ANL/MCS-TM-207

Users Manual for tohtml:
Producing True Hypertext Documents from LaTeX

by

William Gropp
Mathematics and Computer Science Division

MATHEMATICS AND
COMPUTER SCIENCE
DIVISION

Users Manual for tohtml:
Producing True Hypertext Documents from LaTeX

by

William Gropp

Abstract

The World Wide Web has made it possible to use and disseminate documents as "hypertext." One of the major advantages of hypertext over conventional text is that references to other documents or items can be linked directly into the document, allowing the easy retrieval of related information. A collection of documents can also be read this way, jumping from one document to another based on the interests of the reader. This does require that the hypertext documents be extensively cross-linked. Unfortunately, most existing documents are designed as linear documents. Even worse, most authors still think of documents as linear structures, to be read from front to back. To deal with this situation, a number of tools have been created that take documents in an existing word-processing system or markup language and generate "HTML," the hypertext markup language used on the Web. While this process makes a single document available in a convenient form on the Web, it does not give access to cross-document linking, a major advantage of hypertext. This manual describes a program, `tohtml`, that takes LaTeX input files, as well as files of link information, and produces a hypertext document that can contain extensive cross-links. A related program, `doctext`, aids in the generation of manual pages that can be referenced by a LaTeX document.

1 Introduction

Generating true hypertext documentation is a task that requires skill and hard work. The tools described in this report are intended to help you create simple hypertext documents that include multimedia elements as well as extensive cross-referencing, without requiring you to learn a new word-processing system (as long as you already know LaTeX [2]). The documents produced by this system, while relatively simple, do include hot links to other documents, the ability to link individual names (such as routine or person names) to other documents, and simple multimedia elements.

Specifically, we describe the program `tohtml`, which converts LaTeX files to HTML (the Web hypertext markup language). This C program is fast and can handle very large documents. The program `doctext` can be used to generate manual pages for routines written in C that can be referenced automatically by documents built with `tohtml`. Finally, a LaTeX style file, '`tools.core/info2rtf/anlhtext.sty`' provides a way to use LaTeX and include hypertext links, graphics, and movies into a document.

Other tools exist for translating LaTeX to HTML; perhaps the best known is LaTeX2html. This program has different strengths and weaknesses when compared with `tohtml`; if you have trouble with one program, you should try the other.

LaTeX in this document refers to the original LaTeX, not the new and incompatible LaTeX2e. Support for LaTeX2e may be provided in the future.

2 tohtml

The `tohtml` program is a C program that can read LaTeX files and generate HTML files. Command-line options give you control over the output, including the way that the document is split into separate HTML files and the way complicated expressions are handled.

2.1 Getting Started

Using `tohtml` is easy. For example, if you wish to convert a LaTeX file named '`mypaper.tex`' into an HTML document, you can do

```
tohtml -default mypaper.tex
tohtml -default mypaper.tex
```

Note that, just as you normally need to run LaTeX twice to get all of the references correct, you also need to run `tohtml` twice. The command-line option `-default` provides good basic choice of the many options provided by `tohtml`. The first run will also generate a number of error messages (such as `Could not find Introduction in topicctx` and `\ref{fig:upshot1} unknown (mypaper.tex line 792)`); the second run should generate no error messages if all goes well.

The output will be left in a subdirectory with name '`mypaper`'; you can view the result with `xmosaic mypaper/mypaper.html`.

One important note: for `tohtml` to handle bibliography entries correctly, you should keep the files that LaTeX and BibTeX generate (e.g., the '`mypaper.bbl`' file).

2.2 Command Line Arguments

To use `tohtml`, you give it the name of the LaTeX file to process:

```
tohtml foo.tex
```

Command-line options to `tohtml` allow you to change the behavior. The option `-default` sets a common selection of options:

```
tohtml -default foo.tex
```

A complete list of the command-line options follows. Some of these (e.g., `-mapref`) are discussed in more detail later.

The following options control how the document is divided into individual Web pages.

-nocontents Suppress the generation of the contents page (which has links to all of the other sections). This is useful for short documents.

-split n Split the document into separate files, based on the sectioning commands (e.g., `\chapter`, `\section`). The value of `n` indicates where to begin breaking the file: a value of `0` puts chapters into separate files, a value of `2` puts subsections into a separate file, and a value of `-1` forces the entire document into a single file (note that any image or graphics files are still separate).

The following options control how `tohtml` handles various kinds of LaTeX commands.

-citeprefix str

-citesuffix str Set the prefix and suffix strings used around `\cite` in text. The default values are `[` and `]`.

-cvtlatex Convert LaTeX that cannot be represented directly by HTML into bitmaps that are included in the document. If this option is not used, the text is processed as is, with all LaTeX commands removed and text (and arguments to the commands) unchanged.

-cvttables Convert tabular environments into bitmaps. This produces nicer tables than the default (which attempts to line up columns) but can produce large bitmaps that some systems may not be able to handle.

-cvtmath Convert math and display math environments into bitmaps. This produces nicer representation of the formatted mathematics, but can produce large bitmaps.

-iftex Include text in `\begin{iftex}` to `\end{iftex}` (for LaTeXInfo documents).

-simplemath Use italics for expressions in LaTeX math mode that do not involve any special characters. For example, `$3$` would be changed into an italic 3, rather than a small bitmap. Not yet available.

-default Set a common set of options. Equivalent to `-cvtlatex -cvttables -cvtmath -iftex -split 2 -useimg`.

-basedef filename Read filename for additional definitions. This lets you define some TeX and LaTeX commands as having a specific behavior when generating HTML files (see section 5.5 [User-defined Replacements], page 7).

-userpath pathlist For each `documentstyle` option or `usepackage` name, look in `pathlist` for a file of the form `name.def`. For example, `\usepackage{refman}` will cause `tohtml` to search for `refman.def` in `pathlist`. If `-userpath` is not given, no files will be searched for. The environment variable `DOCTEXT_USERPATH` may be used instead of `-userpath`.

The following options control some aspects of the layout of each page, particularly the presence of the navigation buttons to other pages.

-notopnames Suppress generation of the links at the top of the page. This is useful when generating a single-page document, in combination with `-split -1`.

-nonavnames Suppress generation of the links at the bottom of the page. This is useful when the sections in the document are short.

-nobottomnav Suppress generation of the links and buttoms at the bottom of each section.

-beginpage filename Add the HTML in `filename` to the top of every generated HTML page.

-endpage filename Add the HTML in `filename` to the bottom of every generated HTML page. This is particularly useful for adding links to indexes and home pages to a document.

The following command-line arguments control the generation of links to other documents.

-mapref filename Filename contains a list of mappings from citation keys (the name in a `\cite` command) to HTML links (see section 4 [Building a Map File], page 5).

-mapman filename Filename contains a list of mappings from tokens (currently defined as sequences of letters only) to HTML links (see section 4 [Building a Map File], page 5).

These commands control details of the generated HTML file.

-gaudy Use images of colored balls for bullets in itemized lists.

-useimg Instead of generating the bitmap, uses a file that has the name that would be used for that bitmap. This is useful when making the second run of `tohtml`. It is meaningful only when `-cvtlatex` is used.

-basedir dirname Add an HTML base command to the main file. For example, `-basedir "http://www.mcs.anl.gov/mpi"` causes

```
<base href="http://www.mcs.anl.gov/mpi">
```

to be written to the file.

The following commands are used for debugging the behavior of tohtml itself.

-debugout Write out information about the generation of output

-debugdef

-debugscan

-debugfile

3 doctext

One of the most difficult tasks in creating extensive hypertext is generating the initial documents and providing an easy way to link to them. The `doctext` [1] program can be used to generate versions of Unix-style man pages from the C source code of routines. This program can generate `nroff` (for using `man` and `xman`), LaTeX (for generating printed manuals), and HTML. For the HTML to be useful, there must be an easy way to create links to the generate documents. This section describes how to do that; `doctext` itself is documented in [1].

To generate HTML man pages of a collection of source files in '`/home/me/foo`', do the following:

```
cd
mkdir www
mkdir www/man3
cd foo
doctext -html -index ../foo.cit \
        -indexdir http://www.mcs.anl.gov/me/foo/www/man3 \
        -mpath ../www/man3 *.[ch]
cd ..
```

This puts the HTML files into the directory '`www/man3`' and the index (in the correct format for `-mapman`) into file '`foo.cit`'. Once you are sure that the files are correct, you can move them into the Web area with

```
cp -r www /mcs/www/home/me
```

(assuming that '`/mcs/www`' corresponds to `http://www.mcs.anl.gov` in the `-indexdir` argument).

To generate an HTML listing of the routines, you can execute the following script, with, of course, the appropriate text:

```
#! /bin/sh
echo '<TITLE>Web pages for My Routines</TITLE>' >> www/index.html
echo '<H1>Web pages for My Routines</H1>' >> www/index.html
echo '<H2>My Routines</H2>'  >> www/index.html
```

```
echo '<MENU>' >> www/index.html
ls -1 www/man3 | \
    sed -e 's%^\(.*\).html$$%<LI><A HREF="man3/\1.html">\1</A>%g' \
        >> www/index.html
echo '</MENU>' >> www/index.html
```

If you have only a few routines to document, you can dispense with the second directory level above (the 'man3'). However, you might find it valuable to follow (at least loosely) the Unix man-page format, with commands and installation instructions in 'man1' and routines spread across 'man1' to 'man8'.

4 Building a Map File

One of the most useful features of `tohtml` is its ability to automatically replace citations and names with hot links into other documents. This is done by providing one or more `-mapref` files (for citations) and `-mapman` files (for names, currently defined as strings consisting only of letters). Currently, the format of this file is fairly ugly but easy to use:

`cite:+`*cite-name*++*cite-text*++++*kind*+*URL*

where the items are as follows.

cite-name A citation key that is used in a LaTeX `\cite` command, such as `\cite{tohtml-user-ref}`.

cite-text The name that should replace the citation use. For example, if the document contains a `\cite{tohtml-user-ref}`, and you wish this to turn into a hot link with text `[Tohtml User Manual]`, you would use `Tohtml User Manual` for *cite-text*.

kind The field is used to indicate the type of the reference. This is not yet implemented. Use the value `manual` for user manuals and `cite` for everything else for the time being.

URL The URL (Universal or Uniform Resource Locator) that the item will link to. These are paths such as `http://www.mcs.anl.gov/home/gropp/tohtml/tohtml.html`.

All of these items must be on a single line; the exact number of `+` characters must be used.

You can comment out a line in a mapref or mapman file by placing a "`;`" in the *first* column.

Here is a sample map file for citations. Note that `\cite{gropp}` will turn into `gropp@mcs.anl.gov` in the text, with a link to the Web home page for gropp.

```
cite:+sles-user-ref++SLES User Manual++++manual+http://www.mcs.anl.gov/petsc/solvers.html
cite:+gropp++gropp@mcs.anl.gov++++person+http://www.mcs.anl.gov/home/gropp/index.html
cite:+petsc-man-pages++PETSc Manual Pages++++manual+http://www.mcs.anl.gov/petsc/ref/index.html
```

4.1 Map Files for man Pages

Building a map file for a list of man pages generated by, for example, `doctext`, is a daunting prospect. Fortunately, the map file can be generated automatically when the man pages are created if `doctext` is used. To do this, add the options

`-index` *indexfile* `-indexdir` *indexdir*

to the `doctext` command. The *indexfile* is a file to which entries in the proper form for `-mapman` will be *appended*. The value of *indexdir* is the root directory for the generated man pages. For example, to use '`http:/www.mcs.anl.gov/home/gropp/man`' as the root and the file '`manref.cit`' to hold the `-mapman` references, use

```
doctext -index foo.cit \
    -indexdir "http:/www.mcs.anl.gov/home/gropp/man" ...
```

5 Fine-Tuning

This section discusses how to customize the appearance of the pages generated by `tohtml`, how to interpret error messages, and how to extend the translations that `tohtml` recognizes.

5.1 The anlhtext Style File

One way to fine-tune a document is to use a special LaTeX style file that contains LaTeX commands that have special meaning to `tohtml`. This makes it possible to maintain a single document for both hardcopy and hypertext versions of a document.

URL Gives the URL for a document. In LaTeX, it is set in the current font; in HTML, it becomes an active link whose text is the link name. Usage is `\URL{url-text}`.

AURL Gives the URL with a different link text. In LaTeX, gives the URL in the `\tt` font, followed by the link text. In HTML, just the link text is displayed; clicking the link text jumps to the URL. Usage is `\AURL{url}{link text}`.

hcite Includes a link to a URL that appears *only* in the HTML document. Usage is `\hcite{url}`.

hcitea Includes a link to a URL with a specified text string; the URL is used only in the HTML version. Usage is `\hcitea{url}{text string}`.

href Refers to a section or subsection by title. For example, if you used `\subsection{A sample}`, then `\href{A sample}` would provide a link to that subsection in the HTML. In LaTeX, it sets its argument in the current font. Usage is `\href{section title}`.

The following have not yet been implemented. Comments are welcome.

htmlhr Generates an *hrule* in the HTML. Ignored in LaTeX.

html envirionment `\begin{html}` to `\end{html}` may be used to imbed HTML commands directly; in LaTeX, no output will be generated.

movie Includes a reference to an MPEG movie file with a given icon.

sound Includes a reference to a sound file with the generic sound icon.

graphic Includes an image (gif) file.

5.2 Adding Pointers to Other Pages to Every Page

In many situations, you would like to add pointers to the bottom of every page, for example, a pointer back to a home page. With `tohtml` this is very easy. First, create a file, say '`bottom.html`', that contains the appropriate HTML code:

```
Return to <A HREF="node179.html">MPI Standard Index</A><BR>
Return to <A HREF="http://www.mcs.anl.gov/mpi/index.html">MPI home page</A><BR>
```

The first line provides a return to a document in the current directory, in this case, the index of the MPI Standard. The second line provides a return to the MPI home page; this will work from any location.

To have these incorporated into the output of `tohtml`, use the option `-endpage`:

```
tohtml -endpage bottom.html ...
```

5.3 Adding a disclaimer to the top of every page

Sometimes, you may wish to add text to the top of every generated page. You can do this by putting the text into a file, say '`top.html`', and then using the `-beginpage` option of `tohtml`:

```
tohtml -beginpage top.html ...
```

Here '`top.html`' might contain

```
This material is a draft and should not be quoted<BR><HR>
```

5.4 Errors

Errors are reported in two places. Some messages go to standard error. Lists of LaTeX commands unknown to `tohtml` or citations without hyperlinks are written into the file '`latex.err`'. In addition, `tohtml` will not let you redefine some basic LaTeX commands such as `\section`; normally, this is the behavior that you'd want anyway.

5.5 User-defined Replacements

You can provide your own definition of many TeX and LaTeX commands by including a *definitions* file. This file contains entries of the form

operation name optional-string

Lines may be commented out by starting them with the character `#`.

operation This is one of `nop` (no operation and remove arguments), `dimen` (TeX dimension) `asis` (no operation but leave arguments), `name` (replace with a string), or `raw` (replace with verbatim text, such as an HTML command), and `exec` (insert TeX commands).

name This is the name of the TeX or LaTeX command, without the initial backslash. That is, to redefine `\samepage`, use the name `samepage`.

optional-string This is null (for `dimen`), an integer for the number of arguments (for `nop` or `asis`), or the replacement string (for `name`). Currently, there is no way to specify leading or trailing spaces.

For example, when `tohtml` starts, it reads a file that contains the definitions of some basic commands, such as

```
dimen topsep
asis underline 1
nop thispagestyle 1
name mu 0 u
raw seprule 0 <HR>
```

The first line states that `\topsep` is a dimension; `tohtml` will understand what to do when it sees this name. The second line says that `\underline` is a command with a single argument and that the appropriate HTML for this is to use the argument (thus, `\underline{abc}` will become `abc` in the HTML document). The third line says that `\thispagestyle` is a command with a single argument and can be ignored (performs no operation, or 'no-op'). The fourth line tells `tohtml` to replace `\mu` with `u`. The last line tells `tohtml` to replace `\seprule` with the HTML command `<HR>`.

Any number of `-basedef` files may be specified; they are processed in order starting from the left. The default basedef file is always read first.

6 Examples

This section gives examples of using `tohtml` for various kinds of documents.

6.1 Single-Page Descriptions

Sometimes, you wish to use LaTeX sectioning commands to generate headings in the HTML file without generating separate HTML files, for example, when each section is very short. This example shows how to accomplish this. In addition, it

- adds some HTML to the bottom of the page (`-endpage petsc_end.html`),
- suppresses the contents list (`-nocontents`),

- suppresses the links to jump between sections (`-notopnames -nonavnames`),
- makes the use of `\cite` *not* place brackets around the reference (`-citeprefix "" -citesuffix ""`),
- automatically converts citations to links to other documents (`-mapref ../../html/petsc.cit`), and
- places the output into a separate directory (`split 0` instead of `split -1`).

```
tohtml -mapref ../../html/petsc.cit -endpage petsc_end.html \
        -nocontents -notopnames -nonavnames -split 0 \
        -citeprefix "" -citesuffix "" domain.tex
tohtml -mapref ../../html/petsc.cit -endpage petsc_end.html \
        -nocontents -notopnames -nonavnames -split 0 \
        -citeprefix "" -citesuffix "" domain.tex
```

In addition, the option `-gaudy` may be used to add color to any lists.

6.2 Chosing LaTeX executable

Because of the introduction of LaTeX2e, which is often installed as just `latex`, it may be necessary to select a particular `latex` executable. You can do this with the environment variable `TOHTML_LATEX`; just set that to the path needed for LaTeX. For example,

```
setenv TOHTML_LATEX /usr/local/bin/latex
```

6.3 Books

A major advantage of `tohtml` is its ability to handle very large documents. Here are the commands that are used to process the MPI Standard:

```
tohtml -default -endpage ../mpi-tail.html mpi-report.tex
tohtml -default -endpage ../mpi-tail.html mpi-report.tex
```

The `-endpage ../mpi-tail.html` option is used to add pointers to the index of the MPI standard and the MPI home page to every page.

7 Comments, Bugs, and Suggestions

Please send any comments to `gropp@mcs.anl.gov`.

References

[1] William Gropp. Users manual for `doctext`: Producing documentation from C source code. Technical Report ANL/MCS-TM-206, Argonne National Laboratory, March 1995.

[2] Leslie Lamport. *LaTeX: A Document Preparation System.* Addison-Wesley, 1986.