

フォントの列挙

利用可能なフォントの情報

前回は、直接フォント名を文字列で指定していましたが、Font() コンストラクタでは **System.Drawing.FontFamily** クラスを用いて指定する方法もあります

このクラスは、フォント名 (タイプフェイス) のグループを管理することができ
これを用いれば、確実にシステムにインストールされているフォントを使うことができます

```
Object
  MarshalByRefObject
    FontFamily

public sealed class FontFamily : MarshalByRefObject, IDisposable
```

このクラスは、**FontFamily.Families** 静的プロパティを提供しています
このプロパティは、カレントグラフィック・コンテキストと関連付けられる
FontFamily オブジェクトの配列を返します

```
public static FontFamily[] Families {get;}
```

グラフィック・コンテキストとは、描画対象のデバイスの識別子を意味します
GDI+ において、これは Graphics クラスを示しています
通常、グラフィック・コンテキストはディスプレイを表します

FontFamily からフォント名を取得するには **Font.Name** プロパティを用います

```
public string Name {get;}
```

これらのプロパティを用いることによって、カレントグラフィック・コンテキストで利用可能な
フォント名を取得したり、Font オブジェクトを作成することが可能になります

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

class WinMain : Form {
    public static void Main(string[] args) {
        Application.Run(new WinMain());
    }
    override protected void OnPaint(PaintEventArgs e) {
        Graphics g = e.Graphics;
        FontFamily[] ff = FontFamily.Families;
        for (int i = 0 , y = 0 ; i < ff.Length ; i++ , y += 15) {
            Font ft = new Font("MS Serif" , 12);
            Point pt = new Point(0 , y);
            g.DrawString(ff[i].Name , ft , Brushes.Black , pt);
        }
    }
}
```



このプログラムは、FontFamily クラスを用いて
カレントグラフィック・コンテキストに関連付けられているフォント名を列挙します
おそらく、ウィンドウには表示しきれないほどの数が表示されるでしょう

グラフィックス・コンテキストを指定して FontFamily の配列を取得したい場合は
FontFamily.GetFamilies() メソッドを使います

```
public static FontFamily[] GetFamilies(Graphics graphics);
```

graphics には、対象となる Graphics オブジェクトを指定します
メソッドは、指定したグラフィック・コンテキストで使用可能な FontFamily の配列を返します

また、一般的な Serif、SansSerif、Monospace フォントを取得することもできます
これは **FontFamily.GenericMonospace**、**FontFamily.GenericSansSerif**
FontFamily.GenericSerif、これら静的プロパティで取得できます

```
public static FontFamily GenericMonospace {get;}
public static FontFamily GenericSansSerif {get;}
public static FontFamily GenericSerif {get;}
```

GenericMonospace は一般的な Monospace、GenericSansSerif は SansSerif
GenericSerif は Serifを表す FontFamily オブジェクトを返します

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

```

class WinMain : Form {
    public static void Main(string[] args) {
        Application.Run(new WinMain());
    }
    override protected void OnPaint(PaintEventArgs e) {
        Graphics g = e.Graphics;
        FontFamily[] ff = {
            FontFamily.GenericMonospace ,
            FontFamily.GenericSansSerif ,
            FontFamily.GenericSerif
        };
        for (int i = 0 , y = 0 ; i < ff.Length ; i++ , y += 15) {
            Font ft = new Font(ff[i] , 12);
            Point pt = new Point(0 , y);
            g.DrawString(ff[i].Name , ft , Brushes.Black , pt);
        }
    }
}

```



このプログラムは、それぞれの FontFamily を配列に格納し
各フォントでフォント名を列挙しています

この他に FontFamily クラスのコンストラクタでインスタンスを生成し
そこから、必要な情報を取得するという方法もあります
FontFamily クラスのコンストラクタは、次のようなものがあります

```

public FontFamily(GenericFontFamilies genericFamily);
public FontFamily(string name);
public FontFamily(string name , FontCollection fontCollection);

```

genericFamily には、一般的なフォントを表す GenericFontFamilies 列挙型を指定します
name は作成する FontFamily インスタンスが表すフォントの名前を
fontCollection は、この FontFamily を含む FontCollection オブジェクトを指定します

FontCollection 型については、この後で詳しく解説します
GenericFontFamilies を使う場合は、静的プロパティの Generic* を使うケースと同じです
これは **System.Drawing.Text.GenericFontFamilies** 列挙型です

```

public enum GenericFontFamilies

```

System.Drawing.Text 名前空間は、フォントなどに関連した型を提供しています
GenericFontFamilies には、以下のようなメンバが定義されています

メンバ	解説
Monospace	一般的な Monospace フォントを示す
SansSerif	一般的な SansSerifフォントを示す
Serif	一般的な Serifフォントを示す

```

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

class WinMain : Form {
    public static void Main(string[] args) {
        Application.Run(new WinMain());
    }
    override protected void OnPaint(PaintEventArgs e) {
        Graphics g = e.Graphics;
        FontFamily[] ff = {
            new FontFamily("Mural Script") ,
            new FontFamily(System.Drawing.
                Text.GenericFontFamilies.Serif)
        };
        for (int i = 0 , y = 0 ; i < ff.Length ; i++ , y += 15) {
            Font ft = new Font(ff[i] , 12);
            Point pt = new Point(0 , y);
            g.DrawString(ff[i].Name , ft , Brushes.Black , pt);
        }
    }
}

```

このプログラムは、生成した FontFamily が示すフォントでフォントの名前を描画します

System.Drawing.Text.FontCollection クラスを用いれば
インストールされているフォントや、独自のフォントなどを取得できます

```

public abstract class FontCollection : IDisposable

```

このクラスは、フォントのコレクションを表すためのものです
ここで重要なのは **FontCollection.Families** プロパティです

```

public FontFamily[] Families {get;}

```

これは、フォントコレクションが表す `FontFamily` 型の配列を取得するプロパティです
フォントコレクションが何を表しているかは、この抽象クラスを実体化しているクラスによります
例えば **`System.Drawing.Text.InstalledFontCollection`** クラスは
システム上にインストールされているフォントコレクションを管理しています

```
Object
  FontCollection
    InstalledFontCollection

public sealed class InstalledFontCollection : FontCollection
```

このクラスのコンストラクタは、引数の無いコンストラクター一つだけです
インスタンスを作成すれば、あとは `Families` プロパティで配列を取得できます

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

class WinMain : Form {
    public static void Main(string[] args) {
        Application.Run(new WinMain());
    }
    override protected void OnPaint(PaintEventArgs e) {
        Graphics g = e.Graphics;
        FontFamily[] ff = (new InstalledFontCollection()).Families;
        for (int i = 0 , y = 0 ; i < ff.Length ; i++ , y += 15) {
            Font ft = new Font("MS Serif" , 12);
            Point pt = new Point(0 , y);
            g.DrawString(ff[i].Name , ft , Brushes.Black , pt);
        }
    }
}
```

このプログラムは、システムにインストールされているフォントを列挙します