リッチテキスト

書式付テキスト情報

テキストボックスは、純粋なテキストデータを取り扱うのに最も適したコントロールです。
しかし、テキストの中に色やフォントなどの書式を扱う情報が必要な場合は
テキストボックスコントロールでは、この要求に応えることができません

テキストボックスも Control クラスを継承しているので ForeColor や
Font プロパティなどの設定を変えることで、色やフォントを変更することができます
ですがそれは、テキストボックス全体の概観を変更するに過ぎません

そこで、テキストの一部の色やフォントを変更したいという場合は
System.Windows.Forms.RichTextBox クラスを用います
このクラスはリッチテキストをボックスをカプセル化しています

```
System.Object
   System.MarshalByRefObject
      System.ComponentModel.Component
         System.Windows.Forms.Control
            System.Windows.Forms.TextBoxBase
               System.Windows.Forms.RichTextBox

public class RichTextBox : TextBoxBase
```

このクラスのコンストラクタも、他のコントロール関連クラスと同様に
コンストラクタはデフォルトコンストラクタしか定義されていません

リッチテキストボックスは、Microsoft が書式付の文書を表現できるように
1986 年に考案した仕様の RTF と呼ばれる形式のドキュメントを扱えます
このフォーマットを採用すれば、より表現豊かな文書を作成することができます

特定の場所の色を変更するには RichTextBox.SelectionColor プロパティを
フォントを変更するには RichTextBox.SelectionFont プロパティを使います

public Color SelectionColor {get; set;}
public Font SelectionFont {get; set;}

これらのプロパティを設定すると、現在の選択範囲の色やフォントを変更します
get アクセッサから情報を取得する場合は、選択範囲の色やフォントを取得しますが
選択範囲が複数のフォントや色を含んでいる場合は取得することができません
その場合、フォントは null を、色は Color.Empty を返します

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

public class WinMain : Form {
 MenuItem[] menuItem;
 RichTextBox textBox1 = new RichTextBox();

 public static void Main() {
  Application.Run(new WinMain());
 }

 public WinMain() {
  menuItem = new MenuItem[] {
   new MenuItem("色を変える" , new EventHandler(menuItemClick)) ,
   new MenuItem("フォントを変える" , new EventHandler(menuItemClick)) ,
  };
  textBox1.Dock = DockStyle.Fill;
  textBox1.ContextMenu = new ContextMenu(menuItem);
  Controls.Add(textBox1);
 }

 private void menuItemClick(object sender , EventArgs e) {
  if(sender == menuItem[0])
   textBox1.SelectionColor = Color.Red;
  else if(sender == menuItem[1])
   textBox1.SelectionFont = new Font("MS Serif" , 20);
 }
}
```

図を見てわかるように、リッチテキストの中には複数のフォントや色を含ませることができ
より複雑なドキュメントの作成や編集に適しているでしょう

RTF 形式の入出力

リッチテキストボックスを用いて書式付のテキストを扱えるようになりましたが
誰もが、これをディスクに保存する方法を知りたいと思うことでしょう

方法のひとつとしては、リッチテキストを解析して XHTML に変換することです

.NET では XML 名前空間が定義されており、高度なレベルでマークアップを制御できます
ですが、そんな面倒なことをせずに RTF 形式で保存する方法がもっともスマートです
(ただし、そのようなローカルな仕様のファイル形式をユーザーが望むかどうかは別です)

リッチテキストボックスに入力されている情報を RTF として保存する方法は
実は RichTextBox クラスがメソッドとしてすでに実装しています
RTF の保存には **RichTextBox.SaveFile()** メソッドを用います

public void SaveFile(string path);
public void SaveFile(Stream data , RichTextBoxStreamType fileType);
public void SaveFile(string path , RichTextBoxStreamType fileType);

path には保存するファイル名を、data にはストリームを指定します
Stream に対して保存すれば、ディスク以外の保存領域(主記憶)などに保存できます
fileType は、リッチテキストの入出力ストリーム方を指定します

RichTextBoxStreamType は次のように定義されています

[Serializable]
public enum RichTextBoxStreamType

この列挙型のメンバは次のような意味を持っています
例えば、この列挙型を用いて文字コードを Unicode または ASCII などを指定できます

メンバ	意味
PlainText	OLE (Object Linking and Embedding) オブジェクトの代わりに 空白が含まれているプレーン テキスト ストリーム
RichNoOleObjs	OLE オブジェクトの代わりの空白を持つリッチ テキスト形式 (RTF: Rich Text Format) のストリーム この値は、RichTextBox コントロールの SaveFile メソッドを使用する場合に限り有効です
RichText	リッチ テキスト形式 (RTF) ストリーム
TextTextOleObjs	OLE オブジェクトのテキスト表現を持つプレーン テキスト ストリーム この値は、RichTextBox コントロールの SaveFile メソッドを使用する場合に限り有効です
UnicodePlainText	OLE オブジェクトの代わりに空白が含まれているテキスト ストリーム テキストは Unicode でエンコードされています

SaveFile() メソッドを用いてディスクにリッチテキストを保存すれば
Microsoft Word やワードパットなどで開くことができます

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

public class WinMain : Form {
 MenuItem[] menuItem;
 RichTextBox textBox1 = new RichTextBox();

 public static void Main() {
  Application.Run(new WinMain());
 }

 public WinMain() {
  menuItem = new MenuItem[] {
   new MenuItem("色を変える" , new EventHandler(menuItemClick)) ,
   new MenuItem("フォントを変える" , new EventHandler(menuItemClick)) ,
   new MenuItem("保存" , new EventHandler(menuItemClick)) ,
  };
  textBox1.Dock = DockStyle.Fill;
  textBox1.ContextMenu = new ContextMenu(menuItem);
  Controls.Add(textBox1);
 }

 private void menuItemClick(object sender , EventArgs e) {
  if(sender == menuItem[0])
   textBox1.SelectionColor = Color.Red;
  else if(sender == menuItem[1])
   textBox1.SelectionFont = new Font("MS Serif" , 20);
  else if (sender == menuItem[2])
   textBox1.SaveFile("test.rtf" , RichTextBoxStreamType.RichNoOleObjs);
 }
}
```

これは、先ほどのプログラムに保存メニューを設けた拡張版です
保存を選択すると、プログラムを起動したフォルダに test.rtf ファイルが作成されます

逆に RTF 形式のデータをリッチテキストボックスに入力するには
同様に **RichTextBox.LoadFile()** メソッドを用いて読み込むことができます

public void LoadFile(string path);
public void LoadFile(Stream data , RichTextBoxStreamType fileType);
public void LoadFile(string path , RichTextBoxStreamType fileType);

パラメータの意味は SaveFile() メソッドと同じです

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

```
public class WinMain : Form {
 RichTextBox textBox1 = new RichTextBox();

 public static void Main() {
  Application.Run(new WinMain());
 }

 public WinMain() {
  textBox1.Dock = DockStyle.Fill;
  textBox1.LoadFile("test.rtf");
  Controls.Add(textBox1);
 }
}
```

このプログラムは、カレントフォルダ内の test.rtf ファイルを読み込みます
先ほどのプログラムで出力したファイルを、このプログラムで確認することができるでしょう

因みに、RTF 形式で保存される書式情報はバイナリではなくテキストデータです
RTF の情報も含めた完全なテキスト情報がほしければ **RichTextBox.Rtf** プロパティを使います

public string Rtf {get; set;}

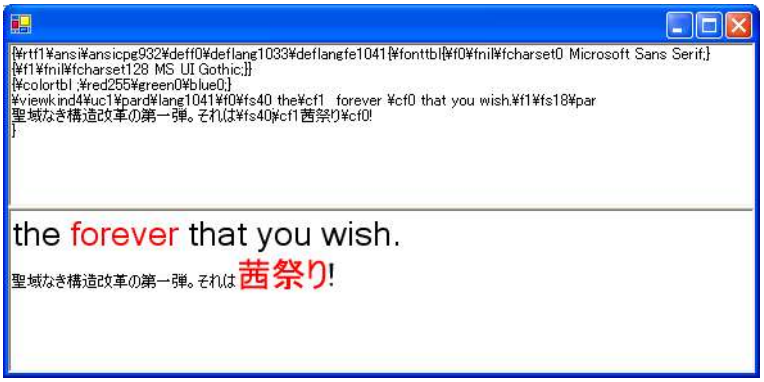
このプロパティを用いれば、RTF テキストを取得したり
あるいはプログラムから直接 RTF テキストを入力することができるようになります

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

public class WinMain : Form {
 TextBox textBox1 = new TextBox();
 RichTextBox textBox2 = new RichTextBox();

 public static void Main() {
  Application.Run(new WinMain());
 }

 public WinMain() {
  textBox1.Dock = DockStyle.Top;
  textBox1.Multiline = textBox2.Multiline = true;
  textBox1.TextChanged += new EventHandler(textBox1_TextChanged);
  textBox2.Dock = DockStyle.Bottom;
  textBox2.ReadOnly = true;
  Controls.AddRange(new Control[] {textBox1 , textBox2});
 }
 private void textBox1_TextChanged(object sender , EventArgs e) {
  try {textBox2.Rtf = textBox1.Text;}
  catch(Exception) {}
 }
 override protected void OnLayout(LayoutEventArgs e) {
  textBox1.Height = textBox2.Height = ClientSize.Height / 2;
  base.OnLayout(e);
 }
}
```



このプログラムは、リッチテキスト形式の生書きを学習するのに適したプログラムです
上の通常のテキストボックスにリッチテキストを入力すると、下部にプレビューされます
RTF 形式の詳細はこの場では説明しませんし、できません
逆に、リッチテキストをリアルタイムでテキストボックスに表示させれば
ある程度、リッチテキスト形式の仕様を理解することができるでしょう
本格的に学習したいのであれば、Microsoft の文献を参照してください

前のページへ　戻る　次のページへ