



# TMS to the rescue!

## Merging with AdvRichEdit

By Joe C. Hecht

A customer needed an application that could insert a “blog like” post into the middle of a corporate HTML page, making a sandwich of sorts between a header and footer template. Further, they needed the application to provide complete text formatting and image insertion. The app needed to be WYSIWYG (what you see is what you get), targeted to users with no experience at writing web code.

This application had to be very easy to use, quick to write, and I needed to deliver results – *fast!*

Writing a real time HTML editor and viewer is no walk in the park. I have a lot of experience at writing RTF editors and converting the results to PDF, and have enough experience to know that the Microsoft Windows® RTF control has exactly what I need to get the job done. Hooking up a Microsoft Windows® RichEdit to a pile of controls to set attributes will easily require a thousand lines of code, and even more code will be needed for conversion to HTML. Even with all the RTF related code I have archived back over the years, there was no way I could get the job out quickly using the Microsoft Windows® RichEdit control.

I purchased the TMS Component Pack from [tmssoftware.com](http://tmssoftware.com). The TMS Component Pack contains a massive collection of over 400 different VCL controls. Included is the TAdvRichEdit control, along with a wide range of associated controls used to enhance the job of RTF editing.

Using a TAdvRichEdit with one of the RTF Editing Toolbars, within a few minutes, I not only had a gorgeous interface to show off, I was able to easily output the RTF’s results as both a HTML and a PDF stream. So far so good!

TMS Software has a wide variety of controls specifically designed to make short work of setting the attributes of the text and other elements available in a TAdvRichEdit. TMS Software has tool bars, ribbon bars, popup menus, a host of “Office” like controls, and even a spell checker! The available attribute settings were amazing, including the ability to define lists, list types, add special characters and images, adjusting alignments and fonts, along with setting all the available text attributes, including subscript, superscript, and custom indents. They even have controls to support emoticons!

Leveraging the TMS Component Pack, I was able to quickly deliver several working mock-ups of the app, each with a different look and feel, and impress the customer with both an amazingly fast turnaround and lots of great looking choices! Did I mention how “*quickly*”? *In minutes!*

The customer loved it! They handed out the apps to the users to try out, asked the users to take notes, and then called back with a very short “wish list” – and a contract! *Cha-Ching! Sale!!*

The customer chose the TAdvRichEditorFormatToolBar for setting the RTF’s attributes, and the matching TAdvRichEditorEditToolBar for loading, saving, cut, copy, paste, undo, and re-do. Perfect!

Now for the fun part! The wish list. It seemed simple enough, but knowing the internals of RTF editing, I have the experience to know that often enough, simple requests may not always be so easy.

## The “wish list”

Wish lists usually border on fantasies. TMS Software did a great job, since all the major up front work was done.

This list was short, sweet and realistic.

- Add a “Color Button” to show the overall RTF’s background color.
- Show only the “web like” fonts “Arial”, “Courier new”, and “New Times Roman”.
- Override the “Load” and “Save” buttons to customize how they work.

*Easy enough, right?*

## Plan of attack

Every wish list item requires access to the TMS toolbar child controls. A trap will be set in during the FormCreate() method that will obtain references to the needed TMS toolbar child controls at run time, remove the “non-web” fonts from the font combo box, and dynamically override the events for “Load” and “Save” to provide the requested changes.

### **Item: Add a “Color Button” to set and show the overall background color**

Adding an extra button to the toolbar was super easy. Right clicking on the toolbar produced a long list of available controls to add. The TAdvOfficeColorSelector was the perfect choice for consistency, as it matches the other controls used by the TAdvRichEditorFormatToolBar to set color properties and also contains all the functionality required to display a drop down color selector. The “SelectColor()” event of the new button will be used to apply the background color to the RTF control.

### **Clicking the background color button**

The background color of the RTF control is set via its “Color” property. The color is applied to the overall background when *nothing is selected*, *SelectStart is -1*, and *SelLength is zero*. When *something is selected*, or *SelectStart is not -1*, the color property will set either the background color at the caret, or the item that is selected.

To set background color, the “SelectColor()” event of the new button must:

- Save the caret and current selection (if any) in the RTF control.
- Make sure nothing is selected in the RTF control.
- Set the RTF’s SelectStart to -1 and SelectLength to 0.
- Set the background color.
- Restore the caret and selection (if any).

Finally, the AdvRichEditor's DoSelectionChanged() method is called to synchronize the RTF control and the associated tool bar controls with the editing changes made by the new background color button.

**Item: Show only the “web like” fonts “Arial”, “Courier new”, and “New Times Roman”**

No objects or pointers are stored in the TAdvOfficeFontSelector items.objects array, so the list items for unneeded fonts are safely removed from the TAdvOfficeFontSelector (in reverse order).

**Item: Override the “Load” and “Save” buttons to customize how they work**

The OnClick() events for these buttons will be totally replaced, allow for custom processing.

## Setting the trap

In the FormCreate() method, **FindRTFEditControls()** is called, where the child controls of the “TAdvRichEditorFormatToolBar” and “TAdvRichEditorEditToolBar” toolbars are identified, and saved back (as required). If the needed controls are not found, **FindRTFEditControls()** returns false, an error message is displayed, and the application is terminated.

A call is then made to **TrapRTFEditControls()** to actually set the traps, where the events for the “Load” and “Save” buttons are replaced, and the unneeded “non-web fonts” are safely removed from the TAdvOfficeFontSelector.

**OpenFile** – Normally, a dialog box is displayed, allowing the user to load the RTF from a number of different file formats. Per the customer's requirements, the RTF is simply re-loaded from the same file every time (if it exists), otherwise default settings are used and no further processing is required.

The title for the web page is conveniently stored in the “Author” property of the RTF file, where it is extracted from the RTF file and used to set the “Text” property of the “WebpageTitle” edit control.

The background color of the RTF control is obtained by selecting “nothing” and retrieving the value of the “Color” property of the RTF control, then using the value retrieved to set the “SelectedColor” property of the AdvOfficeBkColorSelector control.

Finally, the caret is positioned in front of the first character in the RTF control (with no selection), and the AdvRichEditor's DoSelectionChanged() method is called to synchronize the RTF control and the associated tool bar controls (containing the new background color button).

**SaveFile** – Normally, a dialog box is displayed, allowing the user to save the RTF in a number of formats. Per the customer's requirements, the RTF's contents are saved as a “.rtf” file always to the same name (for later recall), and the RTF's contents are also saved as an HTML string to be merged with a corporate header and footer HTML template, producing a completed web page. The web page

title is replaced in the “corporate header” with the contents of the “WebpageTitle” edit control, and the “Title” is saved as the “Author” property in the “.rtf” file (for later recall when the .rtf file is reloaded).

A “TAdvRichEditorMiniHTMLIO” component is used to convert the RTF’s contents to HTML. Setting the “PlainHTML” property causes no output of an HTML or body tag. An HTML DIV tag is used to surround the contents, using an inline style with the default font, size, and background color. Using a DIV tag allows additional style properties to be added (for example, additional margins and padding). Images used in the RTF are set to encode as “HTML inline images”, requiring no external images to be gathered for publishing. The end result is a single, self-contained web page that is ready to upload.

## **TAdvRichEditorProtectedAccess**

TMS Software does a great job at making RTF easy. TMS Software wrote their own RTF controls, and have versions available for almost every platform on the planet! They have RTF controls for Delphi<sup>®</sup>, C++Builder<sup>®</sup>, and Lazarus, and available on the Linux<sup>®</sup>, OS X<sup>®</sup>, and Microsoft Windows<sup>®</sup> platforms. At the same time, TMS Software was very careful to implement a UI interface that closely follows the Microsoft Windows<sup>®</sup> RTF control (for UI consistency).

TMS Software uses brilliantly simple interfaces to wrap very complex controls. You can’t have everything. A simple interface sometimes means you may have to roll up your sleeves to get to a nitty-gritty internal.

In working with the internals of the TAdvRichEdit and associated controls, I found many of the protected methods and properties to be very useful. You may be thinking the same thing!

Adding custom functionality to a control often requires the need to write (and install) a descending class to access the protected methods and properties, however, simply declaring a descending class and casting an object reference works just as well (if nothing is added). This concept is known as a *THack()* class.

If I recall correctly, *THack()* was originally conceived in the early days of Delphi at Borland, by two bright young engineers I had the honor of working with: Andrew Kern and David Powell.

This concept is demonstrated by the use of the declared “TAdvRichEditorProtectedAccess” class, and is used to access the AdvRichEditor’s protected DoSelectionChanged() method.

Know that protected methods are not intended for use by the application. They are intended for component writers, and are internal to the object. You must be extremely careful in using them. Wherever possible, use public and published methods and properties as protected methods and properties are usually not documented, and the implementation (and even the existence) of these internals are subject to change at any time (and without warning).

## Taking it further

The example presented demonstrates a great starting place for both the use and extension of only a few of the large number TMS Software's RTF based controls available in the TMS Component Pack. Additional possibilities for use and enhancements of these controls are seemingly unlimited. In fact, I have a very excellent real time aware RTF edit popup menu that I plan to interface. In addition to the TMS Software RTF based controls, many other TMS controls come to mind that can be immediately (and easily) leveraged (such as the large number of TMS HTML based controls). TMS Software also provides for importing and exporting to and from many additional formats (such as PDF), support for many different image and vector formats, and also has database aware versions for most TMS controls.

## Code Notes

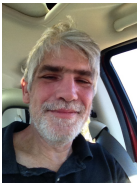
HTML header and footer files used in the published example contain our own web code [plug].

The style used is typical of a Borland Pascal example, with function separator "comment" lines added that are similar to what is used by C++Builder® and what is displayed by default in the Lazarus IDE.

Three exceptions to the "Borland style" are used:

- Code is always blocked using the rule "Never a then without a begin".
- Parentheses enclose every mathematical, logical, and bit-wise operation (for absolute clarity).
- Function names are assigned rather than using the Delphi® automatic function "result" variable (allowing code to easily port to other compilers using a "native" code compile mode).

Most of my coding style comes from many years of experience, and a good read of Steve McConnell's widely acclaimed books '*Code Complete*' and '*Code Complete 2*' ISBN 978-0-7356-1967-8 (2<sup>nd</sup> ed.)



## Author's Lebenslauf

Joe C. Hecht is a software engineer that specializes in graphics engines, print drivers, PDF, SDK design, C++, and Delphi®.

Joe hails from Borland's Delphi® group, where he earned the moniker "TJoe".

His works include several award winning desktop publishing software titles including 'TypeStudio', 'Effects Specialist', and 'OptiScript'.

Joe was a featured writer for the 'Windows Tech Journal', and went on to author many technical white papers during his time at Borland. He is acknowledged as a contributing author in many books that have been published on the subjects of computer graphics, programming, and Delphi®.

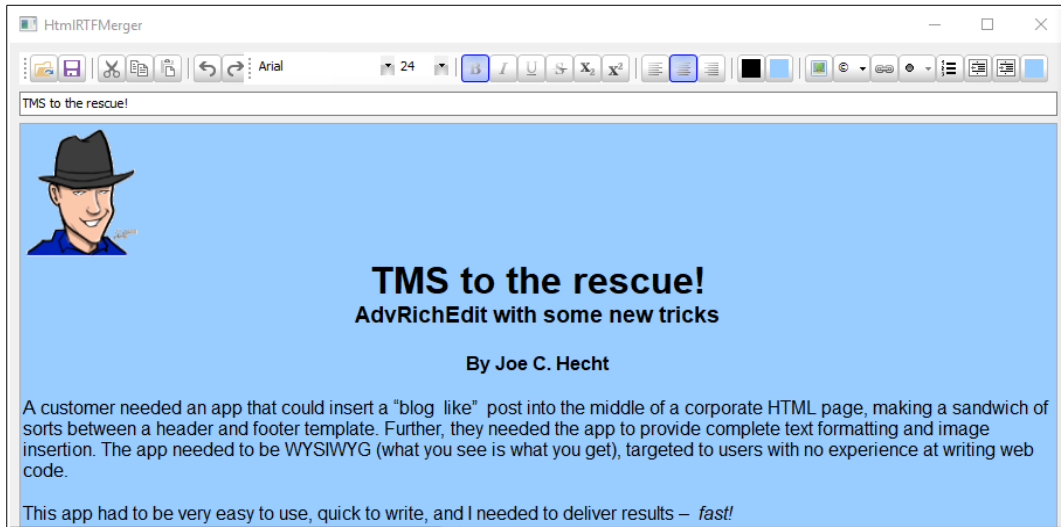
Joe participated in the 3D gaming engines used by Disney Interactive and 7<sup>th</sup> Level to produce many best-selling titles including Disney's 'The Hunchback of Notre Dame' and 'Timon & Pumbaa's Jungle Games', along with 'Monty Python & the Quest for the Holy Grail', 'Monty Python's Complete Waste of Time', 'Ace Ventura', 'Battle Beast', and Howie Mandel's 'Adventures in Tuneland'.

Joe is the lead engineer for Code4Sale LLC and a long time Embarcadero Technology Partner. His current projects include the open source ÜberPDF™ SDK, the GX Graphics™ high speed render engine, and cross platform development for Microsoft Windows®, OS X®, Linux®, Mobile, and Embedded.

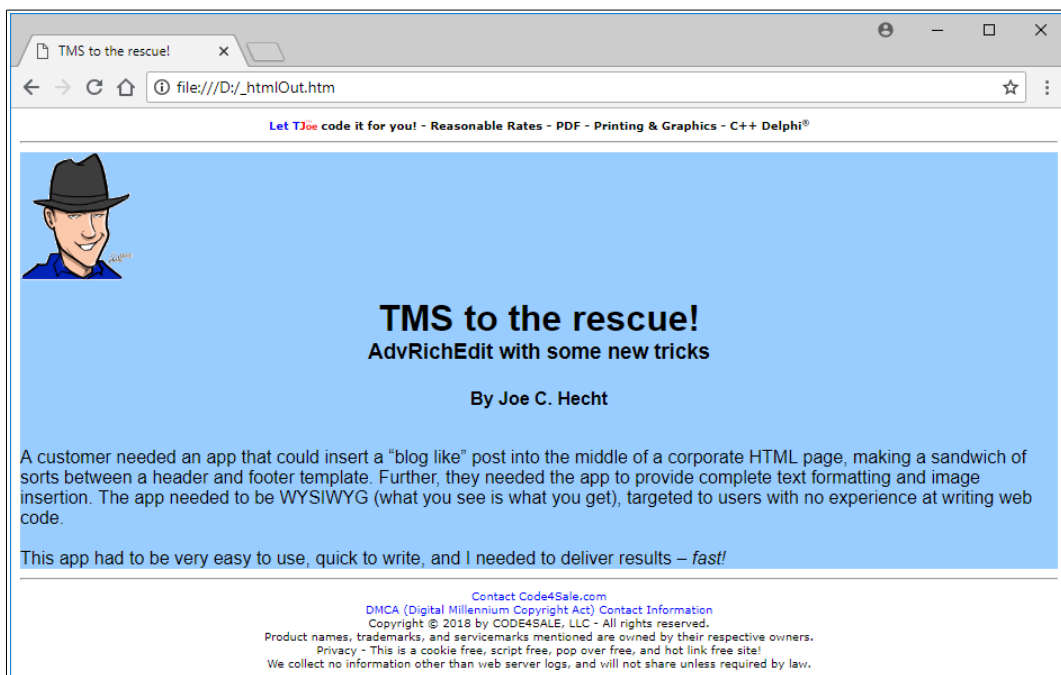
When not coding, Joe can be found sailing the clear waters of Florida's Emerald Coast aboard his sloop *El deseo de padre*.

Joe is actively soliciting quality code and other works in need of authorship and can be contacted at: <https://code4sale.com/email/JoeHecht/>

# The application



# The HTML result (With header and footer templates)



# The Code

## \_htmlBegin.txt

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
<HEAD>
<title>TMS AdvRichEdit to the rescue!</title>
<META NAME="description" CONTENT="Code4Sale - Your place to buy and sell code!">
<META NAME="keywords" CONTENT="Code4Sale, CodeForSale, Code For Sale, Delphi, CBuilder, C+
+Builder, TExcellentImagePrinter, TExcellentFormPrinter, Joe C. Hecht, Joe Hecht, TJoe,
printing, blank pages, StretchDIBits, Form print, Form.Print, GetFormImage, VCL forms, Image
print, VCL Image">
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=UTF-8">
<META HTTP-EQUIV="no-cache" CONTENT="no-cache">
<META HTTP-EQUIV="Expires" CONTENT="0">
<META HTTP-EQUIV="Pragma" CONTENT="no-cache">
<META HTTP-EQUIV="Cache-Control" CONTENT="no-cache">
<STYLE TYPE="text/css">
a:active {
    color: blue;
    text-decoration: none;
}
a:hover {
    color: blue;
    text-decoration: none;
}
a:link {
    color: blue;
    text-decoration: none;
}
a:visited {
    color: blue;
    text-decoration: none;
}
</STYLE>
</HEAD>
<BODY>
<CENTER>
<FONT SIZE="-2">
<FONT style="font-family: Verdana, Arial, Helvetica, sans-serif;">
<A HREF="https://code4sale.com/email/JoeHecht/">
<B>
Let
<span style="color: blue; letter-spacing:-0.1em;">T</span><span style="color:red; margin-
left:0.05em; letter-spacing:-0.075em;">J<sup style="font-weight:normal; font-size: 0.55em;
margin-left:0.1em;">&trade;</sup></span><span style="color:red; font-size: 0.75em; margin-
left:-0.65em;">oe</span>
<FONT COLOR="#000000">
code it for you! - Reasonable Rates - PDF - Printing & Graphics - C++
Delphi<SUP><SMALL>&#x00AE;</SMALL></SUP>
</FONT>
</B>
</A>
</FONT>
</FONT>
</CENTER>
<HR>
```



## htmlEnd.txt

```
<HR>
<CENTER>
<FONT SIZE="-2">
<FONT style="font-family: Verdana, Arial, Helvetica, sans-serif;">
<A HREF="https://code4sale.com/email/">Contact Code4Sale.com</A><BR>
<A HREF="https://code4sale.com/copyright/"><FONT COLOR="#0000FF">DMCA (Digital Millennium
Copyright Act) Contact Information</FONT></A><BR>
Copyright © 2018 by CODE4SALE, LLC - All rights reserved.<BR>
Product names, trademarks, and servicemarks mentioned are owned by their respective
owners.<BR>
Privacy - This is a cookie free, script free, pop over free, and hot link free site!<BR>
We collect no information other than web server logs, and will not share unless required by
law.<BR>
</FONT>
</FONT>
</CENTER>
</BODY>
</HTML>
```

## HtmlRTFMerger.dpr

```
program HtmlRTFMerger;

uses
  Vcl.Forms,
  HtmlRTFMergerMainUnit in 'HtmlRTFMergerMainUnit.pas' {Form1};

{$R *.res}

begin
  Application.Initialize;
  Application.MainFormOnTaskbar := True;
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.
```

## HtmlRTFMergerMainUnit.pas

```
unit HtmlRTFMergerMainUnit;
//-----

//-----
interface
//-----

//-----
uses
//-----
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
  System.Classes, Vcl.Graphics, Vcl.Controls, Vcl.Forms, Vcl.Dialogs,
  Vcl.StdCtrls, Vcl.ExtCtrls,
  //-----
  ShellApi, // Needed to launch the system web browser
  AdvOfficeComboBox, // Needed to access TMS toolbar comboboxes
  AdvRichEditorMiniHTMLIO, // Needed for specific HTML output
  //-----
  AdvToolBar,
  AdvToolBarExt,
  AdvRichEditorToolBar,
  AdvRichEditorBase,
  AdvRichEditor,
  AdvScrollControl,
  AdvOfficeSelectors,
  AdvGlowButton;
//-----

//-----
// Used to access the protected methods of TAdvRichEditor.
//-----
type TAdvRichEditorProtectedAccess = class(TAdvRichEditor);
//-----

//-----
type
//-----
  TForm1 = class(TForm)
    EditWebpageTitle: TEdit;
    AdvRichEditor: TAdvRichEditor;
    AdvRichEditorEditToolBar: TAdvRichEditorEditToolBar;
    AdvRichEditorFormatToolBar: TAdvRichEditorFormatToolBar;
    AdvOfficeBkColorSelector: TAdvOfficeColorSelector;
    procedure FormCreate(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure AdvOfficeBkColorSelectorSelectColor(Sender: TObject;
      AColor: TColor);
  protected
    RTFDefaultFontName: string;
    RTFDefaultFontSize: integer;
    TMSOpenFileButton: TAdvCustomGlowButton;
    TMSSaveFileButton: TAdvCustomGlowButton;
    TMSFontListComboBox: TAdvOfficeFontSelector;
    TMSFontSizeComboBox: TAdvOfficeFontSizeSelector;
    function FindRTFEditContols() : boolean;
```

```

    procedure TrapRTFEditContols();
    procedure NewTMSOpenFileButtonOnClick(Sender: TObject);
    procedure NewTMSSaveFileButtonOnClick(Sender: TObject);
public
end;
//-----

//-----
var
    Form1: TForm1;
//-----

//-----
implementation
//-----
{$R *.dfm}
//-----

//-----
procedure TForm1.FormCreate(Sender: TObject);
//-----
begin
    // Set the default font and size for AdvRichEditor.
    RTFDefaultFontName := 'Arial';
    RTFDefaultFontSize := 12;
    AdvRichEditor.Font.Name := RTFDefaultFontName;
    AdvRichEditor.Font.Size := RTFDefaultFontSize;
    // Find the RTFEditContols
    if (not(FindRTFEditContols())) then begin
        ShowMessage('RTFEditContols not found - terminating');
        Application.Terminate();
    end;
    // Trap the RTFEditContols
    TrapRTFEditContols();
    // Open the RTF file (if it exists)
    NewTMSOpenFileButtonOnClick(self);
end;
//-----

//-----
procedure TForm1.FormShow(Sender: TObject);
//-----
begin
    AdvRichEditor.SetFocus();
    AdvRichEditor.SelectText(0, 0);
end;
//-----

//-----
function TForm1.FindRTFEditContols() : boolean;
//-----
var
    idx: integer;
    aControl: TControl;
begin
    // Get a reference to the open and save buttons in AdvRichEditorEditToolBar

```

```

TMSOpenFileButton := AdvRichEditorEditToolBar.GetButton(integer(btFileOpen));
TMSSaveFileButton := AdvRichEditorEditToolBar.GetButton(integer(btFileSave));
// Get a reference to the font ComboBoxes in AdvRichEditorEditToolBar
for idx := 0 to (AdvRichEditorFormatToolBar.ControlCount - 1) do begin
  if ((Assigned(TMSFontListComboBox)) and
      (Assigned(TMSFontSizeComboBox))) then begin
    break;
  end;
  aControl := AdvRichEditorFormatToolBar.Controls[idx];
  if (aControl is TAdvOfficeFontSelector) then begin
    TMSFontListComboBox := TAdvOfficeFontSelector(aControl);
    continue;
  end;
  if (aControl is TAdvOfficeFontSizeSelector) then begin
    TMSFontSizeComboBox := TAdvOfficeFontSizeSelector(aControl);
    continue;
  end;
end;
// Make sure all references are valid
if (not((Assigned(TMSOpenFileButton)) and
        (Assigned(TMSSaveFileButton)) and
        (Assigned(TMSFontListComboBox)) and
        (Assigned(TMSFontSizeComboBox)))) then begin
  FindRTFEditContols := false;
  exit;
end;
FindRTFEditContols := true;
end;
//-----

//-----
procedure TForm1.TrapRTFEditContols();
//-----
var
  idx: integer;
  fontName : string;
begin
  // Set the new OnClick events for the OpenFile and SaveFile buttons
  TMSOpenFileButton.OnClick := NewTMSOpenFileButtonOnClick;
  TMSSaveFileButton.OnClick := NewTMSSaveFileButtonOnClick;
  // Safely remove all non web compatible fonts from the FontListComboBox
  for idx := (TMSFontListComboBox.Items.Count - 1) downto 0 do begin
    fontName := TMSFontListComboBox.Items[idx];
    if ((fontName <> 'Arial') and
        (fontName <> 'Courier New') and
        (fontName <> 'Times New Roman') and
        (not (Assigned(TMSFontListComboBox.Items.Objects[idx]))) then begin
      TMSFontListComboBox.Items.Delete(idx);
    end;
  end;
end;
//-----

//-----
procedure TForm1.AdvOfficeBkColorSelectorSelectColor(Sender: TObject;
                                                    AColor: TColor);
//-----
var
  selection : TSelection;
  caret : TCaret;
begin
  AdvRichEditor.SetFocus();

```

```

// Save the selection (if any)
selection := TSelection.Create();
selection.Assign(AdvRichEditor.Selection);
// Save the caret
caret := TCaret.Create();
caret.Assign(AdvRichEditor.Caret);
// Select nothing and set SelStart to -1 to set the RTF's background color
AdvRichEditor.UnSelect();
AdvRichEditor.SelStart := -1;
AdvRichEditor.SelLength := 0;
// Set RTF's background color
AdvRichEditor.Color := AColor;
// Restore the selection (if any)
AdvRichEditor.Selection.Assign(selection);
selection.Free();
// Restore the caret
AdvRichEditor.Caret.Assign(caret);
caret.Free();
// Update AdvRichEditor and the toolbar
TAdvRichEditorProtectedAccess(AdvRichEditor).DoSelectionChanged();
end;
//-----

//-----
procedure TForm1.NewTMSOpenFileButtonOnClick(Sender: TObject);
//-----
begin
  if (not(FileExists('__AdvRichEditContents.rtf'))) then begin
    exit;
  end;
  AdvRichEditor.LoadFromFile('__AdvRichEditContents.rtf');
  // The web-page title is stored in the AdvRichEditor.Author property
  EditWebpageTitle.Text := Trim(AdvRichEditor.Author);
  // Select nothing and set SelStart to -1 to get the RTF's background color
  AdvRichEditor.UnSelect();
  AdvRichEditor.SelStart := -1;
  AdvRichEditor.SelLength := 0;
  AdvOfficeBkColorSelector.SelectedColor := AdvRichEditor.Color;
  // Set the caret to the first text char (if any - with no selction)
  AdvRichEditor.SelectText(0, 0);
  // Update AdvRichEditor and the toolbar
  TAdvRichEditorProtectedAccess(AdvRichEditor).DoSelectionChanged();
end;
//-----

//-----
procedure TForm1.NewTMSSaveFileButtonOnClick(Sender: TObject);
//-----
var
  htmlSL: TStringList;
  htmlOutText: string;
  uppercaseTestString: string;
  posStart: integer;
  posEnd: integer;
  advRichEditorMiniHTMLIO: TadvRichEditorMiniHTMLIO;
  rtfContentsAsHtml: string;
  backgroundRGBColor: DWORD;
begin
  //-----
  // Produce an HTML page by merging:
  // _htmlBegin.txt + rtfContentsAsHtml + _htmlEnd.txt
  // Replace '<title>Title</title>' in _htmlBegin.txt with EditTitle.Text.

```

```

// Replace '&nbsp;' in rtfContentsAsHtml with spaces.
// Add <DIV> tag of RTFBackgroundColor and default Font to rtfContentsAsHtml.
// Save with Unix line endings, encoded in UTF-8
//-----
// Create a stringlist to load and save html code with
htmlSL := TStringList.Create();
// Load the HTML Header text
htmlSL.LoadFromFile('_htmlBegin.txt');
htmlOutText := htmlSL.Text;
// Make an uppercase version of htmlOutText to search for the title tag
uppercaseTestString := UpperCase(htmlOutText);
// Search for the title tag and replace it (if found)
posStart := pos('<TITLE>', uppercaseTestString);
if (posStart > 0) then begin
    posEnd := pos('</TITLE>', uppercaseTestString);
    if (posEnd > 0) then begin
        // delete and insert to htmlOutText
        delete(htmlOutText,
            posStart,
            (posEnd - posStart));
        insert('<title>' + Trim(EditWebpageTitle.Text), htmlOutText, posStart);
    end;
end;
// Create a advRichEditorMiniHTMLIO to get the RTF contents as HTML
advRichEditorMiniHTMLIO := TadvRichEditorMiniHTMLIO.Create(nil);
advRichEditorMiniHTMLIO.RichEditor := AdvRichEditor;
advRichEditorMiniHTMLIO.PlainHTML := true;
advRichEditorMiniHTMLIO.InlineImages := true;
// Get the RTF's HTML text
rtfContentsAsHtml := advRichEditorMiniHTMLIO.AsString();
// Replace '&nbsp;' with spaces.
posStart := pos('&nbsp;', rtfContentsAsHtml);
while (posStart > 0) do begin
    delete(rtfContentsAsHtml, posStart, length('&nbsp;'));
    insert(#32, rtfContentsAsHtml, posStart);
    posStart := pos('&nbsp;', rtfContentsAsHtml);
end;
// Add rtfContentsAsHtml to htmlOutText, adding a <DIV> tag around the
// rtfContentsAsHtml with the background color, default font, and font size.
backgroundRGBColor := ColorToRGB(AdvOfficeBkColorSelector.SelectedColor);
htmlOutText := htmlOutText +
    '<DIV style="background-color:#' +
        IntToHex(GetRValue(backgroundRGBColor), 2) +
        IntToHex(GetGValue(backgroundRGBColor), 2) +
        IntToHex(GetBValue(backgroundRGBColor), 2) +
        '; font-family: ' +
        RTFDefaultFontName +
        '; font-size:' +
        IntToStr(RTFDefaultFontSize) +
        'pt;">' +
    rtfContentsAsHtml +
    '</DIV>';
// Load the HTML Footer text
htmlSL.LoadFromFile('_htmlEnd.txt');
// Add the HTML Footer text to the htmlOutText
htmlOutText := htmlOutText + htmlSL.Text;
// Save the htmlOutText with Unix line endings, encoded in UTF8.
htmlSL.Text := htmlOutText;
htmlSL.LineBreak := #10;
htmlSL.SaveToFile('_htmlOut.htm', TEncoding.UTF8);
htmlSL.Free();
// Save the WebpageTitle in AdvRichEditor.Author
AdvRichEditor.Author := Trim(EditWebpageTitle.Text);
// Save the RTF
AdvRichEditor.SaveToFile('__AdvRichEditContents.rtf');
// Open the resulting HTML file in the default browser

```

```
ShellExecuteW(self.Handle,  
              'open',  
              PWChar(ExpandFileName('_htmlOut.htm')),  
              nil,  
              nil,  
              SW_SHOWNORMAL);
```

```
end;
```

```
//-----
```

```
//-----
```

```
end.
```

## HtmlRTFMergerMainUnit.dfm

```
object Form1: TForm1
  Left = 0
  Top = 0
  Caption = 'HtmlRTFMerger'
  ClientHeight = 664
  ClientWidth = 896
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'Tahoma'
  Font.Style = []
  OldCreateOrder = False
  OnCreate = FormCreate
  OnShow = FormShow
  DesignSize = (
    896
    664)
  PixelsPerInch = 96
  TextHeight = 13
  object EditWebpageTitle: TEdit
    Left = 8
    Top = 42
    Width = 879
    Height = 21
    Anchors = [akLeft, akTop, akRight]
    TabOrder = 0
    Text = 'Webpage Title'
  end
  object AdvRichEditor: TAdvRichEditor
    Left = 8
    Top = 69
    Width = 879
    Height = 587
    Cursor = crIBeam
    HorzScrollBar.Tracking = True
    HorzScrollBar.Visible = False
    VertScrollBar.Range = 6
    VertScrollBar.Tracking = True
    Anchors = [akLeft, akTop, akRight]
    Font.Charset = DEFAULT_CHARSET
    Font.Color = clBlack
    Font.Height = -11
    Font.Name = 'Arial'
    Font.Style = []
    TabOrder = 1
    TabStop = True
    ParentFont = False
    Version = '1.6.19.15'
  end
  object AdvRichEditorEditToolBar: TAdvRichEditorEditToolBar
    Left = 8
    Top = 8
    Width = 201
    Height = 28
    AllowFloating = True
    Caption = ''
    CaptionFont.Charset = DEFAULT_CHARSET
    CaptionFont.Color = clWindowText
    CaptionFont.Height = -11
    CaptionFont.Name = 'Tahoma'
    CaptionFont.Style = []
    CompactImageIndex = -1
```



```

ShowRightHandle = False
TextAutoOptionMenu = 'Add or Remove Buttons'
TextOptionMenu = 'Options'
ParentStyler = False
ParentOptionPicture = True
ToolBarIndex = -1
RichEditor = AdvRichEditor
Hints.FileOpenTitle = 'Open (Ctrl+O)'
Hints.FileOpenContent = 'Open new document from file'
Hints.FileSaveTitle = 'Save (Ctrl+S)'
Hints.FileSaveContent = 'Save document to file'
Hints.CutTitle = 'Cut (Ctrl+X)'
Hints.CutContent = 'Remove the selection to the clipboard'
Hints.CopyTitle = 'Copy (Ctrl+C)'
Hints.CopyContent = 'Put a copy of the selection on the clipboard'
Hints.PasteTitle = 'Paste (Ctrl+V)'
Hints.PasteContent = 'Add content on the clipboard to your document'
Hints.UndoTitle = 'Undo (Ctrl+Z)'
Hints.UndoContent = 'Undo typing'
Hints.RedoTitle = 'Redo (Ctrl+Y)'
Hints.RedoContent = 'Redo typing'
Options = [btFileOpen, btFileSave, btCopy, btPaste, btCut, btUndo, btRedo]
end
object AdvRichEditorFormatToolBar: TAdvRichEditorFormatToolBar
  Left = 200
  Top = 8
  Width = 692
  Height = 28
  AllowFloating = True
  Caption = ''
  CaptionFont.Charset = DEFAULT_CHARSET
  CaptionFont.Color = clWindowText
  CaptionFont.Height = -11
  CaptionFont.Name = 'Tahoma'
  CaptionFont.Style = []
  CompactImageIndex = -1
  ShowRightHandle = False
  TextAutoOptionMenu = 'Add or Remove Buttons'
  TextOptionMenu = 'Options'
  ParentStyler = False
  ParentOptionPicture = True
  ToolBarIndex = -1
  RichEditor = AdvRichEditor
  Hints.BoldTitle = 'Bold (Ctrl+B)'
  Hints.BoldContent = 'Select bold font style'
  Hints.ItalicTitle = 'Italic (Ctrl+I)'
  Hints.ItalicContent = 'Select italic font style'
  Hints.UnderlineTitle = 'Underline (Ctrl+U)'
  Hints.UnderlineContent = 'Select underline font style'
  Hints.StrikeThroughTitle = 'Strikethrough'
  Hints.StrikeThroughContent = 'Select strikethrough font style'
  Hints.SubScriptTitle = 'Subscript'
  Hints.SubScriptContent = 'Set subscript text'
  Hints.SuperScriptTitle = 'Superscript'
  Hints.SuperScriptContent = 'Set superscript text'
  Hints.InsertPictureTitle = 'Insert picture'
  Hints.InsertPictureContent = 'Insert a picture from file'
  Hints.InsertSpecialCharTitle = 'Insert special character'
  Hints.InsertSpecialCharContent = 'Insert a special character'
  Hints.BulletTitle = 'Insert bullet'
  Hints.BulletContent = 'Insert a bullet for list'
  Hints.NumberedBulletTitle = 'Start list'
  Hints.NumberedBulletContent = 'Start a numbered list'
  Hints.TextColorTitle = 'Text color'
  Hints.TextColorContent = 'Set selection text color'
  Hints.BackgroundColorTitle = 'Text background color'

```

```

Hints.BackgroundColorContent = 'Set selection background color'
Hints.AlignLeftTitle = 'Align text left (Ctrl+L)'
Hints.AlignLeftContent = 'Align the text to left'
Hints.AlignCenterTitle = 'Align center (Ctrl+E)'
Hints.AlignCenterContent = 'Center text'
Hints.AlignRightTitle = 'Align text right (Ctrl+R)'
Hints.AlignRightContent = 'Align the text to right'
Hints.InsertHyperlinkTitle = 'Set hyperlink'
Hints.InsertHyperlinkContent = 'Set hyperlink for text'
Hints.IndentTitle = 'Increase indent'
Hints.IndentContent = 'Increase the indent level of the paragraph'
Hints.UnIndentTitle = 'Decrease indent'
Hints.UnIndentContent = 'Decrease the indent level of the paragraph'
Options = [btBold, btItalic, btUnderline, btStrikeThrough, btSubscript, btSuperScript,
btInsertPicture, btInsertspecialChar, btBullet, btNumberedBullet, btTextColor,
btBackgroundColor, btAlignLeft, btAlignCenter, btAlignRight, btInsertHyperlink, btIndent,
btUnindent]
object AdvOfficeBkColorSelector: TAdvOfficeColorSelector
  Left = 664
  Top = 2
  Width = 24
  Height = 24
  Hint = 'Bold (Ctrl+B)'
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'Tahoma'
  Font.Style = []
  FocusType = ftHot
  NotesFont.Charset = DEFAULT_CHARSET
  NotesFont.Color = clWindowText
  NotesFont.Height = -11
  NotesFont.Name = 'Tahoma'
  NotesFont.Style = []
  OfficeHint.Title = 'Bold (Ctrl+B)'
  OfficeHint.Notes.Strings = (
    'Select bold font style')
  ParentFont = False
  Version = '1.6.0.4'
  TabOrder = 20
  AllowFloating = False
  CloseOnSelect = False
  CaptionAppearance.BorderColor = clNone
  CaptionAppearance.Color = 13198890
  CaptionAppearance.ColorTo = clNone
  CaptionAppearance.Direction = gdHorizontal
  CaptionAppearance.TextColor = clWhite
  CaptionAppearance.TextColorHot = clBlack
  CaptionAppearance.TextColorDown = clBlack
  CaptionAppearance.ButtonAppearance.ColorChecked = 16111818
  CaptionAppearance.ButtonAppearance.ColorCheckedTo = 16367008
  CaptionAppearance.ButtonAppearance.ColorDisabled = 15921906
  CaptionAppearance.ButtonAppearance.ColorDisabledTo = 15921906
  CaptionAppearance.ButtonAppearance.ColorDown = 16111818
  CaptionAppearance.ButtonAppearance.ColorDownTo = 16367008
  CaptionAppearance.ButtonAppearance.ColorHot = 16117985
  CaptionAppearance.ButtonAppearance.ColorHotTo = 16372402
  CaptionAppearance.ButtonAppearance.ColorMirrorHot = 16107693
  CaptionAppearance.ButtonAppearance.ColorMirrorHotTo = 16775412
  CaptionAppearance.ButtonAppearance.ColorMirrorDown = 16102556
  CaptionAppearance.ButtonAppearance.ColorMirrorDownTo = 16768988
  CaptionAppearance.ButtonAppearance.ColorMirrorChecked = 16102556
  CaptionAppearance.ButtonAppearance.ColorMirrorCheckedTo = 16768988
  CaptionAppearance.ButtonAppearance.ColorMirrorDisabled = 11974326
  CaptionAppearance.ButtonAppearance.ColorMirrorDisabledTo = 15921906
  DragGripAppearance.BorderColor = clGray

```

```
DragGripAppearance.Color = clWhite
DragGripAppearance.ColorTo = clWhite
DragGripAppearance.ColorMirror = clSilver
DragGripAppearance.ColorMirrorTo = clWhite
DragGripAppearance.Gradient = ggVertical
DragGripAppearance.GradientMirror = ggVertical
DragGripAppearance.BorderColorHot = clBlue
DragGripAppearance.ColorHot = 16117985
DragGripAppearance.ColorHotTo = 16372402
DragGripAppearance.ColorMirrorHot = 16107693
DragGripAppearance.ColorMirrorHotTo = 16775412
DragGripAppearance.GradientHot = ggRadial
DragGripAppearance.GradientMirrorHot = ggRadial
DragGripPosition = gpTop
Appearance.ColorChecked = 16111818
Appearance.ColorCheckedTo = 16367008
Appearance.ColorDisabled = 15921906
Appearance.ColorDisabledTo = 15921906
Appearance.ColorDown = 16111818
Appearance.ColorDownTo = 16367008
Appearance.ColorHot = 16117985
Appearance.ColorHotTo = 16372402
Appearance.ColorMirrorHot = 16107693
Appearance.ColorMirrorHotTo = 16775412
Appearance.ColorMirrorDown = 16102556
Appearance.ColorMirrorDownTo = 16768988
Appearance.ColorMirrorChecked = 16102556
Appearance.ColorMirrorCheckedTo = 16768988
Appearance.ColorMirrorDisabled = 11974326
Appearance.ColorMirrorDisabledTo = 15921906
SelectedColor = clNone
ShowRGBHint = True
ColorDropDown = 16251129
ColorDropDownFloating = 16374724
SelectionAppearance.ColorChecked = 16111818
SelectionAppearance.ColorCheckedTo = 16367008
SelectionAppearance.ColorDisabled = 15921906
SelectionAppearance.ColorDisabledTo = 15921906
SelectionAppearance.ColorDown = 16111818
SelectionAppearance.ColorDownTo = 16367008
SelectionAppearance.ColorHot = 16117985
SelectionAppearance.ColorHotTo = 16372402
SelectionAppearance.ColorMirrorHot = 16107693
SelectionAppearance.ColorMirrorHotTo = 16775412
SelectionAppearance.ColorMirrorDown = 16102556
SelectionAppearance.ColorMirrorDownTo = 16768988
SelectionAppearance.ColorMirrorChecked = 16102556
SelectionAppearance.ColorMirrorCheckedTo = 16768988
SelectionAppearance.ColorMirrorDisabled = 11974326
SelectionAppearance.ColorMirrorDisabledTo = 15921906
SelectionAppearance.TextColorChecked = clBlack
SelectionAppearance.TextColorDown = clWhite
SelectionAppearance.TextColorHot = clWhite
SelectionAppearance.TextColorDisabled = clGray
SelectionAppearance.TextColor = clBlack
SelectionAppearance.Rounded = False
Tools = <
  item
    BackGroundColor = clBlack
    Caption = 'Automatic'
    CaptionAlignment = taCenter
    Enable = True
    Hint = 'Automatic'
    ImageIndex = -1
    ItemType = itFullWidthButton
  end
end
```

```
item
  BackGroundColor = clBlack
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = 13209
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = 13107
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = 13056
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = 6697728
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = clNavy
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = 3486515
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = 3355443
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = clMaroon
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = 26367
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = clOlive
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
```

```
item
  BackGroundColor = clGreen
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = clTeal
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = clBlue
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = 10053222
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = clGray
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = clRed
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = 39423
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = 52377
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = 6723891
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = 13421619
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = 16737843
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
```

```
item
  BackGroundColor = clPurple
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = 10066329
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = clFuchsia
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = 52479
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = clYellow
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = clLime
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = clAqua
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = 16763904
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = 6697881
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = clSilver
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
item
  BackGroundColor = 13408767
  CaptionAlignment = taCenter
  Enable = True
  ImageIndex = -1
end
```

```

    item
      BackGroundColor = 10079487
      CaptionAlignment = taCenter
      Enable = True
      ImageIndex = -1
    end
  item
    BackGroundColor = 10092543
    CaptionAlignment = taCenter
    Enable = True
    ImageIndex = -1
  end
  item
    BackGroundColor = 13434828
    CaptionAlignment = taCenter
    Enable = True
    ImageIndex = -1
  end
  item
    BackGroundColor = 16777164
    CaptionAlignment = taCenter
    Enable = True
    ImageIndex = -1
  end
  item
    BackGroundColor = 16764057
    CaptionAlignment = taCenter
    Enable = True
    ImageIndex = -1
  end
  item
    BackGroundColor = 16751052
    CaptionAlignment = taCenter
    Enable = True
    ImageIndex = -1
  end
  item
    BackGroundColor = clWhite
    CaptionAlignment = taCenter
    Enable = True
    ImageIndex = -1
  end
  item
    BackGroundColor = -2
    Caption = 'More Colors...'
    CaptionAlignment = taCenter
    Enable = True
    Hint = 'More Colors'
    ImageIndex = -1
    ItemType = itFullWidthButton
  end>
  OnSelectColor = AdvOfficeBkColorSelectorSelectColor
end
end
end

```

**[END OF CODE]**