

基于 RichTextBox 控件的 WYSIWYG 预览

WYSIWYG Preview based on RichTextBox Control

黄建华 (桂林电子工业学院通信与信息工程系 541004)

刘超 (上海交通大学电子工程系 200030)

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,调用 Windows API 及其未公开的信息,使其能够按打印输出的结果来预览其内容,即如何实现 RichTextBox 控件的 WYSIWYG 预览。

关键词: RichTextBox 控件 所见即所得 预览 应用程序接口

1 引言

现今成熟规范的应用软件,无论其应用于何种领域,只要具备打印功能,就具有相应的打印预览功能,并且这种打印预览是 WYSIWYG(What You See Is What You Get,所见即所得)的。WYSIWYG 已经成为一种软件技术规范。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。WYSIWYG 预览使用户可以在文档打印输出之前就在屏幕上看到(预览)与打印输出完全一致的效果,从而决定是付诸打印,还是进一步修改调整文档。

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。尽管它会忽略诸如对象版式、行间距等高级设置,但已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,试题编辑器应能插入图片和对象(主要是公式),所以使用了 RichTextBox 控件。为了让试卷的预览具备 WYSIWYG 特性,作者着意对 RichTextBox 控件及其 WYSIWYG 预览作了一番研究,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 预览,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。

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_FORMATRANGE 消息。EM_FORMATRANGE 消息以指定的坐标,把 RichTextBox 控件中的内容,向输出设备一次发送一页。在 VB 中使用该消息就使得令 RichTextBox 控件支持 WYSIWYG 预览成为可能。

下面的 WYSIWYG_Preview 示例函数根据页边距和纸张大小等参数,在预览窗体的图片框控件组中对 RichTextBox 控件的内容以 WYSIWYG 方式进行分页预览。

功能: WYSIWYG_Preview, 根据提供的纸张大小和页边距,在预览窗体的图片框控件组中对 RichTextBox 控件的内容以 WYSIWYG 方式进行分页预览。

参数: rtf, RichTextBox 类型,应设为需要进行 WYSIWYG 预览的 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。

```

' = = = = =
Public Sub WYSIWYG_Preview( rtf As RichTextBox, Page-
Margin As RECT, PageSize As POINTAPI)
' 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
' 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
' FORMATRANGE 结构包含着 RichTextBox 控件为特定
设备格式化其输出所必需的信息
Dim fr As FORMATRANGE
' 页面上要打印的矩形范围(相对于可打印区域)
Dim rcDrawTo As RECT
' 页面上可打印区域的矩形范围
Dim rcPage As RECT
' RichTextBox 中内容的长度(以字符数衡量)
Dim TextLength As Long
' 下一次打印起始处的字符的索引
Dim NextCharPosition As Long
' 记录上一次打印起始处的字符的索引
Dim LastCharPosition As Long
' SendMessage 函数返回值
Dim r As Long
' 当前页码
Dim iPage As Integer
' 页码字符串
Dim strPage As String
' 循环变量
Dim i As Integer
' 错误捕获
On Error GoTo ErrHandle
' 设置打印机的坐标度量单位为 twip(缇), linch(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, PHYSICAL-
OFFSETX)
LeftOffset = Printer.ScaleX(LeftOffset, vbPixels,
vbTwips)
TopOffset = GetDeviceCaps(Printer.hDC, PHYSICAL-
OFFSETY)
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 - BottomMargin-
Height) - TopOffset
' 设定页面上可打印的矩形区域的范围。
rcPage.Left = 0
rcPage.Top = 0
rcPage.Right = Printer.ScaleWidth
rcPage.Bottom = Printer.ScaleHeight
' 设定页面上要打印的矩形区域的范围(相对可打印
区域)。
rcDrawTo.Left = LeftMargin
rcDrawTo.Top = TopMargin

```

```

rcDrawTo.Right = RightMargin
rcDrawTo.Bottom = BottomMargin
`根据打印机纸张大小调整预览窗体中用于打印预览
的图片框控件的大小。
frmPreview.SizePreview Printer.Width, Printer.Height
`设置打印信息(FORMATRANGE 结构)
`用以绘制的实际 DC
fr.hDC = frmPreview.picPreview(0).hDC
`用来决定文本格式的目标 DC
fr.hdcTarget = frmPreview.picPreview(0).hDC
`指定页面上要打印的矩形区域
fr.rc = rcDrawTo
`指定页面上可打印的矩形区域
fr.rcPage = rcPage
`指定打印的起始位置,0 从头开始
fr.chrg.cpMin = 0
`指定打印的终止位置,-1 到末尾
fr.chrg.cpMax = -1
`获取 RichTextBox 控件中内容的长度(以字符数衡
量)。
`如下的简单语句出现在微软知识库 Q146022 中,但
它并不能得到正确的长度值。
TextLength = Len(rtf.Text)
`本人使用如下方法,可获得正确的长度值。
TextLength = SendMessage(rtf.hWnd, WM_GET-
TEXTLENGTH, CLng(0), CLng(0))
`当前页码 = 1
iPage = 1
`循环打印每一页直到结束
Do
`以预览窗体为缺省对象
With frmPreview
`如果当前页码 > 1
If iPage > 1 Then
`为预览窗体中的预览页码下拉列表框增加一
项
.AddPage iPage
`以新的预览图片框设定 DC
fr.hDC = .picPreview(iPage - 1).hDC
fr.hdcTarget = .picPreview(iPage - 1).hDC

```

```

End If
End With
`发送 EM_FORMATRANGE 消息以打印某页,其返
回值为本次打印的最后一个字符的索引 + 1
NextCharPosition = SendMessage(rtf.hWnd, EM-
_FORMATRANGE, True, fr)
`问题:某些情况下某些打印机会在最后一页陷入
死循环
If NextCharPosition <= LastCharPosition Then
If NextCharPosition < LastCharPosition Then
frmPreview.RemovePage iPage
iPage = iPage - 1
End If
NextCharPosition = TextLength
End If
LastCharPosition = NextCharPosition
`如果下一个打印字符的索引已经超出 RTF 的内
容长度,则打印完毕,退出
If NextCharPosition >= TextLength Then Exit Do
`否则,打印的起始位置定位于上次打印的返回值
(即上次打印的最后一个字符的索引 + 1)
fr.chrg.cpMin = NextCharPosition ` Starting posi-
tion for next page
`页码 + 1
iPage = iPage + 1
Loop
`释放预览 rtf 时所占用的内存
r = SendMessage(rtf.hWnd, EM_FORMATRANGE,
False, ByVal CLng(0))
`显示预览窗体
frmPreview.Show vbModal
Exit Sub
`错误处理
ErrHandle:
Select Case err.Number
Case 482
MsgBox "未安装打印机或打印机属性错!"
Exit Sub
Case Else
MsgBox err.Number & " " & err.Description

```

Resume Next

End Select

End Sub

WYSIWYG_Preview 函数中与显示相关的部分参数和变量的含义可参见本人拙作《基于 RichTextBox 控件的 WYSIWYG 显示》一文^[3]。

3 相关 API

在 WYSIWYG_Preview 示例函数中使用了两个 Windows API 函数,下面分别作一简单介绍。

Windows 中使用了设备上下文 (Device Context, DC) 的概念。一个设备上下文通常涉及物理设备,如显示器、打印机等。有时候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 函数来获取此类的信息^[4]。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。示例通过调用 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。示例就是通过调用 SendMessag 发送 EM_FORMATRANGE 把一个 RichTextBox 控件的内容一页一页地发送到输出设备。通过指定输出设备为图片框控件,并根据页边距和纸张大小等参数做相应地设置,从而实现了 WYSIWYG 预览。关于 EM_FORMA-

TRANGE 消息及与其密切相关的 FORMATRANGE 结构的详尽说明可参考微软 MSDN。

另外,示例中还使用 SendMessage 函数来发送 WM_GETTEXTLENGTH 消息。发送该消息是为了获得与某个窗口 (这里是 RichTextBox 控件) 相联系的文本的以字符计的 length。这一点对于程序正确执行很重要,因为像微软知识库 Q146022^[2] 那样简单地使用 Len 函数并不总是能返回正确的结果,尤其是当 RichTextBox 控件中插入了图片和对象的时候。

4 结束语

本文所述之基于 RichTextBox 控件的 WYSIWYG 预览技术已成功应用到作者所开发的 ePaper 题库软件,感兴趣的读者可以从作者的主页 <http://liuchao.nease.net/> 上下载该软件考察试用。

参考文献

- 1 MSDN Communities. Printing from a RichTextBox [EB/OL]. <http://www.developerfusion.com/show/244/>.
- 2 Microsoft Knowledge Base Article - 146022, HOW-TO: Set Up the RichTextBox Control for WYSIWYG Printing [EB/OL]. <http://support.microsoft.com/default.aspx?scid=kb;en-us;146022>.
- 3 刘超, 基于 RichTextBox 控件的 WYSIWYG 显示, 计算机系统应用, 2004, 7.
- 4 朱友芹, 新编 Windows API 参考大全 [M], 电子工业出版社, 2000 年.