WinWord 2.0 to HTML

Package:

This package includes following files:

```
HTML.DOT          WinWord 2.0 template file including some
                  styles for writing and some macros to handle
                  document as HTML.

             The conversion DLL.
             The inifile for the converter including good defaults.

RTFTOHTM.LIB            The import library for the DLL.
RTFTOHTM.H        Public interface of the DLL.

HTMTOOLS.DOC            Some documentation about the converter in Word 6 format (this
file).
HTMTOOLS.HTM            Documentation converted to .HTM.

RTF2HTM.EXE       QuickWin application for converting files.
RTF2HTM.C         Source for above.
RTF2HTM.MAK       Makefile for above.

README.TXT        Quick document.
```

There are couple words about at the end of this document.

Changes and corrections from 0.8x to 0.9x

The template file with some macros for WinWord 6.0.
Minor corrections to rtf2htm.exe.
Empty buffer handling corrected in converter.

Changes and corrections from 0.70 to 0.8x

Local references (within one document) can be made by specifying Bookmarks. Bookmarks generate an anchor (<A NAME=...>). If the local references are done by pressing **'D'** button in the toolbar, Word shows the bookmarks name (as invisible). This feature can not be used, if the old features are in use.

Jump button for local reference can be made by specifying Gotobutton. Gotobutton generate an URL to local reference (<A HREF=#ref>xxx</A>

URLs can be made as WinWord Annotations.
The contents of \fldrslt are to be printed into current destination.
Base style added for use within Header to produce <BASE> tag.
(UL,OL) can be nested.
style added.
can be given by WinWord Summary Info or by creating First Page Header
WinWord now convert (somehow) to preformatted tab separated 'tables'.
All supported paragraph styles are now configurable (to the certain level).
Some of the System variables can be set via an AboutBox (question mark button).
Several minor bugs corrected.

Changes and corrections from 0.60 to 0.70

added. There are some variables in .INI file including line length proposals for preformatted and the rest of styles.

The <HTML> tags added.

Some extra (unneeded) <P> tag generations removed.

The attribute values of <A> and <IMG> tags enclosed by double quotes. Some browsers and editors require this.

Changes and corrections from 0.50 to 0.60

Some entities have been removed.

Handling of \par<CRLF>\par<CRLF> correct now. Thanks to Robert Thompson to notice this bug.

Some internal structures changed to be more robust.

The function RTFtoHTM returns the length of the destination correctly.

Document template

HTML.DOT includes some HTML styles. This document template is not meant to be perfect HTML editing system, but I found it quite useful with companion of the conversion dll; RTFTOHTM.DLL. Here are some features offered by this software.

Document HEADer

There is possible to use page header to generate <HEAD> section to html file. There is no error checking on it, so the feature should be used correctly by convention. There should be only one header in the document. You can create such by View_Header/Footer. In the Header/Footer dialog You should check 'Different First Page', and select Header. This makes 'headerf' style to the beginning of the first page.

Within the header You should use only TITLE, BASE, and LINK styles. You can make a paragraph to TITLE by selecting it, and clicking **T** button in the button bar or selecting a style **Title** from style list in ribbon. You can make LINKs into header by selecting text and setting its style to Link.

If you define a header into document, converter creates both HEAD and BODY sections to resulting HTML file.

NewFeature:

This header info can be given by using File Summary Info command in WinWord. Within you can define which of the Info fields are used, and within which tags they are produced into resulting file.

Heading 1 to Heading 6

Paragraph to be converted to a heading should be selected. It is enough if a caret is blinking within a paragraph. Then either clicking **E** button in the button bar or selecting an appropriate style from a style list.

Emphasizing

There are some emphasis inline styles. You can use **Bold**, *Italics*, Underline, Word Underline by appropriate WinWord emphasis. The interpretation of **Bold** is <B>, *Italics* is <I>, Underline is <U>, and Word Underline is <EM>.

Block emphasizing

There are some types of block emphasizing styles:

Preformatted

```
This is a preformatted text as HTML style <PRE>.
This style is applied by selecting this text, and clicking P button in the button bar.
This style can contain other HTML styles, such as anchors, emphasizing inline styles...

The default interpretation for Line breaks is <CRLF> and for Paragraph breaks
<CRLF><CRLF>. Best way to break lines in preformatted style is to use
LineBreak (Shift-Enter) in WinWord. See more info at chapter .
```

Listing

```
This is a text as HTML style <LISTING>. This style is essentially the same as
preformatted, and was made mostly to allow separate (different style for 'preformatted
text. This style is applied by selecting this text, and selecting its style (Listing)
from style list in WinWord cormatting bar (ribbon). This style can also contain
other HTML styles.

The default interpretation of Line and Paragraph breaks are both <CRLF>.
```

Block Quote

> This style is BlockQuote. This is intended to quote some larger block of text, and is emphasized in some way. This style is applied by selecting text, and clicking **B** button in the button bar.

Address

This is an Address style. These styles are presented differently in different parsers. You can correct a presentation in WinWord simply by formatting style by your own taste. Paragraph written in this style is aligned right.

Lists

There are two types of lists available. Ordered list as style Order List, and bulleted list as style Unnum List. These types You can attach to text by selecting text, and clicking a number list or bullet list button in the button bar. Each paragraph within selected text becomes a list item. The converter doesn't allow nesting these styles. If you want to nest them, you must use HTML style (below) and write a markup yourself.

Numbered list

1. This is a numbered list item one
2. This is item two
3. And item three
4. Style is ***Order List***

Bulleted list

- This is a bulleted list item one
- This is item two
- And item three
- Style is ***Unnum List***

Bulleted lists can be nested

- This is a bulleted list item one
- This is item two
  - This is subitem one for two
  - This is subitem two for two
- And item three
- Style is ***Unnum List***

You can nest the list items simply giving them some indentation. The converter interprets the more indentation the deeper nesting. The same amount of indentation means the same level of nesting.

Tables

The tables made by defining table in WinWord now converts (somehow) to HTML preformatted. Columns are separated by tabs (one or more). Here is an example:

ID	**Name**	Description	Done	To do
R001	FORM01	Manage the member	5	x
R002	FORM02	Browse the members	5	x
R003	FORM03	Manage the department	x	y
R004	FORM04	Browse the departments	5	x

HR style

Those horizontal lines visible in WinWord document are made by making an empty line, and selecting HR style for it.

HTML style

The style HTML allows user to embed 'plain' HTML into Word document. The converter doesn't touch to this style in any way. No entity interpretation is made. The style HTML allows user to enter such HTML tags or styles this converter doesn't support in other way. Here is an example.

```
<FORM ACTION="http://jch.vtkk.fi/testpost" METHOD="POST">
( E-mail to:
<SELECT NAME="mailinfo" SIZE=1>
<OPTION SELECTED>"Jorma.Hartikka@csc.fi"
<OPTION>"jhartik@finsun.csc.fi"
<OPTION>"Xhartik@finsun.csc.fi"
</SELECT>
Subject:
<SELECT NAME="subjectinfo" SIZE=3>
<OPTION SELECTED>"Testing 1..."
<OPTION>"Testing 2..."
<OPTION>"Testing 3..."
<OPTION>"Testing 4..."
</SELECT> )
<HR>
Name: <INPUT NAME="username" VALUE="Testusername" SIZE=30><P>
E-Mail: <INPUT NAME="usermail" VALUE="Testusermail" SIZE=40>
Please enter any comments on whether or not you found this server useful, and how it could be improved.<P>
<INPUT NAME="testfeedback" SIZE="60,4" VALUE="Testfeedback"><P>
Please push this <INPUT TYPE="submit" VALUE="S&ubmit"> to mail your feedback.
</FORM>
```

Hypertext links

Hypertext links has specific syntax in this template, but nothing denies user to make his own... The styleurl is composed as follows: There is an url separated by space from following text. An url part of the text becomes hidden red, and button part becomes double underlined blue.

http://www.ncsa.uiuc.edu/demoweb/demo.html This is a demo page of NCSA.

By selecting whole url including a button text and clicking **L** button in the button bar makes text as following:

If ButtURL = 0 in the macro generates this.

This is a demo page of NCSA.

If ButtURL = 1 in the macro generates this. Double clicking the symbol before the button text opens WinWord's annotation pane, where you can edit this (and other) URLs in this document. This is new feature since version 080.

This is a demo page of NCSA.

You can also define an hypertext link pointing to specific part of current document by defining a internal refernce in the link. Here is an example of that kind of link. Only difference to the 'ordinary' link is that a position (bookmark) within destination document is present.

The anchor can be made by defining a name and label. This can be made by selecting an anchor text and pressing D button in the button bar.

This is a destination for above 'testjump'.

Notice that although I use SmallCaps style for a name of the anchor, the case of the characters is preserved in the resulting HTML file.

Inline images

Inline graphics is quite problematic in WYSIWYG point of view. I have made it in easy way (for now). It is presented simply as strike through formatting in WinWord. This allows its use as a link button in the hypertext link.

This is stand alone image: ~~/images/JCHS.GIF~~

Here is a button image: ~~/images/JCHS.GIF~~

Inline image can act as button. Converter cannot generate following:

<A HREF=anchor><IMG SRC=jchs.gif> The Author</A>

A button field of the hyperlink can contain either an image or plain text but not both.

Undoing HTML

N button in the button bar resets style of the selected paragraph(s) back to normal. **U** button removes some character formatting properties from the selected text. It can be used to reset hyperlinks or image links back to normal.

Saving a file

You can save a file as *.HTM by pressing S button in the button bar. File can be saved as *.RTF by pressing R button. Saved file gives a same name as current WinWord document and an extension .HTM or .RTF depending a save format.

Fine tuning of a resulted HTML file

The resulted HTML file should be edited with plain text editor to clean it. WinWord 2.0 RTF filter may do nasty breaks within anchors, button text, or inline, if it spans certain memory limits, so they should be corrected before publishing a file.

Installation

Copy HTML.DOT into WinWord directory, and RTFTOHTM.DLL somewhere in PATH (eg. WINWORD or WINDOWS). To begin the new document using this template, open New from File menu and choose HTML.DOT as document template. To attach HTML.DOT into old document, use Template... command from File menu. In Template dialog choose 'Attach Document to' HTML, and click 'Store new Macros and Glossaries as' 'With document Template'; click OK. This installs the styles into current document. To install the macros, use Format Style command. In Style dialog click Define, Merge, and choose html.dot; click From Template; click Close.

RTFTOHTM.INI

If you choose to change .INI settings, RTFTOHTM.INI file must be in the same directory with the converter dll. If this .INI file doesn't exist, the converter uses internal defaults which are equivalent with following .INI settings:

```
[System]
ZoneMin = 64
ZoneMax = 80
GetStyles = 0
ButtonURL = 1
Bookmark = bkmk
HtmlDir = D:\pub\txtsrv\testdocs
FirstTab = 0
TabStops = 8
Entities = 1
Debug = 0

[Header]
;Process the RTF Info header
ProcessInfo = 1
;Don't process the RTF header section
ProcessHead = 0
;Get a title
TitleOK = 1
SubjectOK = 0
AuthorOK = 0
KeywordsOK = 0
;Get a document comment
DoccommOK = 1
TitleTag = "\n<TITLE>"|"</TITLE>"
SubjectTag = "\n<TITLE>"|"</TITLE>"
AuthorTag = "\n<AUTHOR>"|"</AUTHOR>"
KeywordTag =
DoccommTag ="<BASE HREF="|">"

[Styles]
;StyleName    Begin              End                Para         Line         Interpret
Normal      = "\n<P>"|           "\n<P>"|           "\n<P>"|     "<BR>\n"|    IPR_NONE
Header      = ""|                ""|                ""|          ""|          IPR_NONE
Title       = "\n<TITLE>"|       "</TITLE>\n"|       ""|          ""|          IPR_NOBREAK
Link        = "\n<LINK HREF="|   ">\n"|             ""|          ""|          IPR_NOBREAK
Base        = "\n<BASE HREF="|   ">\n"|             ""|          ""|          IPR_NOBREAK
Heading 1   = "\n<H1>"|          "</H1>\n"|          ""|          ""|          IPR_NOBREAK
Heading 2   = "\n<H2>"|          "</H2>\n"|          ""|          ""|          IPR_NOBREAK
Heading 3   = "\n<H3>"|          "</H3>\n"|          ""|          ""|          IPR_NOBREAK
Heading 4   = "\n<H4>"|          "</H4>\n"|          ""|          ""|          IPR_NOBREAK
Heading 5   = "\n<H5>"|          "</H5>\n"|          ""|          ""|          IPR_NOBREAK
Heading 6   = "\n<H6>"|          "</H6>\n"|          ""|          ""|          IPR_NOBREAK
Address     = "\n<ADDRESS>\n"|   "\n</ADDRESS>\n"|   "\n<P>"|    "<BR>\n"|    IPR_BREAK
BlockQuote  = "\n<BLOCKQUOTE>\n"|"\n</BLOCKQUOTE>\n"|"\n<P>"|   "<BR>\n"|    IPR_BREAK
Preformat   = "\n<PRE>\n"|       "\n</PRE>\n"|       "\n\n"|      "\n"|        IPR_CRLF
Listing     = "\n<LISTING>\n"|   "\n</LISTING>\n"|   "\n"|        "\n"|        IPR_LISTING
Unnum List  = "\n<UL>"|          "\n</UL>"|          "\n<LI>"|    "<BR>\n"|    IPR_LIST
```

```
Order List  = "\n<OL>"|         "\n</OL>"|          "\n<LI>"|   "<BR>\n"|   IPR_LIST
HTML        = ""|               ""|                "\n"|       "\n"|       IPR_HTML
HR          = "\n<HR>\n"|       ""|                ""|        ""|        IPR_NONE
```

System section

ZoneMin and **ZoneMax** (numeric) Define the proposal for line break zone for most of the styles in resulting .HTM file. These variables do not affect to Preformatted and Listing styles. If one of these is zero, the converter ignores both of these variables. Anchor and Img tags may go over the limits. The converter doesn't break line within those tags.

GetStyles (Boolean 0 or 1). If 1, the converter gets style models from section [Styles] in RTFTOHTM.INI.

ButtonURL (Boolean 0 or 1). If 1, the new features according to the way specifying an URL to the hypertext link are in effect, otherwise the old method is.

FirstTab (Numeric). Affects the table conversion. Defines, how many tabs the converter should place to the beginning of each row.

TabStops (Numeric). Affects the table conversion. Defines, how many characters is one tab stop.

Entities (Boolean 0 or 1). If 1, certain characters are to be converted to the entities defined in HTML DTD, otherwise they are not converted. Use this carefully, because if set to zero, you can not put any HTML reserved characters in the text.

Bookmark (text). Defines the bookmark prefix.

HtmlDir (dirpath). If present, defines the path to save the *.HTM files. If omitted, the converter uses the same directory where the *.doc document resides

Debug (Boolean 0 or 1). If 1, the converter produces some (internal) debugging info into .HTM file.

Header Section

ProcessInfo (Boolean 0 or 1). If 1, The converter searches the {\info ...} section in .RTF file to produce <HEAD> section to resulting file. This information can be given by using File Summary Info command.

ProcessHead (Boolean 0 or 1). If 1, The converter searches the {\headerf ...} style in .RTF file to produce <HEAD> section to resulting file. Both or either of previous variables can be omitted (or 0). If both are 1, the converter uses ProcessInfo.

TitleOK (Boolean 0 or 1). If 1, Seraches document Title.

SubjectOK (Boolean 0 or 1). If 1, Seraches document Subject.

AuthorOK (Boolean 0 or 1). If 1, Seraches document Author.

KeywordsOK (Boolean 0 or 1). If 1, Seraches document Keywords.

DoccommOK (Boolean 0 or 1). If 1, Seraches document Comments.

TitleTag (text). Defines the begin and end markups for appropriate field. If the field is found, the converter concatenates first (to the left of vertical bar) string, the field contents, and last string. The tag can be omitted.If only the first part is omitted, vertical bar must be there as placeholder.

SubjectTag (text). As in previous.

AuthorTag (text). As in previous.

KeywordsTag (text). As in previous.

DoccommTag (text). As in previous.

Styles section

In this section the variable names refer to the appropriate style in WinWord. The value consists five parts:

Begin tag
What the converter produces to beginning of this style.
End tag
What the converter produces to end of this style.
Para
What the converter uses as paragraph separator within this style.
Line
What the converter uses as line separator within this style.
Interpret
Tells, how the converter interprets the paragraph (\par), paragraph definition (\pard), and line (\line) tags from .RTF. These symbols can not be combined.
IPR_NONE:
Nothing special. When the parser sees \par, it produces the paragraph mark appropriate to current style.
IPR_BREAK:
This interpretation is essentially the same as IPR_NONE.
IPR_NOBREAK:
This style can not contain any line breaking characters.
IPR_CRLF:
This is mostly used in PRE. The entities are used, and paragraph and line breaks are as they are in .RTF file.
IPR_LISTING:
This is essentially the same as IPR_PRE. The main reason for this is to give two different styles for 'preformatted' data.
IPR_HTML:
This is mandatorial interpretation. When this is in effect, the text read from .RTF file is produced to resulting file as is. No entity translations is made.
IPR_LIST:
This interpretation is made for nestable lists and menus.

RTFTOHTM.DLL

There is one exported function in the converter DLL. C-prototype is as follows:

```
long FAR PASCAL RTFtoHTM(LPSTR iname, int ilen, LPSTR oname, int olen);
LPSTR ibuff;      //input buffer or filename.
int   ilen; //length of input buffer. if zero, iname is treated as filename.
LPSTR obuff;      //output buffer or filename.
int   olen; //length of output buffer. if zero, oname is treated as filename.
```

The return value represents a length (in chars) of a resulted HTML file or return buffer.

Basic (or Word macro) declaration is as follows:

```
Declare Function RTFtoHTM Lib "rtftohtm.dll" (iname$, ilen As Integer, oname$, olen As
Integer) As Integer
```

The meaning of parameters are the same as above. If the function is to be used with Visual Basic, all the parameters must be declared ByVal.

Name: Jorma Hartikka Senior Programmer Analyst
Org: VTKK Government Systems ltd/Information Systems
Email: Jorma.Hartikka@csc.fi