Methods (tt)

Public Methods

Method	Description
FromHfont	Creates a Font object from the specified Windows handle.
FromLogFont	Overloaded. Creates a Font object from the specified GDI LOGFONT structure.
GetHashCode	Overridden. Gets the hash code for this Font object.
GetHeight	Overloaded. Returns the line spacing of this font.
GetLifetimeService	Retrieves the current lifetime service object that controls the lifetime policy for this instance.

Font Methods (tt)

Public Methods

Method	Description
GetType	Gets the Font of the current instance.
InitializeLifetimeService	Obtains a lifetime service object to control the lifetime policy for this instance.
ToHfont	Returns a handle to this Font object.
ToLogFont	Overloaded. Creates a GDI LOGFONT structure from this Font object.

Font Methods (tt)

Public Methods

Method	Description
<code>ToString</code>	Overridden. Returns a human-readable string representation of this <code>Font</code> object.

Protected Methods

Method	Description
<code>Finalize</code>	Overridden.
<code>MemberwiseClone</code>	Creates a shallow copy of the current <code>Object</code> .

Ví dụ về Font

Lấy thông tin về font



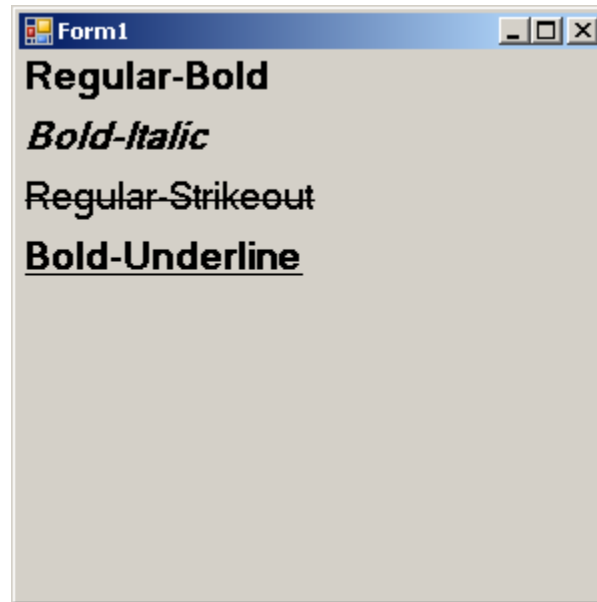
Lấy thông tin về font (tt)

```
private void AllAboutForm_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    Font font = new Font("Time New Roman",12,FontStyle.Regular|FontStyle.Bold);
    string strText = "Name:" + font.Name + "\n" +
                    "FontFamily:" + font.FontFamily + "\n" +
                    "FontStyle:" + font.Style + "\n" +
                    "Bold:" + font.Bold + "\n" +
                    "Italic:" + font.Italic + "\n" +
                    "Underline:" + font.Underline + "\n" +
                    "Strikeout:" + font.Strikeout + "\n" +
                    "Size:" + font.Size + "\n" +
                    "GraphicsUnit:" + font.Unit + "\n" +
                    "SizeInPoints:" + font.SizeInPoints + "\n" +
                    "Height:" + font.Height + "\n" +
                    "GdiCharset:" + font.GdiCharSet + "\n" +
                    "GdiVerticalFont:" + font.GdiVerticalFont + "\n" +
                    "GetHeight:" + font.GetHeight() + "\n" +
                    "GetHeight(Graphics):" + font.GetHeight(g) + "\n" +
                    "GetHeight(100DPI):" + font.GetHeight(100);

    g.DrawString(strText, font, Brushes.Black,Point.Empty);
}
```

Ví dụ FontStyle

Hiển thị tất cả các dạng FontStyle



Hiển thị tất cả các dạng FontStyle (tt)

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    Font font;

    font= new Font("Time New Roman", 14, FontStyle.Regular | FontStyle.Bold);
    g.DrawString("Regular-Bold", font, Brushes.Black, 0,0);

    font = new Font("Time New Roman", 14, FontStyle.Bold | FontStyle.Italic);
    g.DrawString("Bold-Italic", font, Brushes.Black, 0, 30);

    font = new Font("Time New Roman", 14, FontStyle.Regular | FontStyle.Strikeout);
    g.DrawString("Regular-Strikeout", font, Brushes.Black, 0, 60);

    font = new Font("Time New Roman", 14, FontStyle.Bold | FontStyle.Underline);
    g.DrawString("Bold-Underline", font, Brushes.Black, 0, 90);

    font.Dispose();
}
```

Ví dụ về font (tt)

Tạo lớp font mới



Tạo lớp font mới (tt)

```
using System.Drawing;
using System.Drawing.Text;
using System.Windows.Forms;

private void Form1_Paint(object sender, PaintEventArgs e)
{
    Graphics G = e.Graphics;
    G.TextRenderingHint = TextRenderingHint.AntiAlias;
    int y = 0;

    G.DrawString("Regular", MyFonts.Arial_30, Brushes.Black, 50, y += 20);
    G.DrawString("Italic", MyFonts.ArialItalic_30, Brushes.Black, 50, y += 50);
    G.DrawString("Regular", MyFonts.Chain_30, Brushes.Black, 50, y += 60);
    G.DrawString("Italic", MyFonts.ChainItalic_30, Brushes.Black, 50, y += 50);
    G.DrawString("Regular", MyFonts.Comic_30, Brushes.Black, 50, y += 60);
    G.DrawString("Italic", MyFonts.ComicItalic_30, Brushes.Black, 50, y += 50);
}

public sealed class MyFonts
{
    private static PrivateFontCollection PFC;
    private static FontFamily Arial_FF;
    private static FontFamily Comic_FF;
    private static FontFamily Chain_FF;
```

Tạo lớp font mới (tt)

```
static MyFonts()  
{  
    PFC = new PrivateFontCollection();  
    PFC.AddFontFile("H:\\WINNT\\Fonts\\Arial.ttf");  
    Arial_FF = PFC.Families[0];  
    PFC.AddFontFile("H:\\WINNT\\Fonts\\chainletters.ttf");  
    Chain_FF = PFC.Families[1];  
    PFC.AddFontFile("H:\\WINNT\\Fonts\\comic.ttf");  
    Comic_FF = PFC.Families[2];  
}  
  
public static FontFamily[] Families  
{  
    get { return PFC.Families; }  
}  
  
#region Arial Font  
public static Font Arial_30  
{  
    get { return new Font(Arial_FF, 30, FontStyle.Regular, GraphicsUnit.Point); }  
}  
public static Font ArialItalic_30  
{  
    get { return new Font(Arial_FF, 30, FontStyle.Italic, GraphicsUnit.Point); }  
}  
#endregion
```

Tạo lớp font mới (tt)

```
#region Chain font
public static Font Chain_30
{
    get { return new Font(Chain_FF, 30, FontStyle.Regular, GraphicsUnit.Point); }
}
public static Font ChainItalic_30
{
    get { return new Font(Chain_FF, 30, FontStyle.Italic, GraphicsUnit.Point); }
}
#endregion

#region Comic Font
public static Font Comic_30
{
    get { return new Font(Comic_FF, 30, FontStyle.Regular, GraphicsUnit.Point); }
}
public static Font ComicItalic_30
{
    get { return new Font(Comic_FF, 30, FontStyle.Italic, GraphicsUnit.Point); }
}
#endregion
}
```

FontFamily Class

- Mô tả lớp FontFamily
- Các thuộc tính lớp FontFamily
- Các hàm khởi tạo lớp FontFamily
 - + Tập hợp GenericFontFamilies
- Các phương thức lớp FontFamily
- Các ví dụ

Mô tả lớp FontFamily

- Xác nhận một nhóm các font được thiết kế về cách thể hiện và kiểu dáng tương tự nhau
- Không cho phép kế thừa
- **Namespace:** System.Drawing
- **Assembly:** System.Drawing (in system.drawing.dll)

FontFamily Properties

Public Properties

Name	Description
Families	Returns an array that contains all the FontFamily objects associated with the current graphics context.
GenericMonospace	Gets a generic monospace FontFamily.
GenericSansSerif	Gets a generic sans serif FontFamily object.
GenericSerif	Gets a generic serif FontFamily.
Name	Gets the name of this FontFamily.

FontFamily Constructor

Public Constructors

Name	Description
FontFamily	Overloaded. Initializes a new FontFamily that uses the specified name.

Overload List

FontFamily (GenericFontFamilies)
FontFamily (String)
FontFamily (String, FontCollection)

Tập hợp GenericFontFamilies

Member Name	Value	Description
Serif	0	Ví dụ như font Times New Roman
SansSerif	1	Ví dụ như font Arial
Monospace	2	Ví dụ như font Courier New

FontFamily Methods

Public Methods

Name	Description
CreateObjRef	Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object
Dispose	Releases all resources used by this FontFamily.
Equals	Determines whether the specified object equals this FontFamily .
GetCellAscent	Returns the cell ascent, in design units, of the FontFamily of the specified style.
GetCellDescent	Returns the cell descent, in design units, of the FontFamily of the specified style.

FontFamily Methods (tt)

Public Methods

Name	Description
GetEmHeight	Gets the height, in font design units, of the em square for the specified style.
GetFamilies	Returns an array that contains all the FontFamily objects available for the specified graphics context.
GetHashCode	Gets a hash code for this FontFamily .
GetLifetimeService	Retrieves the current lifetime service object that controls the lifetime policy for this instance.
GetLineSpacing	Returns the line spacing, in design units, of the FontFamily of the specified style.

FontFamily Methods (tt)

Public Methods

Name	Description
GetName	Returns the name, in the specified language, of this FontFamily .
GetType	Gets the Type of the current instance.
InitializeLifetimeService	Obtains a lifetime service object to control the lifetime policy for this instance.
IsStyleAvailable	Indicates whether the specified FontStyle enumeration is available.
ReferenceEquals	Determines whether the specified Object instances are the same instance.

FontFamily Methods (tt)

Public Methods

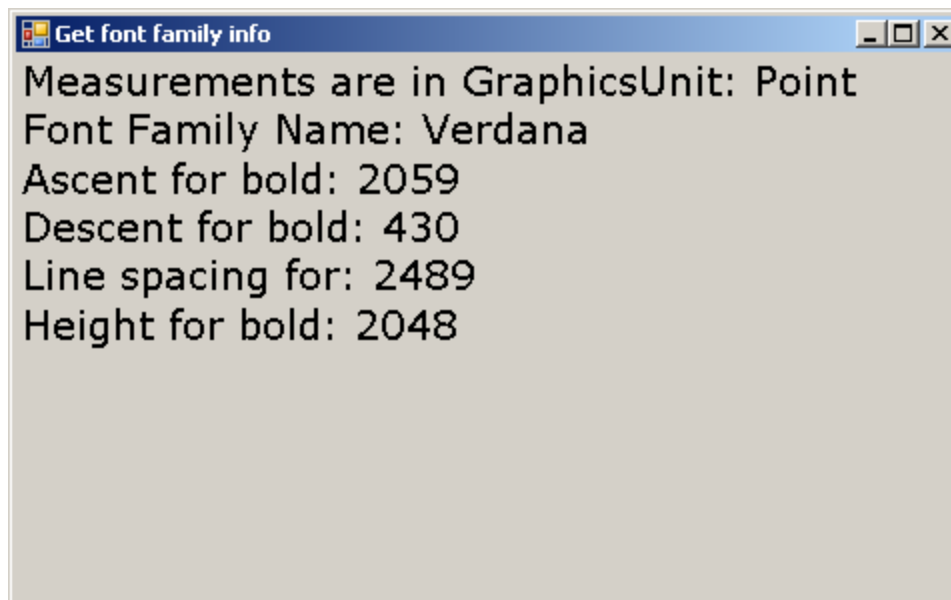
Name	Description
ToString	Converts this FontFamily to a human-readable string representation.

Protected Methods

Name	Description
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.
MemberwiseClone	Overloaded.

Ví dụ về Font Family

Lấy thông tin về font family



Lấy thông tin về font family (tt)

```
-----using System.Drawing;
        using System.Windows.Forms;

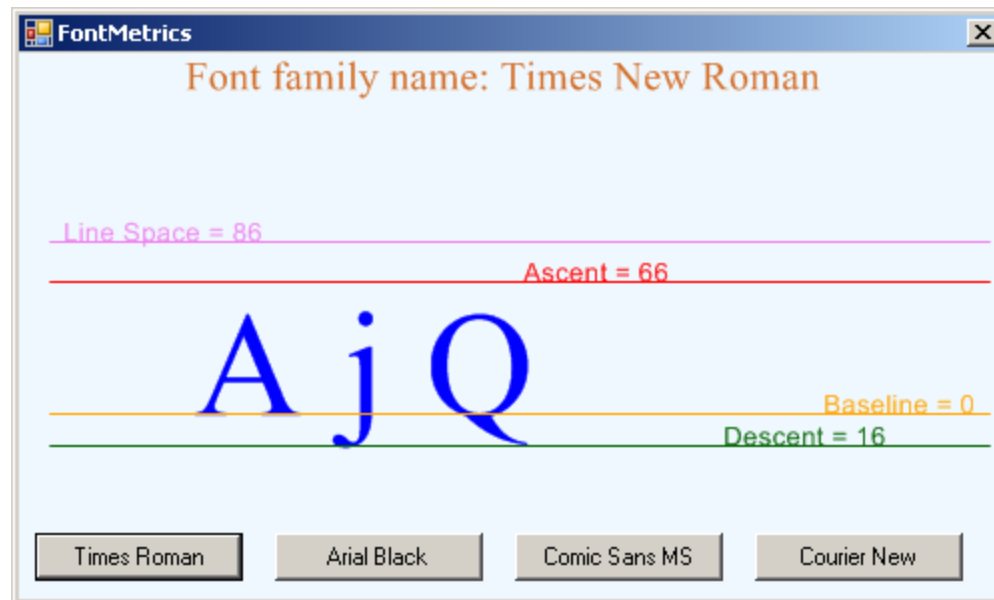
private void Form1_Paint(object sender, PaintEventArgs e)
{
    FontFamily myFamily = new FontFamily("Verdana");
    Font myFont = new Font(myFamily, 15, FontStyle.Regular);
    Graphics g = e.Graphics;

    // Print our Family ties...
    string strText = "Measurements are in GraphicsUnit: " +
                    myFont.Unit.ToString() + "\n" +
                    "Font Family Name: " + myFamily.GetName(0) + "\n" +
                    "Ascent for bold: " +
                    myFamily.GetCellAscent(FontStyle.Bold) + "\n" +
                    "Descent for bold: " +
                    myFamily.GetCellDescent(FontStyle.Bold) + "\n" +
                    "Line spacing for: " +
                    myFamily.GetLineSpacing(FontStyle.Bold) + "\n" +
                    "Height for bold: " +
                    myFamily.GetEmHeight(FontStyle.Bold);

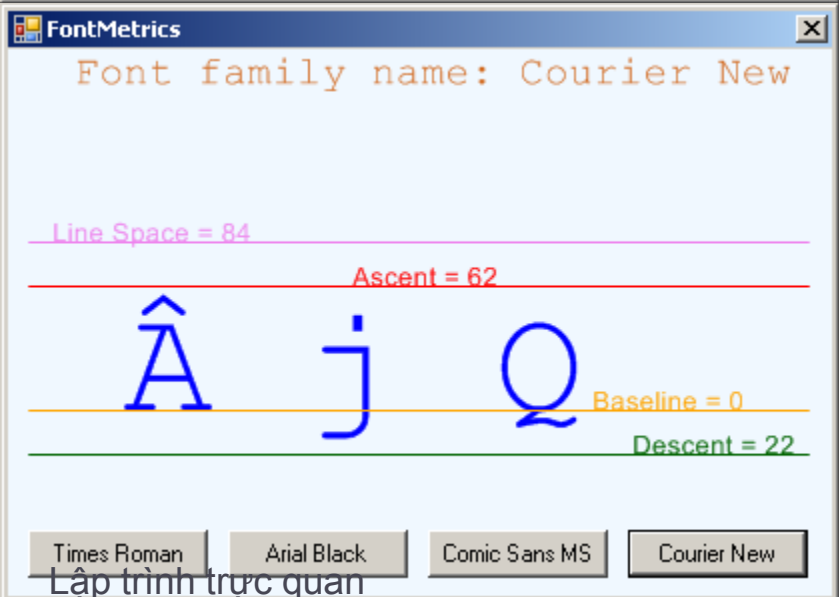
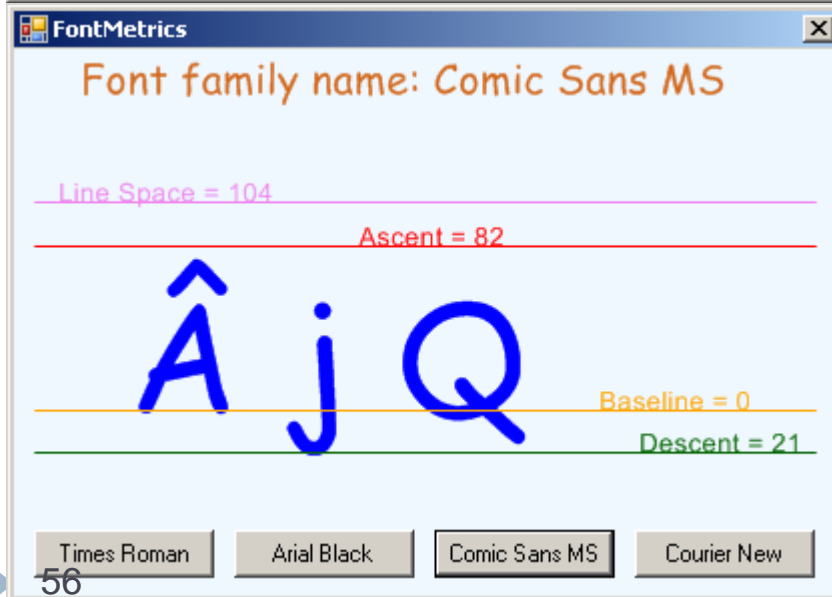
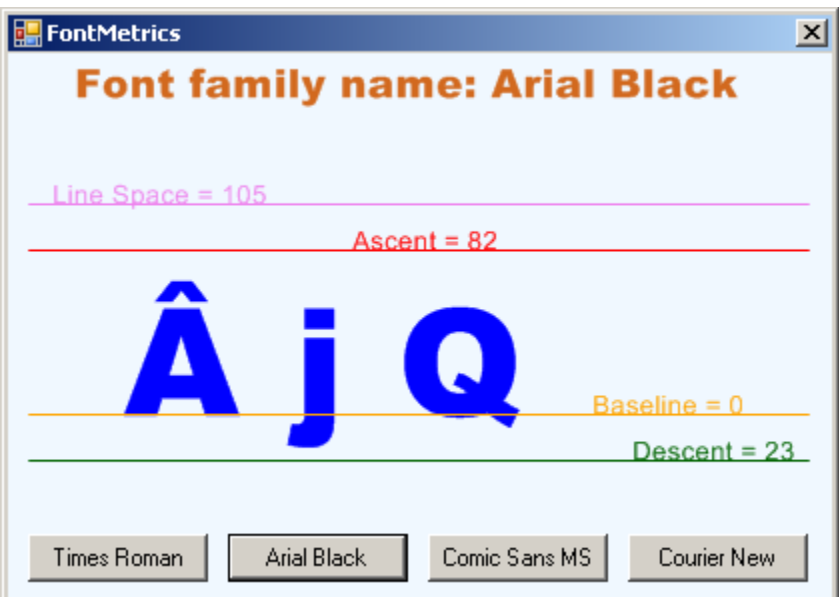
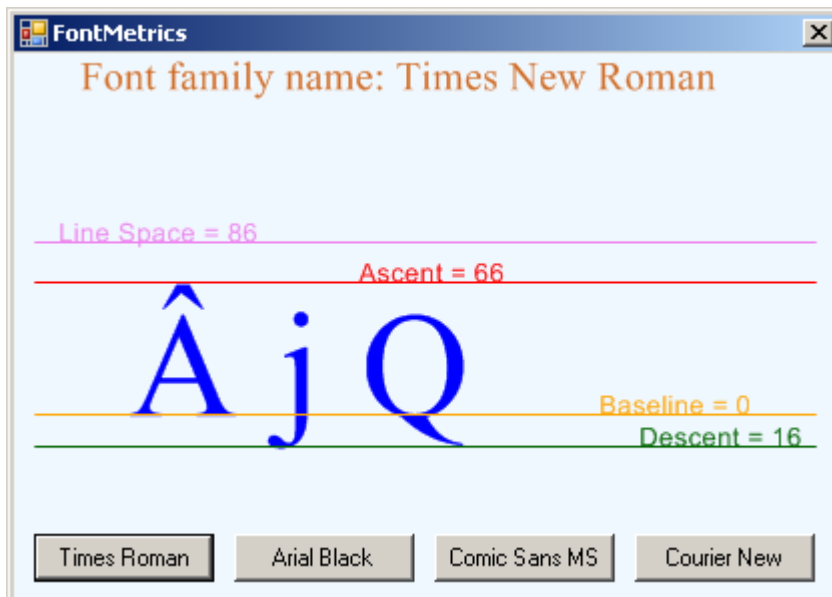
    g.DrawString(strText, myFont, Brushes.Black, Point.Empty);
    myFont.Dispose();
}
```

Ví dụ các thông số Họ Font (tt)

Vẽ các đường thẳng tại các nơi ascent, descent, baseline, line space



Vẽ các đường thẳng (tt)



Vẽ các đường thẳng (tt)

```
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Drawing.Text;
using System.Windows.Forms;

namespace FontMetrics_c
{
    public partial class FontMetrics : Form
    {
        private System.Windows.Forms.Button cmdRoman;
        private System.Windows.Forms.Button cmdArial;
        private System.Windows.Forms.Button cmdComic;
        private System.Windows.Forms.Button cmdCourier;

        private void DisplayFontMetrics(FontFamily fontfamily, Font font)
        {
            //Create graphics object and make it pretty
            Graphics G = this.CreateGraphics();
            G.SmoothingMode = SmoothingMode.AntiAlias;
            G.TextRenderingHint = TextRenderingHint.AntiAlias;

            //Get the Design Unit metrics from the font family
            int emSizeInDU = fontfamily.GetEmHeight(font.Style);
            int LineSpaceInDU = fontfamily.GetLineSpacing(font.Style);
            int DescentInDU = fontfamily.GetCellDescent(font.Style);
            int AscentInDU = fontfamily.GetCellAscent(font.Style);
        }
    }
}
```

Vẽ các đường thẳng (tt)

```
-----//Calculate the Graphics Unit from the font family
float emSizeInGU = font.Size;
int LineSpace = (int)(LineSpaceInDU * emSizeInGU / emSizeInDU);
int Descent = (int)(DescentInDU * emSizeInGU / emSizeInDU);
int Ascent = (int)(AscentInDU * emSizeInGU / emSizeInDU);

//Create the base line to sit the text on
Point BaseLineStart = new Point(10, this.Height * 3 / 5);
Point BaseLineEnd = new Point(this.Width - 20, this.Height * 3 / 5);
//Top left corner of text is the ascent
Point StringPoint = new Point(45, (int)(BaseLineStart.Y - Ascent));

//Clear the screen and draw the string on a base line
G.Clear(Color.AliceBlue);
G.DrawString("Â j Q", font, Brushes.Blue, StringPoint);
G.DrawLine(Pens.Orange, BaseLineStart, BaseLineEnd);

//Draw the annotation lines
Size LineSize = new Size(0, LineSpace);
Size AscentSize = new Size(0, Ascent);
Size DescentSize = new Size(0, Descent);
G.DrawLine(Pens.Violet, BaseLineStart - LineSize, BaseLineEnd - LineSize);
G.DrawLine(Pens.Red, BaseLineStart - AscentSize, BaseLineEnd - AscentSize);
G.DrawLine(Pens.DarkGreen, BaseLineStart + DescentSize, BaseLineEnd + DescentSize);
```

Vẽ các đường thẳng (tt)

```
----- //Annotate
Font myfont = new Font(fontfamily, 15);
G.DrawString("Font family name: " + fontfamily.GetName(0), myfont,
    Brushes.Chocolate, 30, 0);

Font AnnoFont = new Font("Arial", 10);
G.DrawString("Line Space = " + LineSpace.ToString(), AnnoFont,
    Brushes.Violet, 20, (int)(BaseLineStart.Y - LineSpace - 12));

G.DrawString("Ascent = " + Ascent.ToString(), AnnoFont,
    Brushes.Red, 170, (int)(BaseLineStart.Y - Ascent - 12));

G.DrawString("Baseline = 0" , AnnoFont,
    Brushes.Orange, 290, (int)(BaseLineStart.Y - 12));

G.DrawString("Descent = " + Descent.ToString(), AnnoFont,
    Brushes.DarkGreen, 310, (int)(BaseLineStart.Y + Descent - 12));
}

private void cmdRoman_Click(object sender, System.EventArgs e)
{
    FontFamily fontfamily = new FontFamily("Times New Roman");
    Font font = new Font(fontfamily, 75, FontStyle.Regular, GraphicsUnit.Pixel);

    DisplayFontMetrics(fontfamily, font);
}
```

Vẽ các đường thẳng (tt)

```
private void cmdArial_Click(object sender, System.EventArgs e)
{
    FontFamily fontfamily = new FontFamily("Arial Black");
    Font font = new Font(fontfamily, 75, FontStyle.Regular, GraphicsUnit.Pixel);

    DisplayFontMetrics(fontfamily, font);
}

private void cmdComic_Click(object sender, System.EventArgs e)
{
    FontFamily fontfamily = new FontFamily("Comic Sans MS");
    Font font = new Font(fontfamily, 75, FontStyle.Regular, GraphicsUnit.Pixel);

    DisplayFontMetrics(fontfamily, font);
}

private void cmdCourier_Click(object sender, System.EventArgs e)
{
    FontFamily fontfamily = new FontFamily("Courier New");
    Font font = new Font(fontfamily, 75, FontStyle.Regular, GraphicsUnit.Pixel);

    DisplayFontMetrics(fontfamily, font);
}
}
```

Một số thành phần liên quan đến chuỗi ký tự

- Tập hợp `TextRenderingHint`
- Tập hợp `TextFormatFlags`
- Thuộc tính `Graphics.TextContrast`
- Phương thức `Graphics.MeasureString`

Tập hợp TextRenderingHint

- Mô tả tập hợp TextRenderingHint
- Các thành viên tập hợp TextRenderingHint
- Các ví dụ

Mô tả tập hợp TextRenderingHint

- Quy định việc hiển thị chuỗi văn bản theo một chất lượng cụ thể

Namespace: System.Drawing.Text

Assembly: System.Drawing (in system.drawing.dll)

TextRenderingHint

Member Name	Description
AntiAlias	Each character is drawn using its antialiased glyph bitmap without hinting. Better quality due to antialiasing. Stem width differences may be noticeable because hinting is turned off.
AntiAliasGridFit	Each character is drawn using its antialiased glyph bitmap with hinting. Much better quality due to antialiasing, but at a higher performance cost.
ClearTypeGridFit	Each character is drawn using its glyph ClearType bitmap with hinting. The highest quality setting. Used to take advantage of ClearType font features.

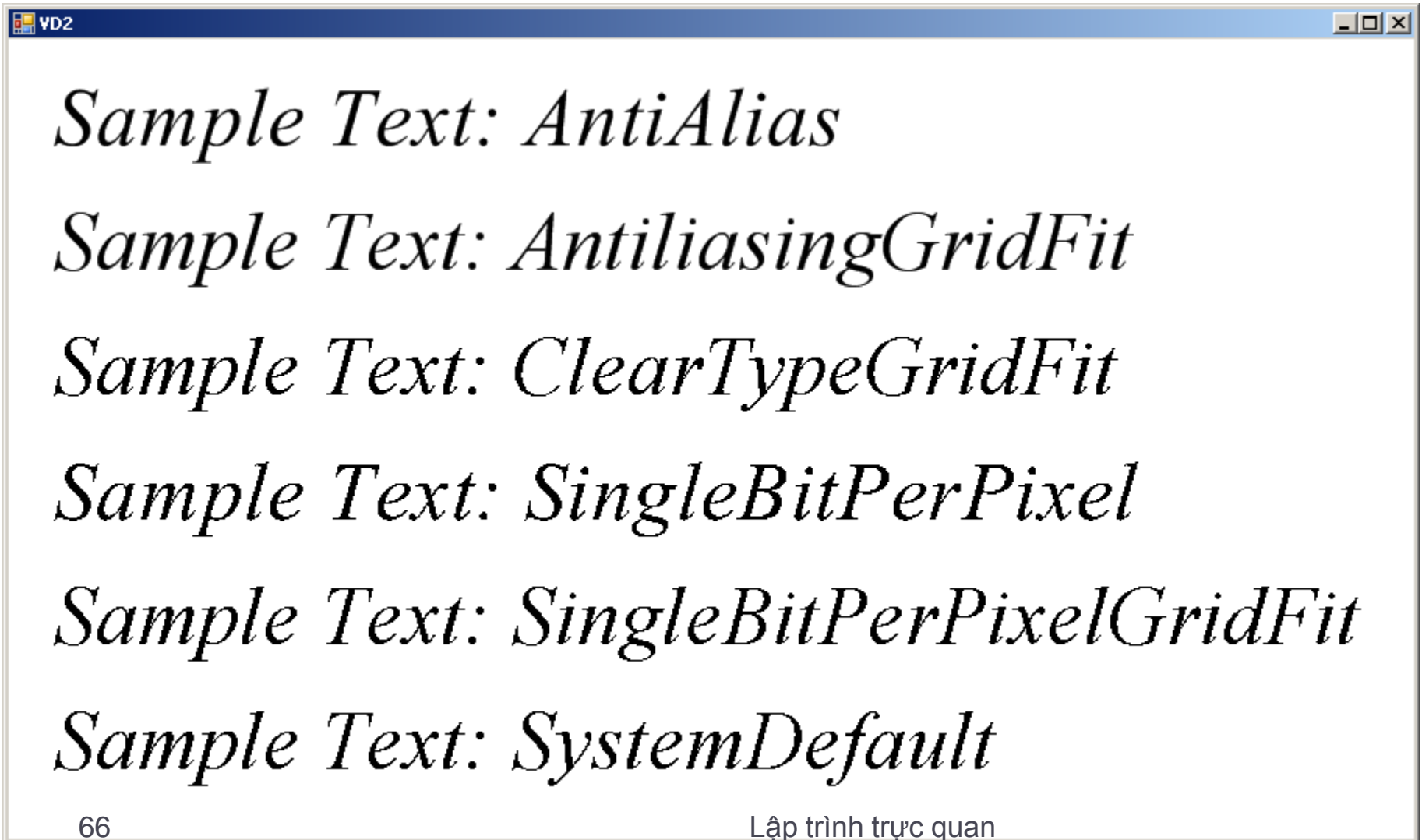
Các thành viên tập hợp

TextRenderingHint (tt)

Member Name	Description
SingleBitPerPixel	Each character is drawn using its glyph bitmap. Hinting is not used.
SingleBitPerPixelGridFit	Each character is drawn using its glyph bitmap. Hinting is used to improve character appearance on stems and curvature.
SystemDefault	Each character is drawn using its glyph bitmap, with the system default rendering hint. The text will be drawn using whatever font-smoothing settings the user has selected for the system.

Ví dụ về TextRenderingHint

Hiển thị chuỗi theo các định dạng của TextRenderingHint -----



Hiển thị chuỗi theo các định dạng của TextRenderingHint (tt)

```
private void VD2_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    Font TextFont = new Font("Times New Roman", 40, FontStyle.Italic);
    g.Clear(Color.White);

    // AntiAlias
    g.TextRenderingHint = TextRenderingHint.AntiAlias;
    g.DrawString("Sample Text: AntiAlias", TextFont, Brushes.Black, 20, 20);
    // AntialiasingGridFit
    g.TextRenderingHint = TextRenderingHint.AntiAliasGridFit;
    g.DrawString("Sample Text: AntialiasingGridFit", TextFont, Brushes.Black, 20, 100);
    // ClearTypeGridFit
    g.TextRenderingHint = TextRenderingHint.ClearTypeGridFit;
    g.DrawString("Sample Text: ClearTypeGridFit", TextFont, Brushes.Black, 20, 180);
    // SingleBitPerPixel
    g.TextRenderingHint = TextRenderingHint.SingleBitPerPixel;
    g.DrawString("Sample Text: SingleBitPerPixel", TextFont, Brushes.Black, 20, 260);
    // SingleBitPerPixelGridFit
    g.TextRenderingHint = TextRenderingHint.SingleBitPerPixelGridFit;
    g.DrawString("Sample Text: SingleBitPerPixelGridFit", TextFont, Brushes.Black, 20,
                                                         340);
    // SystemDefault
    g.TextRenderingHint = TextRenderingHint.SystemDefault;
    g.DrawString("Sample Text: SystemDefault", TextFont, Brushes.Black, 20, 420);
}
```

Tập hợp TextFormatFlags

- Mô tả tập hợp TextFormatFlags
- Các thành viên tập hợp TextFormatFlags
- Minh họa về Padding Flags
- Các ví dụ

Mô tả tập hợp TextFormatFlags

- Quy định sự hiển thị và bố cục của chuỗi ký tự
- **Namespace:** System.Windows.Forms
- **Assembly:** System.Windows.Forms
(in system.windows.forms.dll)




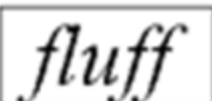
Các thành viên tập hợp TextFormatFlags

Bottom	Internal
Default	Left
EndEllipsis	LeftAndRightPadding
ExpandTabs	ModifyString
ExternalLeading	NoClipping
GlyphOverhangPadding	NoFullWidthCharacterBreak
HidePrefix	NoPadding
HorizontalCenter	NoPrefix

Các thành viên tập hợp TextFormatFlags (tt)

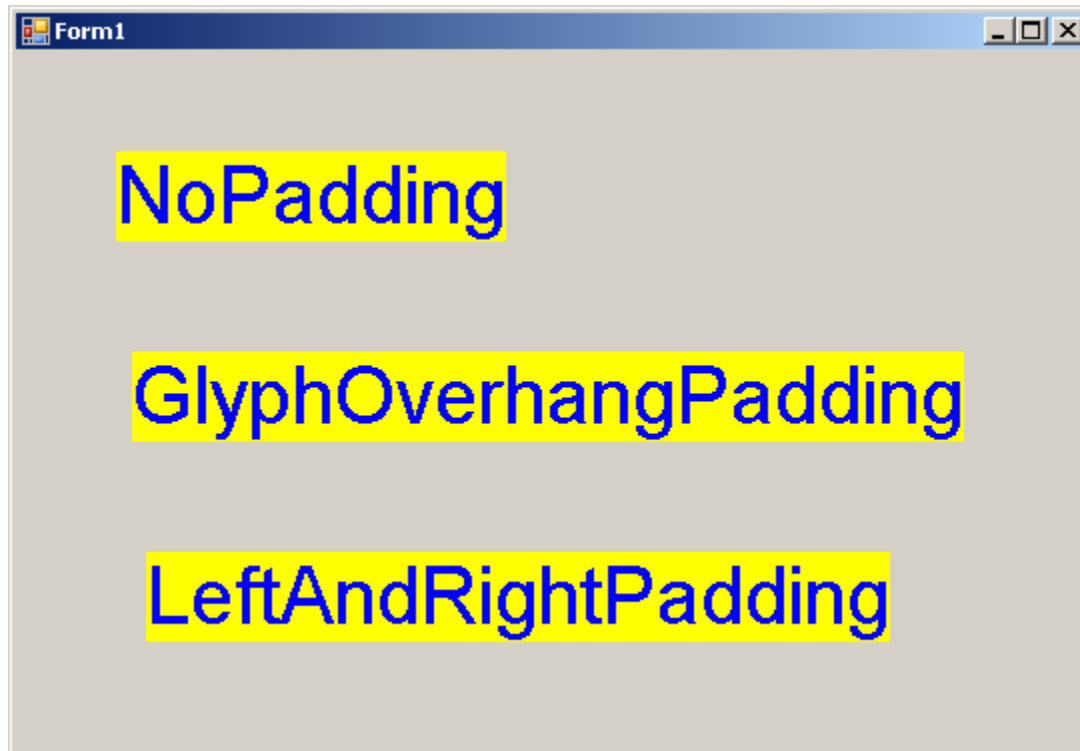
PathEllipsis	Top
PrefixOnly	VerticalCenter
PreserveGraphicsClipping	WordBreak
PreserveGraphicsTranslateTransform	WordEllipsis
Right	
RightToLeft	
SingleLine	
TextBoxControl	

Minh họa về Padding Flags

TextFormatFlags.NoPadding		
TextFormatFlags.GlyphOverhangPadding		
TextFormatFlags.LeftAndRightPadding		

Ví dụ về TextFormatFlags

Hiển thị chuỗi với các Padding Flags



Hiển thị chuỗi với các Padding Flags (tt)

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    Font textFont = new Font("Arial", 30);
    TextFormatFlags flags;

    flags = TextFormatFlags.NoPadding;
    TextRenderer.DrawText(e.Graphics, "NoPadding",
        textFont, new Rectangle(50, 50, 500, 50),
        Color.Blue, Color.Yellow, flags);

    flags = TextFormatFlags.GlyphOverhangPadding;
    TextRenderer.DrawText(e.Graphics, "GlyphOverhangPadding",
        textFont, new Rectangle(50, 150, 500, 50),
        Color.Blue, Color.Yellow, flags);

    flags = TextFormatFlags.LeftAndRightPadding;
    TextRenderer.DrawText(e.Graphics, "LeftAndRightPadding",
        textFont, new Rectangle(50, 250, 500, 50),
        Color.Blue, Color.Yellow, flags);

    // Dispose of the font object.
    textFont.Dispose();
}
```

Thuộc tính Graphics.TextContrast

- Mô tả thuộc tính TextContrast
- Ví dụ về thuộc tính TextContrast

Mô tả thuộc tính TextContrast

- Graphics.TextContrast Property gets or sets the gamma correction value for rendering text.
- **Namespace:** System.Drawing
- **Assembly:** System.Drawing (in system.drawing.dll)

Ví dụ về thuộc tính TextContrast

Change TextRenderingHint And TextContrast



Change TextRenderingHint And TextContrast (tt)

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    // Retrieve the graphics object.
    Graphics formGraphics = e.Graphics;
    // Declare a new font.
    Font myFont = new Font(FontFamily.GenericSansSerif, 40, FontStyle.Regular);

    // Set the TextRenderingHint property.
    formGraphics.TextRenderingHint = System.Drawing.Text.TextRenderingHint.SingleBitPerPixel;
    // Draw the string.
    formGraphics.DrawString("Hello World", myFont, Brushes.Firebrick, 20.0F, 20.0F);

    // Change the TextRenderingHint property.
    formGraphics.TextRenderingHint = System.Drawing.Text.TextRenderingHint.AntiAliasGridFit;
    // Draw the string again.
    formGraphics.DrawString("Hello World", myFont, Brushes.Firebrick, 20.0F, 80.0F);

    // Set the text contrast to a high-contrast setting.
    formGraphics.TextContrast = 0;
    // Draw the string.
    formGraphics.DrawString("Hello World", myFont, Brushes.DodgerBlue, 20.0F, 140.0F);

    // Set the text contrast to a low-contrast setting.
    formGraphics.TextContrast = 12;
    // Draw the string again.
    formGraphics.DrawString("Hello World", myFont, Brushes.DodgerBlue, 20.0F, 200.0F);

    // Dispose of the font object.
    myFont.Dispose();
}
```

Phương thức Graphics.MeasureString

- Mô tả các hàm MeasureString
- Ví dụ về hàm MeasureString

Mô tả các hàm MeasureString

- Đo chuỗi khi chuỗi được vẽ với 1 loại font cụ thể.
- **Namespace:** System.Drawing
- **Assembly:** System.Drawing (in system.drawing.dll)

----- Mô tả các hàm MeasureString (tt) -----
Đo chuỗi khi được vẽ với 1 loại font cụ thể.

MeasureString Method (String, Font) MeasureString
Method (String, Font, int)

- Đo chuỗi khi được vẽ với 1 loại font cụ thể trong 1 vùng thể hiện.

MeasureString Method (String, Font, SizeF)

Mô tả các hàm MeasureString (tt)

- Đo chuỗi khi được vẽ với 1 loại font cụ thể và được định dạng bởi 1 đối tượng StringFormat.

MeasureString Method (String, Font, int, StringFormat)

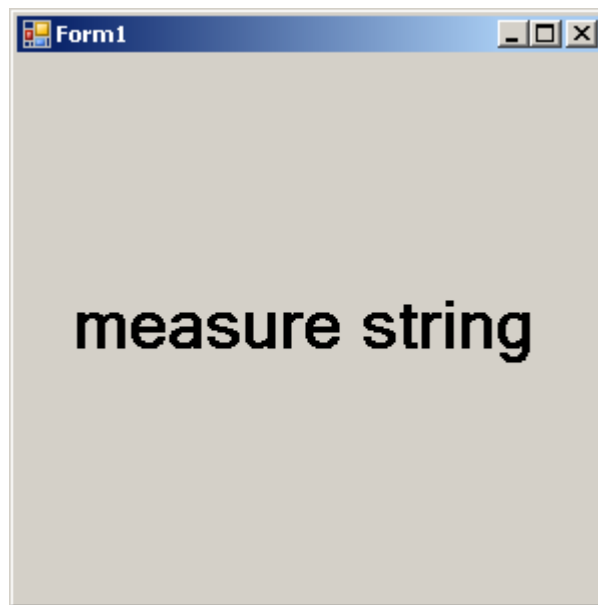
MeasureString Method (String, Font, PointF, StringFormat)

MeasureString Method (String, Font, SizeF, StringFormat)

MeasureString Method (String, Font, SizeF, StringFormat, int, int)

Ví dụ về hàm MeasureString

Hiện thị chuỗi chính giữa form



Hiện thị chuỗi chính giữa form (tt)

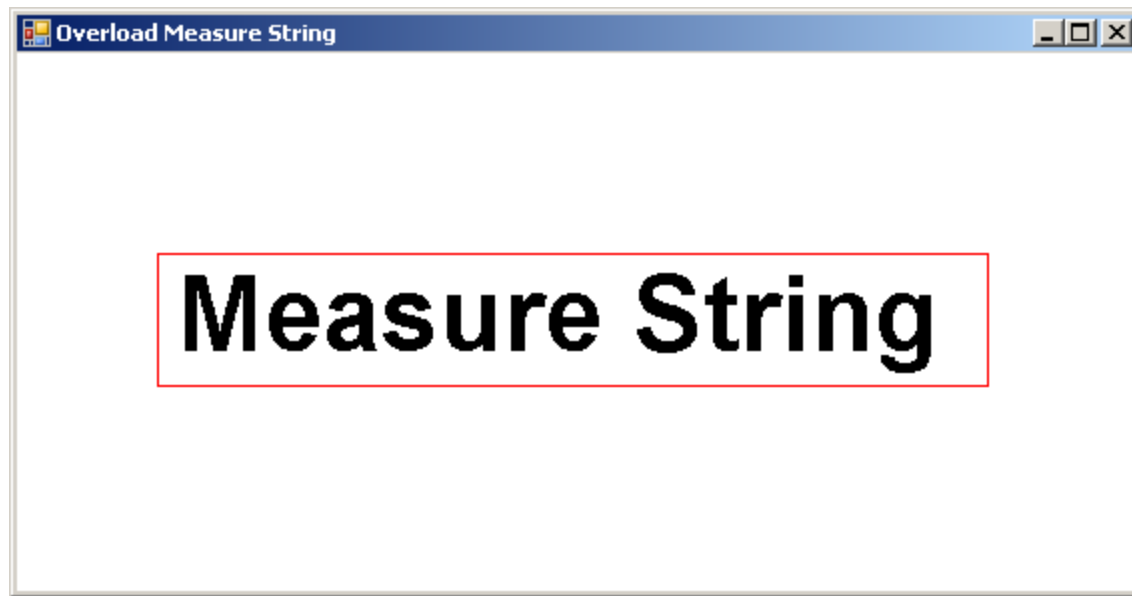
```
using System;
using System.Drawing;
using System.Windows.Forms;

namespace MeasureText
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Paint(object sender, PaintEventArgs e)
        {
            Graphics g = e.Graphics;
            Font font = new Font("Time New Roman", 25, FontStyle.Bold);
            string strText = "measure string";
            SizeF sizef = g.MeasureString(strText, font);
            g.DrawString(strText, font, Brushes.Black,
                (ClientSize.Width - sizef.Width) / 2,
                (ClientSize.Height - sizef.Height) / 2);
        }
    }
}
```

Ví dụ về hàm MeasureString (tt)

Vẽ hình chữ nhật bọc chuỗi



Vẽ hình chữ nhật bọc chuỗi (tt)

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    // Set up string.
    string measureString = "Measure String";
    Font stringFont = new Font("Arial", 40, FontStyle.Bold);
    // Set maximum layout size.
    SizeF layoutSize = new SizeF(500.0F, 500.0F);
    SizeF stringSize = new SizeF();
    //Clear the screen
    e.Graphics.Clear(Color.White);
    //Measure String
    stringSize = e.Graphics.MeasureString(measureString,
                                           stringFont, layoutSize);
    // Draw rectangle representing size of string.
    e.Graphics.DrawRectangle(new Pen(Color.Red, 1), 70.0F, 100.0F,
                             stringSize.Width, stringSize.Height);
    // Draw string to screen.
    e.Graphics.DrawString(measureString, stringFont, Brushes.Black,
                           new PointF(70, 100));
}
```

StringFormat Class

- Mô tả lớp StringFormat
- Các thuộc tính lớp StringFormat
 - + Tập hợp StringFormatFlags
 - + Tập hợp StringTrimming
 - + Tập hợp HotkeyPrefix
- Các hàm khởi tạo lớp StringFormat
- Các phương thức lớp StringFormat
- Các ví dụ

Mô tả lớp StringFormat

- Chứa tất cả các thông tin cách thể hiện (layout) của văn bản (như canh dòng, hướng thể hiện, tab stop)
- Không cho phép kế thừa
- **Namespace:** System.Drawing
- **Assembly:** System.Drawing (in system.drawing.dll)

StringFormat Properties

Public Properties

Name	Description
Alignment	Gets or sets text alignment information on the vertical plane.
DigitSubstitutionLanguage	Gets or sets the language that is used when local digits are substituted for western digits.
DigitSubstitutionMethod	Gets or sets the method to be used for digit substitution.
FormatFlags	Gets or sets a StringFormatFlags enumeration that contains formatting information.

StringFormat Properties (tt)

Public Properties

Name	Description
GenericDefault	Gets a generic default StringFormat object.
GenericTypographic	Gets a generic typographic StringFormat object.
HotkeyPrefix	Gets or sets the HotkeyPrefix object for this StringFormat object.
LineAlignment	Gets or sets the line alignment on the horizontal plane.
Trimming	Gets or sets the StringTrimming enumeration for this StringFormat object.

Tập hợp StringFormatFlags

Member Name	Description
DirectionRightToLeft	Text is displayed from right to left.
DirectionVertical	Text is vertically aligned.
DisplayFormatControl	Control characters such as the left-to-right mark are shown in the output with a representative glyph.
FitBlackBox	Parts of characters are allowed to overhang the string's layout rectangle. By default, characters are repositioned to avoid any overhang.

Tập hợp StringFormatFlags (tt)

Member Name	Description
LineLimit	Only entire lines are laid out in the formatting rectangle. By default layout continues until the end of the text, or until no more lines are visible as a result of clipping, whichever comes first.
MeasureTrailingSpaces	Includes the trailing space at the end of each line. By default the boundary rectangle returned by the MeasureString method excludes the space at the end of each line. Set this flag to include that space in measurement.

Tập hợp StringFormatFlags (tt)

Member Name	Description
NoClip	Overhanging parts of glyphs, and unwrapped text reaching outside the formatting rectangle are allowed to show. By default all text and glyph parts reaching outside the formatting rectangle are clipped.
NoFontFallback	Fallback to alternate fonts for characters not supported in the requested font is disabled.
NoWrap	Text wrapping between lines when formatting within a rectangle is disabled. This flag is implied when a point is passed instead of a rectangle, or when the specified rectangle has a zero line length.

Tập hợp StringTrimming

Member Name	Description
Character	Specifies that the text is trimmed to the nearest character.
EllipsisCharacter	Specifies that the text is trimmed to the nearest character, and an ellipsis is inserted at the end of a trimmed line.
EllipsisPath	The center is removed from trimmed lines and replaced by an ellipsis. The algorithm keeps as much of the last slash-delimited segment of the line as possible.
EllipsisWord	Specifies that text is trimmed to the nearest word, and an ellipsis is inserted at the end of a trimmed line.

Tập hợp StringTrimming (tt)

Member Name	Description
None	Specifies no trimming.
Word	Specifies that text is trimmed to the nearest word.

Tập hợp HotkeyPrefix

Member Name	Description
Hide	Do not display the hot-key prefix.
None	No hot-key prefix.
Show	Display the hot-key prefix.

StringFormat Constructor

Public Constructors

Name	Description
StringFormat	Overloaded. Initializes a new StringFormat object.

Overload List

StringFormat ()
StringFormat (StringFormat)
StringFormat (StringFormatFlags)
StringFormat (StringFormatFlags, Int32)

StringFormat Methods

Public Methods

Name	Description
Clone	Creates an exact copy of this StringFormat object.
CreateObjRef	Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object.
Dispose	Releases all resources used by this StringFormat object.
Equals	Determines whether two Object instances are equal.
GetHashCode	Serves as a hash function for a particular type.

StringFormat Methods (tt)

Public Methods

Name	Description
GetLifetimeService	Retrieves the current lifetime service object that controls the lifetime policy for this instance.
GetTabStops	Gets the tab stops for this StringFormat object.
GetType	Gets the Type of the current instance.
InitializeLifetimeService	Obtains a lifetime service object to control the lifetime policy for this instance.

StringFormat Methods (tt)

Public Methods

Name	Description
ReferenceEquals	Determines whether the specified Object instances are the same instance.
SetDigitSubstitution	Specifies the language and method to be used when local digits are substituted for western digits.
SetMeasurableCharacterRanges	Specifies an array of CharacterRange structures that represent the ranges of characters measured by a call to the MeasureCharacterRanges method.
SetTabStops	Sets tab stops for this StringFormat object.

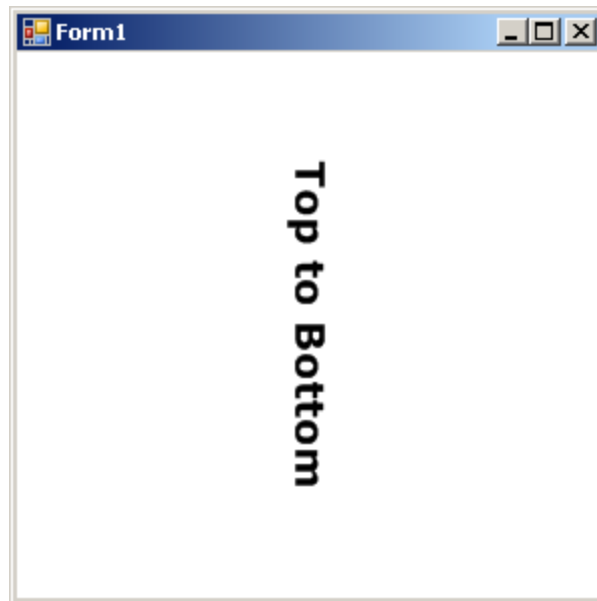
StringFormat Methods (tt)

Protected Methods

Name	Description
Finalize	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.
MemberwiseClone	Overloaded.

Ví dụ về StringFormat

Hiển thị chuỗi theo chiều thẳng đứng



Hiển thị chuỗi theo chiều thẳng đứng (tt)

```
using System.Drawing;
using System.Drawing.Text;
using System.Windows.Forms;

private void Form1_Paint(object sender, PaintEventArgs e)
{
    string strText = "Top to Bottom";
    Graphics g = e.Graphics;
    Font font = new Font("Verdana", 15, FontStyle.Bold);

    // Set string format.
    StringFormat strFormat = new StringFormat();
    strFormat.FormatFlags = StringFormatFlags.DirectionVertical;

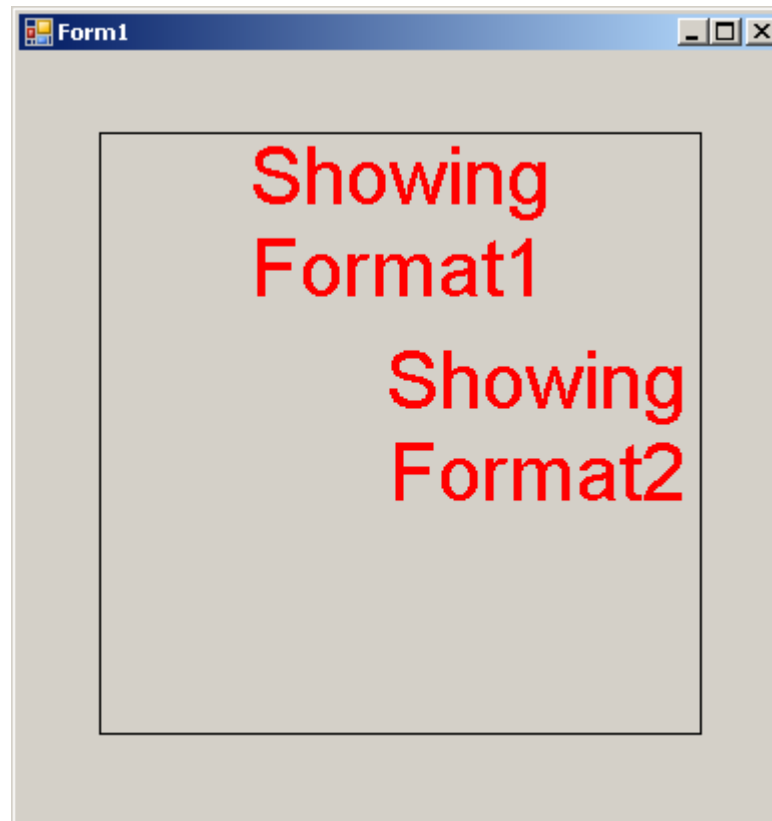
    strFormat.Alignment = StringAlignment.Center;
    strFormat.LineAlignment = StringAlignment.Center;

    // Clear the screen
    g.Clear(Color.White);
    g.TextRenderingHint = TextRenderingHint.AntiAlias;

    // Draw string to screen.
    g.DrawString(strText, font, Brushes.Black,
        new RectangleF(0, 0, ClientRectangle.Width, ClientRectangle.Height), strFormat);
}
```

Ví dụ về StringFormat

Hiển thị chuỗi và canh lề



Hiển thị chuỗi và canh lề (tt)

```
--  
using System.Drawing;                                     --  
using System.Windows.Forms;  
  
private void Form1_Paint(object sender, PaintEventArgs e)  
{  
    Font MyFont = new Font("Arial", 30);  
    // Construct a new Rectangle .  
    Rectangle displayRectangle =  
        new Rectangle(new Point(40, 40), new Size(300, 300));  
  
    // Construct 2 new StringFormat objects  
    StringFormat format1 = new StringFormat(  
        StringFormatFlags.NoClip);  
    StringFormat format2 = new StringFormat(format1);  
  
    // Set the LineAlignment and Alignment properties for  
    // both StringFormat objects to different values.  
    format1.LineAlignment = StringAlignment.Near;  
    format1.Alignment = StringAlignment.Center;  
    format2.LineAlignment = StringAlignment.Center;  
    format2.Alignment = StringAlignment.Far;  
--
```



Hiển thị chuỗi và canh lề (tt)

```
// Draw the bounding rectangle and a string for each
// StringFormat object.
e.Graphics.DrawRectangle(Pens.Black, displayRectangle);
e.Graphics.DrawString("Showing Format1", MyFont,
    Brushes.Red, (RectangleF)displayRectangle, format1);
e.Graphics.DrawString("Showing Format2", MyFont,
    Brushes.Red, (RectangleF)displayRectangle, format2);

MyFont.Dispose();
}
```

TextRenderer Class

- Mô tả lớp TextRenderer
- Các phương thức lớp StringFormat
- Đặc điểm của lớp TextRenderer
- Các thuộc tính của lớp StringFormat và các thành viên tập hợp TextFormatFlags
- Các ví dụ

Mô tả lớp TextRenderer

- Cung cấp phương thức cho việc đo lường và thay đổi hình thức hiển thị của chuỗi văn bản
- Không cho phép kế thừa
- **Namespace:** System.Windows.Forms
- **Assembly:** System.Windows.Forms
(in system.windows.forms.dll)

Lớp TextRenderer

Public Methods

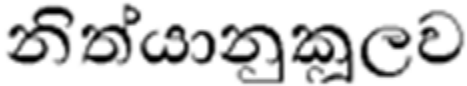
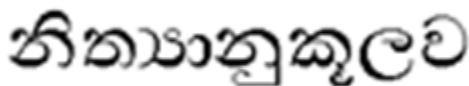
Name	Description
DrawText	Overloaded. Draws the specified text at the specified location, using the specified device context, color, and font.
Equals	Determines whether two Object instances are equal.
GetHashCode	Serves as a hash function for a particular type.
GetType	Gets the Type of the current instance.
MeasureText	Overloaded. Measures the specified text when drawn with the specified font.

Lớp TextRenderer (tt)

Public Methods

Name	Description
ReferenceEquals	Determines whether the specified Object instances are the same instance.
ToString	Returns a String that represents the current Object .

Sự khác biệt của Text Rendering

System.Drawing.Graphics	System.Windows.Forms.TextRenderer
	
Character shapes are not rendered accurately.	Characters are shaped properly with full script support.

Ảnh xạ giá trị các thuộc tính từ lớpStringFormat vào tập hợp TextFormatFlags

StringFormat Property Value	TextFormatFlags Flag Value
Horizontal Alignment	
Alignment.Near	Left
Alignment.Center	HorizontalCenter
Alignment.Far	Right
Vertical Alignment	
LineAlignment.Near	Top
LineAlignment.Center	VerticalCenter
LineAlignment.Far	Bottom
Ellipsis	
Trimming.EllipsisCharacter	End Ellipsis
Trimming.EllipsisWord	WordEllipsis
Trimming.EllipsisPath	PathEllipsis
Trimming.Character	N/A
Trimming.Word	N/A
Trimming.None	<default behavior>

Ảnh xạ giá trị các thuộc tính từ lớp `StringFormat` vào tập hợp `TextFormatFlags` (tt)

Hotkey Prefix	
<code>HotkeyPrefix.None</code>	<code>NoPrefix</code>
<code>HotkeyPrefix.Show</code>	<code><default behavior></code>
<code>HotkeyPrefix.Hide</code>	<code>HidePrefix</code>
N/A	<code>PrefixOnly</code>
Text Padding	
<code>FormatFlags.FitBlackBox *</code>	<code>NoPadding</code>
N/A	<code>LeftAndRightPadding</code>
Text Wrapping	
<code>FormatFlags.NoWrap</code>	<code>SingleLine</code>
<code><default behavior></code>	<code>WordBreak</code>
<code>FormatFlags.LineLimit</code>	<code>WordBreak TextBoxControl</code>
N/A	<code>NoFullWidthCharacterBreak</code>

Ảnh xạ giá trị các thuộc tính từ lớpStringFormat vào tập hợp TextFormatFlags (tt)

Other Flags	
FormatFlags.DirectionRightToLeft	RightToLeft
FormatFlags.NoClip	NoClipping
FormatFlags.DisplayFormatControl	N/A
FormatFlags.NoFontFallBack	N/A
FormatFlags.MeasureTrailingSpaces	N/A
FormatFlags.DirectionVertical	N/A
N/A	PreserveGraphicsClipping
N/A	PreserveGraphicsTranslateTransform
N/A	ExternalLeading
N/A	Internal
N/A	ModifyString
SetTabStops method	ExpandTabs
* Actually means NoFitBlackBox	

Ví dụ về Rendering Text

Rendering Text với TextRenderer



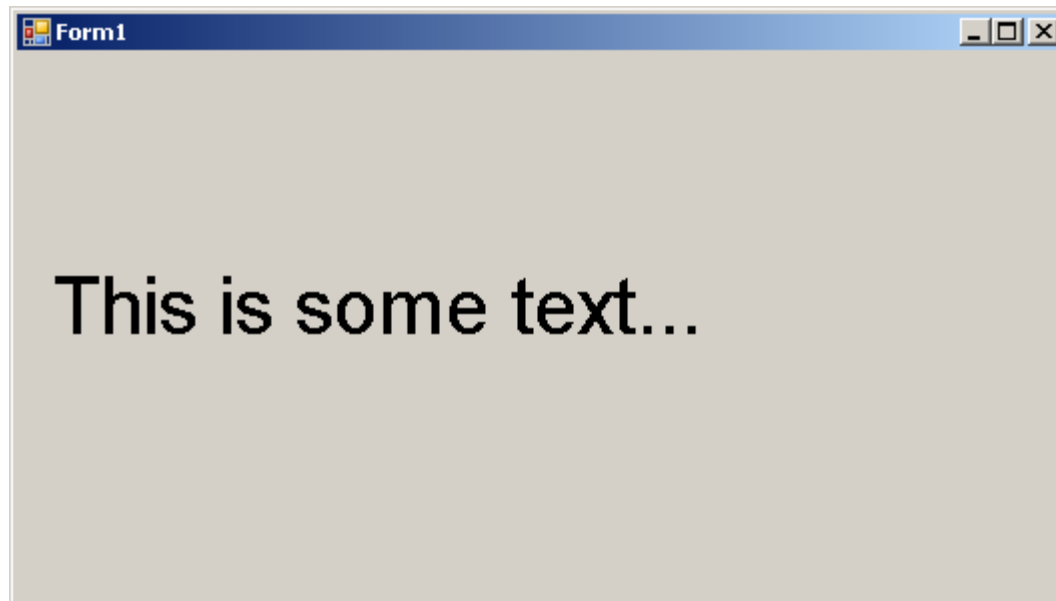
Rendering Text với TextRenderer (tt)

```
Size PaintText(Graphics g, string txt, Font font,
                Point pt, TextFormatFlags flags)
{
    Size size = TextRenderer.MeasureText(g, txt, font,
                                          Size.Empty, flags);
    Rectangle box = new Rectangle(pt, size);
    g.DrawRectangle(Pens.Black, box);
    TextRenderer.DrawText(g, txt, font, box,
                          SystemColors.ControlText, flags);
    return size;
}

private void Form2_Paint(object sender, PaintEventArgs e)
{
    String strText = "This is example\nPaint Text";
    Graphics g = e.Graphics;
    Font font = new Font("Verdana", 12, FontStyle.Bold);
    StringFormat strFormat = new StringFormat();
    Point pt = new Point(80, 100);
    TextFormatFlags flags = TextFormatFlags.HorizontalCenter;
    PaintText(g, strText, font, pt, flags);
}
```

Ví dụ về TextRenderer (tt)

Tự động lược bỏ ký tự và thay bằng dấu ...



Tự động lược bỏ ký tự và thay bằng dấu ... (tt)

```
using System.Drawing;
using System.Windows.Forms;

private void Form1_Paint(object sender, PaintEventArgs e)
{
    Font textFont = new Font("Arial", 30);

    TextFormatFlags flags = TextFormatFlags.Bottom
                           | TextFormatFlags.EndEllipsis;

    TextRenderer.DrawText(e.Graphics,
        "This is some text that will be clipped at the end.",
        textFont, new Rectangle(10, 100, 350, 50),
        SystemColors.ControlText, flags);

    // Dispose of the font object.
    textFont.Dispose();
}
```

Q/A