TextToHTML 1.3.4ß

Internet address: 100025.2724@compuserve.com
Home page: http://ourworld.compuserve.com/homepages/kristiaan

June 11, 1996
If you are updating an older version: the update history is appended at the end of the Readme.

This software is freeware. Please do not distribute it without this Readme file. You may customize this software for your own purposes using the accompanying *Setup TextToHTML* application, but please don't redistribute customized versions: only unmodified versions may be distributed.

This software is distributed 'as is': no warranties are made, either expressed or implied.

The standard TextToHTML adds my name and home page address at the end of converted each page; you may reconfigure TextToHTML not to do this. However, if you do this, I ask you to include my home page address, my name and a mention of TextToHTML in at least one prominent location at the top level of the converted structure, or at the top level of your Web structure.

TextToHTML requires system 7 or higher and > 1MB of free memory.

WHAT DOES IT DO?

TextToHTML quickly converts text or RTF files dropped onto its icon into HTML format. With TextToHTML you can **set up** a **basic WWW structure in 15 minutes**, starting from scratch. 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 include images when they are present.

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

Try the accompanying demo setup first: in the Finder, drop the icon of the folder *Drop me on TextToHTML* onto the icon of *TextToHTML 1.3.3ß,* and, when requested, save *index.html* in the folder *Save* "*index.html*" *in here*. Use your favorite browser to look at the result by opening *index.html* with it.

By default, TextToHTML has the following behaviour:

• 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.

• 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.

• The creator of the converted HTML files is set to MOSS (Netscape).

• If there are GIF or JPEG image files available to use as a logo, bullets, or illustrations, they are be linked in automatically. The names of the image files determines the linking.

• For reference purposes, the filename of the original file is added at the end of the resulting HTML file.

• Most of the features listed above can be customised using the *Setup TextToHTML* application

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: Check your desktop file when TextToHTML refuses to do something. If your Macintosh desktop file is corrupt, TextToHTML will not let you drop anything onto it. In that case, rebuild your desktop first (hold Command-Option during startup).

Tip: If TextToHTML refuses to accept a file for conversion, make sure the file format is correct: the file should really be a text file or an RTF file. If you think a particular file is a text file, and TextToHTML refuses to convert it, try to drop it on SimpleText or TeachText. If they refuse as well, the file is probably **not** a text file.

2) If you select and drop **multiple files** onto the TextToHTML application icon in one move, 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, its sub-folders, its sub-sub-folders, and so on... 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 HTML file contains backlinks to the main index file and to the file one level up in the tree structure. Examine the accompanying example.

Important: Do **not** save the resulting files anywhere **in** the folder you just dropped on TextToHTML. In other words: make sure the destination folder where you save is not contained within the dropped folder. Otherwise, you create some kind of a 'loop': new files to be converted are added to the folder to be converted while the conversion is taking place. Unpredictable behaviour will result.

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! If there is a <TITLE> tag in the HTML file to be copied, the title to be used in the index file will be grabbed from the HTML file. TextToHTML considers a file to be an .html file if it starts with <HTML>. Files that have some other tags at the start will not be recognised.

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 a file called *logo.gif* or *logo.jpg*.

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

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

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

topindex.gif or *topindex.jpg* is used to prefix 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 to small, no write access, or just plain bugs. Let me know if you encounter them!

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

TextToHTML also reads and converts Mac RTF files, but this is still a beta version: 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 similar to the ones TextToHTML uses for your own HTML files, to avoid name conflicts.

HOW TO CUSTOMIZE IT ?

Please do not redistribute customised versions; read the remarks at the top of this Readme file.

Be sure to make a backup copy of TextToHTML before doing customisations.

Using the accompanying *Setup TextToHTML* application, TextToHTML can be heavily customised. Start *Setup TextToHTML* and show it where the TextToHTML application is when it asks for it. *Setup TextToHTML* contains a lot of comments on the customisable items, so just open it and browse through it to learn what you can do with it.

Tip: You can have multiple copies of TextToHTML, each customised in a different way. An example: you could make 3 copies of TextToHTML, call the first one *Français*, the second *English*, and the third *Dutch*. Using *Setup TextToHTML,* you can then open them one by one, and translate the language-dependent features in them. To make life easier on you: all language-dependent strings are concentrated in the *Macro Settings*, macros «50»-«54». You do not need to understand the macro mechanism to translate these strings.

Tip: When TextToHTML generates an HTML file containing an index, the entries in the index are sorted depending on the original file name, not on the title. If you rather want an alphabetical index, you can change the sorting criterium with *Setup TextToHTML*.

Tip: If a certain character is not translated the way you'd like, you can change its translation. An example: the character '∞' (meaning infinity) is currently translated to the string *(inf)*. If you would rather translate it to *oo* (two letters 'o' in a row), which is more visually similar, you can use *Setup TextToHTML* to change the translation.

Tip: When generating web structures, the top level index file is called *index.html* by default. If you'd like another default, you can change it with *Setup TextToHTML*.

Tip: If you want to create a set of pages for use with CompuServe's publishing wizard, which is currently only available for Windows platforms, you can change the rules for generating file names so they can be ported to a DOS environment with 8.3 file names.

Tip: TextToHTML has special handling for carriage return and newline characters: a single carriage return is converted into a <BR>, multiple carriage returns are collapsed into a <P>.   With *Setup TextToHTML, y*ou can turn off this special handling. If you do, you should define the translation of carriage return (ASCII 13) to be *\r<BR>* instead of an empty string.

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 macro string with the same number. If this string contains other macros, they are also evaluated.

• There are also some symbolic macro's and special macro's:

«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.
«DAY» is the day number of the conversion (1-31).

«MONTH» is the month of the conversion (1-12).
«YEAR» is the year of the conversion (2 digits)
«YEAR4» is the year of the conversion (4 digits)
«HOUR» is the hour of conversion
«MINUTE» are the minutes
«SECOND» are the seconds
«DAYNAME» is the name of the weekday
«MONTHNAME» is the name of the month
«3» is today's date in a readable format. You can change the format of «3» with Setup TextToHTML
«55» - «73» are day names and month names.

In later versions, more of these symbolic macros will become available, like time and date,...

• 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 from Mac ASCII 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 from ASCII to HTML - for example for converting accented characters to ISO Latin-1, because it is prefixed with "::". It is easier to type «::français» than to type fran&ccedil;ais.

Tip: Normally, you cannot use 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 string 199 and 200 in the character translations (ASCII Matrix).

SPECIAL FEATURES

• If necessary, *TextToHTML* can use the name of the original file as the document title and header, instead of the first line. This is done with *Setup TextToHTML*, in the General Settings - Title Scan. In this case, the first line is handled like any other line.

• The names of the HTML-files can be changed by changing the C-format string in the General Settings - File Names. For example, by changing the format string from **X%04d_** to **hhh%06d**, generated filenames will look like: hhh000001.html, hhh000002.html, ...

• By prefixing this format string with an equal sign (eg. =**X%04d**_), the filenames for the converted files are equal to the original file names, but with the HTML-suffix appended. Be aware, that if there are duplicate filenames, overwrites will occur, so do not use this option if you are not sure. The format string still applies for the generation of index file names though. Some characters are not allowed in the file names (/, %) and are replaced by something else.

UPDATE HISTORY

NEW in 1.3.4

• Allowed to use the original file name as the title of the generated file.

NEW in 1.3.3

• Allowed customising of generated file names.
• Added the conversion of quotes in RTF files.
• HTML input-files do not need to start with **<html>** anymore to be recognised. They are also recognised if they start with a **<** character and have a file name that ends in *.htm*l or *.htm.*
• Defined the macros «DAY», «MONTH», «YEAR», «YEAR4»: they generate the day, month, year (2 digits), year (4 digits).

• Defined the macros «HOUR», «MINUTE», «SECOND»: they generate the hour, minute, seconds.
• Defined the macros «DAYNAME», «MONTHNAME»: they generate macro's «weekday + 54», «month + 61», which are then converted themselves to daynames and monthnames. To translate daynames and monthnames: check and translate macro's 55-73. Also, check macro 3: it contains a readable date format you might want to change.
• The length of an expanded macro was limited to 4KB in previous versions. Now it is only limited by available memory.

NEW in 1.3.2

• Added <CENTER> tag support for RTF files
• The special handling of carriage returns can now be turned off in Setup TextToHTML.
• Bug fix: overlapping bold, underline, italic zones in RTF are now converted correctly to HTML.
• Added some tips to the ReadMe

NEW in 1.3.1

• Bug fix: added missing " in default macro «15»
• Added some tips to the ReadMe
• Bug fix: fixed 'Save As...' in *Setup TextToHTML*

NEW in 1.3.0

• Minor bug fixes in *Setup TextToHTML*; addition of Edit-menu; new About box.

NEW in 1.2.9

• Release of *Setup TextToHTML* which greatly simplifies configuration of TextToHTML.

NEW in 1.2.8

• Changed the translation of ß from capital B into &szlig; (thanks to Christian Griesbeck for reporting this one).
• Added a few new parameters, so you can influence the length of the generated file names and the extension. If you want to be able to move the generated web structure to a DOS platform, you must change STR# 20000, string 4 from *index.html* into *index.htm* (3 character DOS extension), change STR# 20000, string 14 from *.html* into *.htm* (3 character DOS extension), and change STR# 20000, string 16 from *31* into *12* (max file name length in DOS). I needed to do this in order to be able to use TextToHTML for preparation of web pages on CompuServe: currently you can only upload pages from a Windows environment.

NEW in 1.2.7

• Solved some RTF conversion bugs (special characters like {}\ were not converted OK). Word tables are now more readable after conversion.

• I left Logic, the Belgian AppleCentre where I worked for 5 years, because I am going to live in to New Zealand, where I'll be working at Capital Micro Centre in Wellington. At this very instant, TextToHTML is more or less homeless - this and some future versions will be available from www.logic.be, but because it will be difficult to maintain the Logic WWW pages from New Zealand, I'll try to find a new home for TextToHTML once I am in New Zealand. Also, I'll upload TextToHTML to the Internet Resources forum on CompuServe.

If you want to know where the new home will be, send me an e-mail (100025.2724@compuserve.com) from October 1995 on.

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 introducted 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.