基于 RichTextBox 控件的 WYSIWYG 显示

WYSIWYG Display based on RichTextBox Control

刘 超（桂林电子工业学院通信与信息工程系 541004）

摘要：WYSIWYG 是对编辑软件所普遍要求具备的一种关键技术，该技术使得软件打印输出效果和屏幕显示效果完全一致。RichTextBox 是 VB 所带的标准控件，该控件具备基本的图文显示、编辑和简单的格式设置功能，可以应付普通图文处理的需要。本文主要讨论如何对 RichTextBox 控件进行设置和编程，使其能够按打印输出的结果来显示其内容，即如何实现 RichTextBox 控件的 WYSIWYG 显示。

关键词：RichTextBox 控件 所见即所得 显示 应用程序接口

1 引言

WYSIWYG(What You See Is What You Get，所见即所得)是一种软件技术规范，通常应用于编辑软件。该技术使得用户在视图中所看到文档与该文档的最终产品具有相同的样式，也允许用户在视图中直接编辑文本、图形、或文档中的其他元素。简单地说，WYSIWYG 技术使文档打印输出或 Web 浏览(对于 HTML 文档)的效果和屏幕视图编辑过程中的显示效果完全一致。

RichTextBox 是微软的 Visual Basic(以下简称 VB)所带的标准控件，该控件可用以加载、显示、编辑和保存 RTF 格式的文档，可以设置字体字形字号和颜色，插入图片和对象，但是会忽略诸如对象版式、行间距等高级设置。RichTextBox 控件具备基本的图文显示和编辑功能，完全可以以之为核心构成简单的图文编辑器。

在作者开发的 ePaper 题库软件中需要进行试题题目和答案的编辑录入，考虑到通用性，试题编辑器应具备插入图片和对象(主要是公式)的功能，所以使用了 RichTextBox 控件。为了让试题试卷的编辑显示，以及预览打印，具备 WYSIWYG 特性，作者着意对 RichTextBox 控件及其 WYSIWYG 实现作了一番研究，查阅了不少相关资料[1,2]，在前人的基础上，经过整理和完善，基本实现了基于 RichTextBox 控件的 WYSIWYG 显示、预览和打印，并将该技术成功地应用在 ePaper 题库软件中。现以所得之一二撰文，抛砖引玉，与各方大家探讨。在本文中，主要介绍基于 RichTextBox 控件的 WYSIWYG 显示。

2 具体实现

VB 的 RichTextBox 控件是 Win32 操作系统提供的 RichTextBox 的一个子类控件。操作系统的 RichTextBox 支持的许多消息并没有暴露在 VB 的 RichTextBox 控件中。其中一个就是 EM_SETTARGETDEVICE 消息。EM_SETTARGETDEVICE 消息用来告诉一个 RichTextBox 控件基于一个目标设备(诸如打印机)来显示其内容。在 VB 中使用该消息就有可能使一个 RichTextBox 控件支持 WYSIWYG 显示。

下面的 WYSIWYG_Display 函数完成设置 RichTextBox 控件使之以 WYSIWYG 方式显示其内容的功能。

```
'=========================================
'功能：WYSIWYG_Display，设置 RichTextBox 控件，使之在屏幕上显示的样式和在默认打印机上打印输出的结果一致，即 WYSIWYG(所见即所得)显示。
'参数：rtf，RichTextBox 类型，要设置成 WYSIWYG 显示的 RichTextBox 控件。
'参数：PageMargin，RECT 类型，页边距，包括 Left(左)，Top(上)，Right(右)，Bottom(下)各个页边距值，单位 mm(毫米)。
'参数：PageSize，POINTAPI 类型，纸张大小，包括 X(宽度)，Y(高度)值，单位 mm。
'返回值：Long 型，页面实际可打印区域的宽度，单位 twip。
'=========================================
Public Function WYSIWYG_Display(rtf As RichTextBox, PageMargin As RECT, PageSize As POINTAPI) As Long
    '可打印区域相对于左上角原点的水平和垂直偏移
    Dim LeftOffset As Long, TopOffset As Long
    '左右边距，从可打印区域的左边界衡量起
    Dim LeftMargin As Long, RightMargin As Long
    '上下边距，从可打印区域的上边界衡量起
    Dim TopMargin As Long, BottomMargin As Long
    '实际可打印宽度和高度，由纸张大小和页边距决定
    Dim PrintableWidth As Long, PrintableHeight As
```

```
Long
    '打印机的设备场景句柄
    Dim PrinterhDC As Long
    'SendMessage 函数返回值
    Dim r As Long
    '左、上、右、下页边距的宽/高度,单位 twip
    Dim LeftMarginWidth As Long, TopMarginHeight As Long
    Dim RightMarginWidth As Long, BottomMarginHeight As Long
    '设置打印机的坐标度量单位为 twip(缇),1inch(英寸)=1440Twips。
    Printer.ScaleMode = vbTwips
    '以参数 PageMargin 设置左、上、右、下页边距的宽/高度,以 twip 为单位。1cm(厘米)=567Twips,1mm(毫米)=56.7Twips。全局常量 mm=56.7
    LeftMarginWidth = PageMargin.Left * mm
    TopMarginHeight = PageMargin.Top * mm
    RightMarginWidth = PageMargin.Right * mm
    BottomMarginHeight = PageMargin.Bottom * mm
    '以参数 PageSize 设置打印机的纸张大小(宽度和高度),以 twip 为单位。
    Printer.Width = PageSize.X * mm
    Printer.Height = PageSize.Y * mm
    '得到页面上可打印区域相对于左上角原点的水平和垂直偏移,并转换为单位 twip。
    LeftOffset = GetDeviceCaps (Printer.hDC, PHYSICALOFFSETX)
    LeftOffset = Printer.ScaleX(LeftOffset, vbPixels, vbTwips)
    TopOffset = GetDeviceCaps (Printer.hDC, PHYSICALOFFSETY)
    TopOffset = Printer.ScaleX(TopOffset, vbPixels, vbTwips)
    '计算左右边距,从可打印区域的左边界衡量起。
    LeftMargin = LeftMarginWidth - LeftOffset
    RightMargin = (Printer.Width - RightMarginWidth) - LeftOffset
    '计算上下边距,从可打印区域的上边界衡量起。
    TopMargin = TopMarginHeight - TopOffset
    BottomMargin = (Printer.Height - BottomMarginHeight) - TopOffset
    '计算实际可打印宽度
    PrintableWidth = RightMargin - LeftMargin
    '计算实际可打印高度
    PrintableHeight = BottomMargin - TopMargin
    '创建打印机的设备场景(上下文),并获得其句柄。
    '该设备场景需要保持以用于 RichTextBox 控件的 WYSIWYG 显示。
    PrinterhDC = CreateDC(Printer.DriverName, Printer.DeviceName, 0, 0)
    '发送消息,告之 RichTextBox 控件基于打印机来显示其内容。
    r = SendMessage(rtf.hWnd, EM_SETTARGETDEVICE, PrinterhDC, ByVal PrintableWidth)
    '返回实际可打印区域的宽度
    WYSIWYG_Display = PrintableWidth
End Function
```

WYSIWYG_Display 函数中与显示和打印相关的部分参数和变量的含义如图 1 所示。

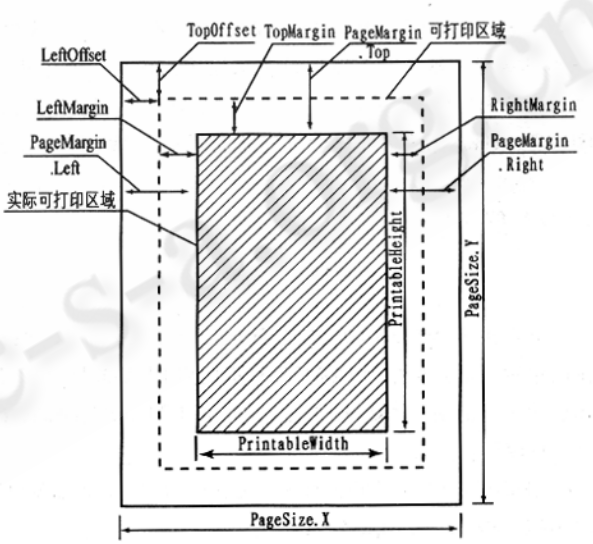

图 1　函数中部分参数的示意图

3　相关 API

在 WYSIWYG_Display 函数中使用了三个 Windows API 函数,下面分别作一简单介绍。

CreateDC 函数为专门设备创建设备场景(上下文)。在 Windows 操作系统中,需要在图形输出设备(例如屏幕或打印机)上绘制图形时,必须首先获得设备上下文的句柄。先给出这个句柄,Windows 才允许程序使用设备,在 GDI(Graphics Device Interface,图形设备接口)函数中将句

柄作为一个参数传入,向 Windows 标明需要使用的设备。设备上下文实际上是一个数据结构,它包括设备的参数及各种图形、文字属性的现行设定值,它们对后面的图形、文字输出命令起控制作用。WYSIWYG_Display 函数调用 CreateDC 函数创建默认打印机的设备上下文,并获得其句柄。

一个设备上下文通常涉及物理设备,如显示器、打印机等,所以需要获取有关该设备的信息,如分辨率大小和彩色深度等。可以通过调用 GetDeviceCaps 函数来获取这样的信息。打印机的可打印区域与打印纸边界一般有一定的边距(偏移),该边距是打印机自身造成的,因此称之为物理边距,并且这些物理边距在不同大小的纸张中是不一样的,因此首先要获取这些数值。WYSIWYG_Display 函数调用 GetDeviceCaps 函数来获得打印机的可打印区域的水平/垂直偏移。

另外一个用到的 API 函数是 SendMessage。顾名思义,SendMessage 函数的功能是"发送消息",即将一条消息发送到指定对象(操作系统、窗口或控件等)上,以产生特定的动作(如滚屏、改变对象外观等),功能非常强大。本文中的 WYSIWYG_Display 函数就是通过调用 SendMessag 函数发送 EM_SETTARGETDEVICE 消息来告诉一个 RichTextBox 控件基于一个目标设备来显示其内容。通过指定目标设备为打印机,并以实际可打印区域的宽度来格式化 RichTextBox 控件的显示宽度,从而实现了 WYSIWYG 显示。

4 结束语

关于上述三个 API 函数更为详尽的说明请参考相关资料[3]。本文所述之基于 RichTextBox 控件的 WYSIWYG 显示的的技术已成功应用到本人所开发的 ePaper 题库软件,感兴趣的读者可以从 http://liuchao.nease.net/works/epaper/ePaperTrialV078_SetupTyp.exe 下载该软件。基于 RichTextBox 控件的 WYSIWYG 预览和打印技术也在该软件中实现了,本人随后将一并撰文介绍。

参考文献

1 MSDN Communities. Printing from a RichTextBox [EB/OL]。http://www.developerfusion.com/show/244/。Microsoft Knowledge Base Article - 146022。

2 Microsoft Knowledge Base Article - 146022, HOWTO: Set Up the RichTextBox Control for WYSIWYG Printing[EB/OL]。http://support.microsoft.com/default.aspx?scid=kb;en-us;146022。

3 朱友芹,新编 Windows API 参考大全[M],电子工业出版社,2000 年。