

Управление **RichEdit** (мы ведем разговор о стандартном управлении Windows, а не о компоненте Delphi) содержит встроенные особенности печати, которые могут использоваться, чтобы послать форматированный текст на принтер или рисовать его на любом холсте с минимальными усилиями для программиста.

Конечно, в стандартный компонент Delphi инкапсулированы эти возможности. Мы можем использовать это, чтобы сделать предварительный просмотр печати с масштабированием или рисованием **RichEdit** на любом управлении. Рисование из управления **RichEdit** на любой холст

использует стандартное сообщение

**EM\_FORMATRANGE**

управления

**RichEdit**

. Параметр

**LParam**

для этого сообщения – указатель на запись

**TFormatRange**

. Эта запись должна быть заполнена перед посылкой сообщения

**RichEdit**

.

Запись **TFormatRange** содержит информацию, которую **RichEdit** управление использует для форматирования вывода для определенного устройства, где:

- **hDC** – устройство для передачи
- **hDCtarget** – устройство, для которого производится форматирование
- **rc** – область передачи. Единицы измерения – *twips*. *Twips* – независимая от экран единица, чтобы гарантировать, что пропорция элементов экрана одинаковая на всех системах отображения.

*wip*

определен как 1/1440 дюйма

- **pcPage** – полная область, предоставленная устройством. Единицы измерения – *twips*

- **chrg** – запись **TCharRange**, которая определяет диапазон текста для форматирования

Эта запись обычно используется с сообщениями **EM\_EXGETSEL** и **EM\_EXSETSEL** и включает два поля:

*cpMin*  
и  
*cpMax*

- *cpMin* – индекс позиции символа, предшествующий первому символу в диапазоне
- *cpMax* – положение символа сразу после последнего символа в диапазоне.

```
function PrintRTFToBitmap(ARichEdit : TRichEdit; ABitmap : TBitmap) : Longint; var
range : TFormatRange; begin  FillChar(Range, SizeOf(TFormatRange), 0); // При
предоставлении к тому же DC мы измеряем.  Range.hdc := ABitmap.Canvas.handle;
Range.hdcTarget := ABitmap.Canvas.Handle; // Set up the page.  Range.rc.left := 0;
Range.rc.top := 0;  Range.rc.right := ABitmap.Width * 1440 div Screen.PixelsPerInch;
Range.rc.Bottom := ABitmap.Height * 1440 div Screen.PixelsPerInch; // По умолчанию
диапазон текста для печати – весь документ  Range.chrg.cpMax := -1;
Range.chrg.cpMin := 0; // форматируем текст  Result := SendMessage(ARichedit.Handle,
EM_FORMATRANGE, 1, Longint(@Range)); // очищаем информацию в кэше
SendMessage(ARichEdit.handle, EM_FORMATRANGE, 0,0); end;
```

Следующий пример показывает, как рисовать RichEdit на любом холсте, а также и рисовать только выбранный диапазон текста.

```
function PrintToCanvas(ACanvas : TCanvas; FromChar, ToChar : integer; ARichEdit :
TRichEdit; AWidth, AHeight : integer) : Longint; var  Range : TFormatRange; begin
FillChar(Range, SizeOf(TFormatRange), 0);  Range.hdc := ACanvas.handle;
Range.hdcTarget := ACanvas.Handle;  Range.rc.left := 0;  Range.rc.top := 0;  Range.rc.right
:= AWidth * 1440 div Screen.PixelsPerInch;  Range.rc.Bottom := AHeight * 1440 div
Screen.PixelsPerInch;  Range.chrg.cpMax := ToChar;  Range.chrg.cpMin := FromChar;
Result := SendMessage(ARichedit.Handle, EM_FORMATRANGE, 1, Longint(@Range));
SendMessage(ARichEdit.handle, EM_FORMATRANGE, 0,0); end;
```

Но как рисовать **RichEdit** с фоновым изображением?

Это безнадежно со стандартным управлением **TRichedit**, потому что он основан на Windows управлении и не имеет никакого дескриптора, чтобы обращаться с фоновым изображением или прозрачностью.

В этом случае мы можем использовать два различных точечных рисунка для фона и **RichEdit**

и после этого их объединить.

```
procedure TForm1.Button2Click(Sender: TObject); var Bmp : TBitmap; begin  Bmp :=  
TBitmap.Create; bmp.Width := 300; bmp.Height := 300;  
PrintToCanvas(bmp.Canvas,2,5,RichEdit1,300,300);  
BitBlt(Image1.Picture.Bitmap.Canvas.Handle, 0, 0,      Bmp.Width, Bmp.Height,  
bmp.Canvas.Handle, 0, 0, srcAND);  Image1.Repaint;  bmp.Free; end;
```

Эта программа разработана Serge Perevoznyk, спасибо, Сергею