

## **TextToHTML 1.2.6B**

by Kris Coppieters - © 1995 Logic n.v.. All rights reserved  
Derbystraat 25  
B-9051 Sint-Denijs-Westrem  
Belgium - Europe  
*Internet address: 100025.2724@compuserve.com*

*July 14, 1995*

Check out <http://www.logic.be> for the latest version.

If you are updating an older version: the update history is appended at the end of this document.

This software is freeware. Please do not distribute it without this Readme file. You may customize this software for your own purposes, but please don't redistribute customized versions: only unmodified versions can be distributed. This software is distributed 'as is': no warranties are made, either expressed or implied.

Thanks to Paul Van Cotthem of Apple Computer Belgium: many of the features of this program were suggested by him.

*Internet address: vancotthem.p@applelink.apple.com*

## **WHAT DOES IT DO?**

TextToHTML quickly converts text or RTF files dropped onto its icon into HTML format. When converting **multiple** text, RTF or HTML files **at once**, it also creates an HTML index file with links to those files, complete with graphical bullets if necessary. It can also automatically link images. With TextToHTML you can set up a basic WWW structure in 15 minutes, starting from scratch.

(RTF = Rich Text Format, used by several word processors; HTML = Hyper Text Markup Language, used for formatting documents on the WWW = World Wide Web).

- The first 'real' text line it finds in an original text or RTF file is interpreted to be the window title and it is also tagged as an <H1> header. The rest of the file is considered to be body text.
- For reference purposes, the filename of the original file is added at the end of the resulting HTML file.
- Single carriage returns or line feeds are converted to the <BR> tag, multiple sequential carriage returns are collapsed into a <P> tag.
- Special and reserved characters are converted according to the list of ISO Latin-1 character entities. Mac characters without ISO Latin-1 counterpart are converted to character sequences with some visual similarity.  
(For more information on this, check out this URL: <http://www.utirc.utoronto.ca/HTMLdocs/NewHTML/entities.html>).
- The creator of the converted HTML files is set to MOSS (Netscape). You can change this (see below).
- If there are GIF or JPEG files present for use as a logo, bullets, or illustrations, they can be linked in automatically.

## **HOW TO USE IT?**

There are three ways to use it:

1) If you **drop a single file** onto the TextToHTML application icon, it gets converted into HTML format and it is saved in the same folder as the original file. The name of the resulting HTML file will be a derivative of the original file's name, with ".html" appended.

Tip: if you have no permission to write in this folder, TextToHTML will just beep, and do nothing. This can occur on CD-ROMS and on file servers. First copy the file to a location where you do have write access.

Tip: if you want to run the conversion again on the same file, you don't have to trash the previous .html file first, because it will be overwritten.

Tip: you can use the resulting .html file as a base to start with, and add graphics and extra links manually with an HTML editor.

Tip: if your Macintosh desktop is corrupt, TextToHTML will not let you drop anything. In that case, rebuild your desktop first (hold Command-Option during startup).

2) If you **drop multiple files** onto the TextToHTML application icon, they all get converted into HTML format, and an additional index file with hot links to the other files is created. The converted files are stored in the same folder as the index file. Except for the index file, the names of the HTML files are of the form "X####\_<name>.html", where #### is a sequential number, and <name> is a derivative of the original file name. Start browsing from the index file. Each "X" file contains a backlink to the main index file.

Tip: If you hold the option key while dropping multiple files onto TextToHTML, the files are processed one by one without creating an index file.

3) If you **drop a folder** onto the TextToHTML application, it converts all the files in that folder and its sub-folders. It creates a main index file, which is the root of a tree-structure that mimics the structure of the folders. The HTML files are all stored in the same folder as the main index file. Each "X" file contains backlinks to the main index file and to the file one level up in the tree structure.

Tip: if TextToHTML finds .html files in the original folder structure, they are just copied without changes, and linked into the menu structure. Their content is not changed or updated in any way - so check their links afterwards!

Tip: you can use this last feature to insert 'gateway'-documents to distant pages in your menu structure.

Tip: don't be alarmed if a certain sub-folder does not appear in the index tree: if there are no convertible documents in a folder or one of it's subfolders, it is not added to the index tree.

## **ADDING GRAPHICS**

If you have illustrations that you want to include, you should first put them in the **destination** folder, before using TextToHTML. The destination folder is the folder where you will save the index file. For every original TEXT or RTF file called <file>, there will be an auto-inclusion of one or more of the files called:

<file>.jpg  
<file>.gif  
<file>\_t.jpg  
<file>\_t.gif  
<file>\_m.jpg  
<file>\_m.gif  
<file>\_b.jpg  
<file>\_b.gif

For example: if you have a file called *Kris* somewhere in the source folder, and you provide a file *Kris.gif* in the destination folder before the conversion, the image *Kris.gif* will be auto-inserted just after the title in the HTML file that results from converting the file *Kris*.

<file>.gif and/or <file>.jpg is included just after the title in the converted file.

<file>\_t.gif and/or <file>\_t.jpg (t stands for 'top') is included before the converted file, just after the optional logo.

<file>\_m.gif and/or <file>\_m.jpg (m stands for 'middle') is included after the title in the converted file, and eventually after <file>.gif and <file>.jpg if one of these is present.

<file>\_b.gif and/or <file>\_b.jpg (b stands for bottom) is included after the body of the converted file.

Try the included example - it will make things easier to understand!

## **ADDING LOGO AND BULLETS**

If you want to insert your logo or some graphical bullets in all HTML files, you should provide one or more of the following four files in the **destination** folder before the conversion:

*logo.gif* or *logo.jpg* is inserted at the top at every converted file. Put your company logo in this file.

*bullet.gif* or *bullet.jpg* is prefixed to every link to a converted document.

*submenu.gif* or *submenu.jpg* is prefixed to every link to a submenu (corresponds to a subfolder).

*index.gif* or *index.jpg* is prefixed to every link back to the previous index file.

*topindex.gif* or *topindex.jpg* is prefixed to every link back to the top index file.

Check the included example. In the example, *topindex.gif* and *index.gif* look equal because of lack of inspiration on my side...

### **YOU SHOULD KNOW THAT...**

TextToHTML has very little error handling: it just beeps and quits. Possible causes: application size too small, no write access, or just plain bugs. Let me know if you encounter them!

TextToHTML currently ignores character formatting in text files. For example if you would convert this Readme file (which was created with SimpleText), the bold, italic and underline styles are not converted into HTML tags.

TextToHTML also reads and converts Mac RTF files, but this is still pretty beta: if you encounter problems with a particular RTF file, I would be happy if you sent it to me over e-mail, so I can further debug TextToHTML.

You should be careful not to use names like the ones TextToHTML uses for your own HTML files, to avoid name conflicts.

### **HOW TO CUSTOMIZE IT ?**

Please do not redistribute customised versions!

From version 1.2.6 on, TextToHTML can be heavily customised. Translation to other languages is easy, generated HTML code can be changed; you could even customise it into something completely different (e.g. an SGML convertor instead of an HTML convertor). You need ResEdit to do these customisations.

There are some STR# resources you can change: 20000, 20001, 20002, and 21000. Most of them are documented internally: just open them and read the comments.

- 20000 contains parameters (creator, texts used,...).

Tip: if you want to change the creator from Netscape (MOSS) to another application, change string 2.

Tip: if you want the resulting index files sorted by file name, rather than by document title, change string 10 to a 1.

**Attention:** when (and only when) you do this, TextToHTML will strip leading spaces and digits from folder names before using them in an index file. This allows you to customise the order of your documents and subfolders in the Finder by prefixing them with one or more spaces and digits 0-9. Use the 'View by Name' setting to look into the main folder. The leading digits and spaces will not show up in the HTML menus.

- 20001 contains the translation table: there are 255 strings, one for every ASCII character except NUL. If a string is empty, the character is not translated; otherwise the character is replaced.

- 20002 contains strains of HTML code that is inserted in different zones of the output file. The following zones are defined (zone names between {}):

a) regular output file

```
{DocumentTop}
Window title
{DocumentHeader}
Text title (equals Window title)
{DocumentBody}
some text followed by a single <CR>... {EndofLine}
some text followed by multiple <CR>... {EndOfParagraph}
```

{EndOfDocument}

b) Index outputfile

```
{IndexTop}  
Window title  
{IndexHeader}  
Index title (equals Window title)  
{IndexBody}  
{DocumentURL}URL{DocumentLinkText}Link text to document{DocumentLinkEnd}  
...  
{IndexURL}URL{IndexLinkText}Linktext to subindex{IndexLinkEnd}  
...  
{EndOfIndex}
```

Attributes from RTF files:

```
{BoldStart}, {BoldEnd}  
{ItalicStart}, {ItalicEnd}  
{UnderlineStart}, {UnderlineEnd}
```

For example: the {DocumentTop} resides in STR# 20002 - string #2, and contains "<HTML>\r<HEAD>\r<TITLE>\r". This inserts the normal HTML header in an output document.

Tip: You can easily insert some extra HTML code (e.g. an extra <HR>) in the converted files in different locations: before the header, after the header, at the end of the file, before a hotlink, after a hotlink,....

In the zone strings, heavy use is made of conditional macros to insert links and images when these are available or necessary.

- 21000 is a set of macro-strings. Currently strings 1-49 are used for lots of different strings: links to images, forward and backward links,... String# 50-... are used to keep most language-dependent strings: normally you only need to translate String #50 and onward, and eventually STR#20000 - String #4 and String #6. See below for an explanation about macros.

## **MACROS**

I do not intend to document these extensively - I implemented them mostly to make life a little easier for myself when customising TextToHTML, and not everything is tested: so no warranties! In short, everything between « and » (ASCII 199 and 200), is interpreted as a macro. Macros are evaluated recursively.

- If this macro string is numeric, it is replaced by the string with the same number from STR# 21000. If this string contains macros, they are also evaluated.

- There are also some symbolic macros:

```
«IFN» is the name of the original file  
«OFN» is the name of the HTML file  
«IDX» is the name of the index file one level up  
«TIDX» is the name of the top level index file  
«PCOUNT» is the number of <P> strings generated up to this point.  
«BRCOUNT» is the number of <BR> strings generated up to this point.
```

In later versions, more of these symbolic macros will become available.

- Conditional macros:

«?string1:=string2» is a conditional macro that only inserts string2 if string1 is an existing file in the destination folder.

«!string1:=string2» is a conditional macro that only inserts string2 if string1 is not empty

- You can also replace the ':= ' by '::'. If you do so, string2 will be put through an extra translation to HTML, allowing you to correctly display special characters from within a macro string.

- You can drop the condition, so «::some text» is an unconditional macro, but "some text" will undergo an extra translation to HTML - for example for converting accented characters, because it is prefixed with "::".

Normally, you cannot insert macros in text documents, because ASCII 199 and ASCII 200 are translated to other strings before macro's are evaluated. If you would like to use your own macro's in text documents, you have to empty STR# 20001, string 199 and 200.

## **UPDATE HISTORY**

### **NEW in 1.2.6**

- Solved a bug in the RTF conversion that would insert spaces if attributes like bold were used in the middle of a word.
- Added RTF conversion for italic.
- Complete redesign of custom strings in resource fork, allowing for much easier customisation. Conditional macro's are used as much as possible. You can now very easily create versions in different languages.
- When including HTML files, their title is now grabbed from between <TITLE> and </TITLE> for insertion into the index file.
- Solved a bug introduced in 1.2.5 that mixed up the index when re-converting previously converted files.
- A lot of string resources have been removed, so do not just cut and paste STR# resources from an older version!

### **NEW in 1.2.5**

- Solved a bug that could cause HTML files that were re-converted to get overwritten and truncated.
- Moved all HTML commands into resource fork for easier customisation.
- Allowed for symbolic characters \r (new line) and \\ (backslash) in strings in resources.
- Cleaned up output files. Split long input lines.

### **NEW in 1.2.4**

- Solved a small bug in the macro expansion, that made original file names with special characters unreadable.

### **NEW in 1.2.3**

- Strip leading digits and spaces from directory names when ordering by file name.
- Changed default file name to 'index.html' (with lower case i).

### **NEW in 1.2.2**

- Macro-strings implemented (not documented) to make conditional inserts in generated HTML code: depending on the presence or absence of files in the destination folder, extra HTML code is generated.
- Restructured STR# 20000 and 20001 using macro strings.
- Auto-insert of bullets, logo, images,... if they are present and named according to a simple naming convention.
- Changeable sorting criterium for generated menu options

### **NEW in 1.2.1**

- If STR# 20000 - string 8 is emptied, the filename of the original file is NOT appended to the converted document.

### **NEW in 1.2.0**

- Resolved bug in RTF file reader: soft hypens are now interpreted OK.
- Added option-drag feature for multiple files: when holding down option key when dragging multiple documents onto TextToHTML, the files are converted without creating index files.

#### **NEW in 1.1.10**

- Added customisable prefixes and suffixes for:
  - normal menu hot links (default <B> and </B>)
  - hot links to sub menus (default empty)
  - hot links to previous level (default empty)
  - hotlinks to upper level (default empty)

#### **NEW in 1.1.9**

- Added translation for the following characters which are supported by Netscape:
  - &reg; -> Registered Trademark -> ®
  - &copy; -> Copyright -> ©
- Removed the <code></code> tags when converting special characters.
- The names of subfolders nested within a dropped folder will now appear in bold in the index file.
- Fixed some RTF conversion bugs.